

ABSTRACT

MAJUMDER, TUHIN. Statistical Learning Using Sparse Markov Models. (Under the direction of Soumendra Nath Lahiri and Donald E.K. Martin).

Discrete time Markov models are extremely popular for analyzing a categorical time series due to their wide applicability. Especially, higher-order Markov models can capture more complex dependence of a categorical time series. However, with increasing order, the complexity of the model also increases in terms of number of parameters. In this dissertation, we consider a more general parsimonious modeling approach is given by Sparse Markov Models (SMMs).

In Chapter 2, we give a thorough review of the large sample properties of Markov chains, which is useful in extending the large sample results in chapter 3 for higher order Markov models including SMM. In Chapter 4, we develop an elegant method of fitting SMMs based on convex clustering algorithms, which minimizes a convex and penalized loss function. Theoretical results establish model selection consistency of our method for large sample size. Extensive simulation and real data example in classifying RNA sequences of different viruses demonstrate the wide applicability of such method. In Chapter 5, we extend the previous method for a more general class of divergence measure. We provide theoretical results which enable us to find a range of the regularization parameter for which the true underlying clusters can be identified. A more relaxed method of convex clustering is proposed, namely SR2C2, which performs comparably with the traditional methods, but in much less time. To compare the time complexity and the model performances for different algorithms, extensive simulation studies have been conducted. In Chapter 6, we propose a bootstrap based prediction algorithm for predicting the h -step ahead future, demonstrated with extensive simulation studies. A computationally efficient method of constructing simultaneous $100(1 - \alpha)\%$ prediction sets for the future observations is introduced, based on an anomaly scoring method. We demonstrate this method in detecting anomalous genes in *Helicobacter Pylori* bacteria.

© Copyright 2022 by Tuhin Majumder

All Rights Reserved

Statistical Learning Using Sparse Markov Models

by
Tuhin Majumder

A dissertation submitted to the Graduate Faculty of
North Carolina State University
in partial fulfillment of the
requirements for the Degree of
Doctor of Philosophy

Statistics

Raleigh, North Carolina
2022

APPROVED BY:

Eric C. Chi

William Boettcher

Soumendra Nath Lahiri
Co-chair of Advisory Committee

Donald E.K. Martin
Co-chair of Advisory Committee

DEDICATION

To my parents, Tushar and Aparna.

BIOGRAPHY

Tuhin Majumder was born on March 13, 1996, and brought up in Burdwan, a medium sized town in the West Bengal state of India. Tuhin completed his secondary education (10-th standard) from Burdwan Municipal High School in 2011, and his higher secondary (12-th standard) from Ramakrishna Mission Vidyalaya, Narendrapur in 2013. After school life, Tuhin joined the Indian Statistical Institute, Kolkata (ISI Kolkata) in 2013, and completed his Bachelor of Statistics (Hons.), or the B.Stat (Hons.) in 2016 with Distinction. Tuhin also received his Master of Statistics (M.Stat) from ISI Kolkata in 2018 with Distinction. In August 2018, Tuhin joined the Department of Statistics in North Carolina State University to pursue his PhD degree. Under the guidance of Dr. Soumendranath Lahiri and Dr. Donald Martin, he successfully finished his dissertation in July, 2022.

ACKNOWLEDGEMENTS

My PhD is a journey of four years, which could not have been so enjoyable without the support and motivation from a bunch of people. I never got an opportunity to thank these people in such an organised way, so I would like to utilize this acknowledgement section to fulfill that purpose.

First of all, I would like to thank my parents. My father, Tushar Kanti Majumder has helped me in my childhood to fall in love with Mathematics, and eventually helped finding a subject to pursue my career. My mother, Aparna Majumder has supported me throughout in my life every time, guided me properly whenever I needed any sort of help. Without their sacrifice, dedication and motivation, even from a distance of 8000 miles, I could not succeed in my life.

Next, I would like to express my gratitude to my advisors, Dr. Soumendra Nath Lahiri and Dr. Donald Martin. I am indebted to them for their care, guidance, motivation, as well as their patience with me for finishing this dissertation. I have been very fortunate to have them as my advisors. For the last four years, they have helped me every time I got stuck in a research problem, helped me preparing for my future endeavours and most importantly, they have helped me understanding my research problem on sparse Markov models. I would thank Dr. Eric Chi for serving as a committee member, who helped me improving my dissertation by his invaluable inputs and suggestions. Also, I have taken two wonderful Statistical computing courses under Eric's instruction, which have played a key role in my thesis. I would acknowledge Bill (Dr. William Boettcher) for agreeing to serve as a committee member. Bill was also one of my collaborators in an inter-disciplinary project under Laboratory of Analytical Sciences (LAS), it was a really nice experience to work with him. I would also like to thank Dr. Subhashis Ghoshal, Dr. Jacqueline Hughes-Oliver, Dr. Marie Davidian, Dr. Ana Maria Staicu, Dr. Sujit Ghosh, Dr. Minh Tang for teaching me different courses over four years. Thanks to Dr. Charles Smith for his candies while I used to take office hours for his courses. I would also thank the staffs of our department; Lanakila and Alison for helping me with all administrative problems, and Terry for his tremendous effort to improve the departmental cluster facility.

Before coming to NC State, I had a few great professors in ISI. I would express my gratitude to Dr. Ayanendranath Basu, who taught us three statistics course and supervised my Master's dissertation.

A special thanks to Dr. Abhik Ghosh. (Abhik Da), who was also a co-author of that paper. I would thank Dr. Parthanil Roy, Dr. Rajat Subhra Hazra, Dr. Saurabh Ghosh and Dr. Tapas Samanta for their wonderful teaching. I would thank Dr. Anil Ghosh and Dr. Indranil Mukherjee for their teaching and recommending me to NC State. Also, I would thank my high school Math teacher Mr. Sayebur Rahaman who first introduced advanced Mathematics to me and inspired me to join a prestigious institute like ISI.

Four years back, when I left my country to pursue the PhD degree in Statistics at NC State, I barely thought I would make a second home away from my home in Raleigh. Fortunately, I have found some wonderful friends in Raleigh, who made my journey so much smooth. First of all, I would like to thank Dr. Salil Koner (Salil Da), for being an immensely helpful and caring roommate. Thank you Sukanya, even though I knew you from ISI, it would not be possible to rediscover you as an wonderful friend and roommate if you were not my batchmate at NC State. Also I thank Shubhajit, Sumitosh and Ashwin for staying with me as roommates over the years. I want to thank Dr. Arnab Chakraborty (Chak), Dr. Suman Majumder (Suman Da), Dr. Rahul Ghosal (Rahul Da), Dr. Indrabati Bhattacharya (Indrabati Di) and Dr. Dhrubajyoti Ghosh (DJ) for making my transition to Raleigh easier, having acquainted me with the grad life and social life at NC State. I would thank Dr. Indranil Sahoo (Sahoo Da), Dr. Sayak Mukherjee (Sayak Da), Rahul Chakraborty (Rahul Chak), Dr. Debanjan Chatterjee (Debanjan Da) and the juniors in the department: Subhankar, Indrila and Samhita. Not only in Raleigh, I have made a few great friends in Chapel Hill and Durham as well. Thank you Dr. Arkopal Chowdhury (Arkopal Da), Dr. Arkaprava Roy (Gol Da), Dr. Akash Roy (Akash Da), Imon and Sampreeti. I would thank Shounak in Durham, my old friend from ISI, with whom I have discussed sports hours after hour via messaging just like our college days. I have been fortunate to have Sumit, my one of the closest friend in the college days in this triangle area. I don't know how I can express gratitude to him, probably by a little leg pulling, not so hard that he injures his ligament for the third time. Jokes apart, he is one of the best human beings I have ever met. I have spent some wonderful moments with everyone, starting from a grand dinner, playing cricket or board games, watching movies, arranging Durga Puja and Saraswati Puja, numerous trips and endless gossip.

Beyond the triangle area, I have a few friends in different parts of USA and India. I would thank

everyone of them, especially Soumya, Imon Banerjee, Arnab, Subha (Bapi Da), Subrata, Sayan Das, Souvik, Debosmita; with whom I spent a lot of time in the college days, and am regularly in touch with them now. I would thank Anamitra, one of my heartiest friends and the only one who was my batch mate in both school and in 5 years of ISI. Thank you Shouryya, my childhood friend, all these years we have grown up but the friendship remained intact. Thank you my youngest pal Arijit from my hometown Burdwan, having you like a younger brother is always enjoyable. Every person mentioned here are very close to my heart, and I feel blessed to have friends like you in my life.

This list of acknowledgement would be incomplete without one person. My dearest friend, philosopher, guide; my go-to in every occasion. Rupam, I don't know how to thank you in this short span of space, probably I would need a document of the length of this thesis to express my gratitude. Whenever I am in joy or trouble, be it academically, emotionally or mentally, I know there is a person, to whom I can express everything. The support, love and faith I have received from you throughout my PhD life and even way before that, have helped me succeed in my life. Even though you have been living in Michigan for the last four years and of course we cannot meet very frequently, I don't feel detached as I share everything with you all the time, be it a meme, criticizing some celebrity or some random gossip. As an emerging Bio-statistician, you are an inspiration for the researchers like me. Best of luck for your upcoming career and a happy married life with Dishari, whom I would also like to thank for being a very good friend of mine.

Last, but not the least, I am extremely grateful to the Real Madrid football club. The weekends after a strenuous week would not become so enjoyable without Real Madrid's matches. The incredible comebacks from a goal or two down, the determination of the players till the final whistle, winning trophies from not being the favourite choice in the beginning, have always motivated me throughout this journey. As a Madridista, I have never given up. Whenever I have faced problems with my research works, I got stuck in the same place, I did not panic. I just believed on myself as a Madridista and remember the famous lines "The Real Madrid shirt is white. It can stain of mud, sweat and even of blood, but never of shame!" Thank you Real Madrid for everything.

And finally, thanks to everyone whom I have missed to mention in this acknowledgement.

TABLE OF CONTENTS

List of Tables	ix
List of Figures	x
Chapter 1 Introduction	1
Chapter 2 Literature Review: Large Sample Properties of Markov Chains	5
2.1 Introduction	5
2.2 Definitions and Notations	8
2.2.1 Countable State Spaces	8
2.2.2 General State Spaces	9
2.3 The Stationary Distribution	11
2.4 Ergodicity	13
2.5 Central Limit Theorems	15
2.5.1 CLT for a Countable State Space	16
2.5.2 CLT in a General State Spaces	17
2.6 Concluding Remarks	20
Chapter 3 Large Sample Properties of Sparse Markov Models	21
3.1 Introduction	21
3.2 Central Limit Theorem	23
3.3 Example	27
3.4 Conclusion	28
Chapter 4 Fitting Sparse Markov Models Using Regularization	29
4.1 Introduction	29
4.2 Methodology	34
4.2.1 Notation	34
4.2.2 Description of the Method	34
4.2.3 Computational considerations	36
4.2.4 Selection of the Tuning Parameter	37
4.3 Conditions and Theoretical Results	39
4.3.1 Conditions	39
4.3.2 Main results	40
4.4 Simulation study	44
4.4.1 Simulation Set-up 1	45
4.4.2 Simulation Set-up 2	48
4.5 Real Data Analysis	49
4.5.1 Data Description	51
4.5.2 Method	52
4.5.3 Results	52
4.5.4 Discussion	56
4.6 Summary	57

Chapter 5	Fitting Sparse Markov Model by Generalized Convex Clustering Algorithm . . .	59
5.1	Introduction	59
5.2	Fitting SMM Using Generalized Loss Function	63
5.2.1	ADMM Updates	64
5.2.2	AMA Updates	67
5.3	Application of SMMGECCO to Density Power Divergence	69
5.3.1	ADMM for Kullback-Leibler Loss	71
5.3.2	AMA for DPD Loss	72
5.3.3	Stopping Criteria	76
5.4	Perfect Recovery Conditions for General Loss	78
5.5	Sparse Relaxed Regularized Convex Clustering (SR2C2) Algorithm	85
5.6	Simulation	88
5.6.1	Computational Complexity Comparison	88
5.6.2	Clustering Performance	91
5.7	Discussion	93
Chapter 6	Prediction in Sparse Markov Models	95
6.1	Introduction	95
6.2	Score Based Simultaneous Prediction Set in SMM	96
6.2.1	Prediction in Simple Order One Markov Models	97
6.2.2	Anomaly Score Computation in SMM	105
6.3	Point Prediction by Bootstrap	107
6.3.1	Simulation Set-up	108
6.3.2	Results	109
6.4	Real Data Application: Anomaly Detection in Helicobacter Pylori Bacteria	115
6.4.1	Method	115
6.4.2	Result	116
6.5	Conclusion	117
References	118
APPENDICES	124
Appendix A	Proof of the Theorems in Chapter 3	125
A.1	Proof of Theorem 3.2.1	125
A.2	Proof of Theorem 3.2.2	128
A.3	Proof of the Example in Section 3.3	130
Appendix B	Proof of the Theorems in Chapter 4	133
B.1	Proof of Theorem 4.3.3	133
B.2	Proof of Theorem 4.3.4	134
B.3	Proof of Theorem 4.3.5	136
B.4	Proof of Theorem 4.3.6	139
B.4.1	Proof of Corollary 4.3.6.1	140
B.4.2	Proof of Corollary 4.3.6.2	140
Appendix C	Supplements for Chapter 5	141
C.1	Stopping Criterion for ADMM	141
C.2	Dual Objective for AMA for Squared Error Loss	143

LIST OF TABLES

Table 4.1	Clustering Performance for $m = 3, w_{i,j} = 1$	46
Table 4.2	Clustering Performance for $m = 3, w_{i,j} = \exp^{-\phi \ \hat{\pi}_i - \hat{\pi}_j\ _2^2} l_{i,j}^k$	46
Table 4.3	Clustering Performance for $m = 3, w_{i,j} = \exp^{-\phi \ \hat{\pi}_i - \hat{\pi}_j\ _2^2} l_{i,j}^{k(\infty)}$	47
Table 4.4	Clustering Performance for $m = 2, w_{i,j} = \exp^{-\phi \ \hat{\pi}_i - \hat{\pi}_j\ _2^2} l_{i,j}^k$	47
Table 4.5	Clustering Performance for $m = 2, w_{i,j} = \exp^{-\phi \ \hat{\pi}_i - \hat{\pi}_j\ _2^2} l_{i,j}^{k(\infty)}$	48
Table 4.6	Clustering Performance for Different Weight Choice and Sample Size for Simulation 2.	49
Table 4.7	Number of clusters and size of each cluster obtained in Model 1.	53
Table 4.8	Confusion Matrices for $\epsilon = 0.05, 0.1$ and 0.25 respectively with mis-classification rates 22.6%, 15.2% and 2.8% in Model 1	55
Table 4.9	Confusion Matrices for $\epsilon = 0.05, 0.1$ and 0.25 respectively with mis-classification rates 22%, 16% and 4% in Model 2	55
Table 5.1	Model selection performance for all algorithms.	92
Table 6.1	Summary of the Predicted Sets.	110
Table 6.2	Bootstrap Estimates of h -step Transition Probabilities with $B = 10000$	110
Table 6.3	Bootstrap Estimates of h -step Transition Probabilities with $B = 25000$	111
Table 6.4	Bootstrap Estimates of h -step Transition Probabilities with $B = 100000$	112
Table 6.5	Bootstrap Estimates of h -step Transition Probabilities with $B = 500000$	113

LIST OF FIGURES

Figure 4.1	Context Tree for a VLMC of Order 3	32
Figure 4.2	Partition of Triplets for SMM of Order 3	32
Figure 4.3	Histories with higher frequency in the reference sequence for each virus	54
Figure 5.1	Box plot for total time elapsed for 100 replications, using different algorithms	89
Figure 5.2	Primal and Dual residuals for Kullback-Leibler Divergence, in \log_{10} scale	90
Figure 5.3	Duality gap for DPD, in \log_{10} scale	91

CHAPTER

1

INTRODUCTION

Discrete time Markov models are extremely popular for analyzing a categorical time series due to their wide applicability. For a Markov chain of order one, the evolution of the process at a future time point is independent on past states given the present state. The evolution of this chain is characterised by the transition probabilities from one state to another. These models are named after the Russian probabilist Andrey Andreyevich Markov, who first introduced this concept back in 1906. Till then, Markov models are useful in modelling many real life problems including complex DNA or RNA sequences, data compression, text classification, network analysis and spatial data. Theoretical properties of Markov chains have been well studied throughout the years which confirm the asymptotic consistency of the functionals of data points under certain assumptions.

Although the initial developments were limited to Markov models of order one, scientists have observed that use of higher-order Markov models can capture more complex dependence of a categorical time series. However, with increasing order, the complexity of the model also increases in terms of number of parameters. Meaningful inference is not possible for higher-order Markov

chains unless we assume some simplified higher order models, for example the model proposed by Raftery (1985). Several other dimension reduction techniques have been adapted for modelling such higher order Markov chains. The most popular one is the use of variable length Markov chains (VLMC) which was first introduced by Rissanen (1983). Further theoretical and computational advancements of VLMC have been developed by Bühlmann et al. (1999), Bühlmann (2000), Begleiter et al. (2004), Galves et al. (2012), Belloni and Oliveira (2017) and Cénac et al. (2018). A more general parsimonious modeling approach is given by Sparse Markov Models (SMMs), where all possible histories of order m are partitioned such that the transition probability vectors are identical for the histories belonging to any particular group, have been introduced by Garcia et al. (2011). Bayesian approaches for fitting SMM have been outlined by Jääskinen et al. (2014), Xiong et al. (2016) and Bennett et al. (2022). In this thesis, we completely focus on sparse Markov models, developing different algorithms for fitting SMM, prediction, large sample properties, simulation studies and demonstrate our methods with real data examples.

In Chapter 2, we give a thorough review of the large sample properties of Markov chains, with a focus on the stationarity, ergodicity and development of Markov chain central limit theorems (CLTs). We discuss the results for both general and countable state space Markov chains, using some suitable examples. This chapter works as a preamble to the Chapter 3, where we extend the large sample results for general higher order Markov chains. The major development in this chapter is that the asymptotic results consider a set-up which allows the order of the chain m to grow with increasing length n of the chain under certain assumptions. A small example of VLMC in binary chains demonstrate how the imposed conditions simplify in practical problems, helps us determining the rate of growth of m with the sample size n .

In Chapter 4, we address the major problem with fitting higher order Markov models is the exponentially growing number of parameters in the model order. We develop an elegant method of fitting SMMs based on convex clustering algorithms. In this approach, we minimize a convex objective function which is the sum of two terms - one of them is a smooth error term, squared error loss to be specific; the other one is a non-differentiable penalty function which involves a regularization tuning parameter. This regularization parameter is selected using the BIC criterion.

Theoretical results establish model selection consistency of our method for large sample size. Extensive simulation results under different set-ups are presented to study finite sample performance of the method. We demonstrate our methodology in modelling a problem of virus classification from the collected samples, where the RNA samples are partially available.

In Chapter 5, we develop a new direction of convex clustering algorithms, where the smooth loss function belongs to a more general class of divergence measure, namely the density power divergence (DPD). This class includes widely used Kullback-Leibler divergence and the squared error loss which was used in the Chapter 4. In such problems, we view the clustering problem as a constrained convex optimization problem. Then we use the variable splitting methods like alternating direction method of multipliers (ADMM) and alternating minimization algorithm (AMA) for clustering the empirical transition probability vectors for the m^{th} order histories. We provide theoretical results which enable us to find a range of the regularization parameter for which the true underlying clusters can be identified when we use the DPD loss. We also propose a more relaxed method of convex clustering, namely SR2C2, which performs comparably with the traditional methods, but in much less time. To compare the time complexity and the model performances for different algorithms, extensive simulation studies have been conducted.

A most important problem in analyzing a categorical time series is prediction. For Markov models of order one, it is relatively easier to predict the state X_{n+h} given the observations X_1, \dots, X_n , predicting the most likely state corresponding to the highest h -step transition probability. For higher order Markov models including SMM, computing such transition probabilities will require lot of computational effort, as one needs to sum over the all possible paths in between. To overcome that problem, we propose a bootstrap based prediction algorithm in Chapter 6, where the bootstrap samples generate the future paths from the fitted SMM, and we predict the most likely state at X_{n+h} as the one which appears most of the time in the bootstrap samples. Extensive simulation studies have been performed to assess how many bootstrap samples are enough for producing reliable estimators of the transition probabilities. We also propose a computationally efficient method of constructing simultaneous $100(1 - \alpha)\%$ prediction sets for the future observations $(X_{n+1}, \dots, X_{n+h})$ based on an anomaly scoring method. This would help one to determine whether an h -tuple

belongs to the prediction set or not. The anomaly score associated to each h -tuple is important in finding anomalies in sequence data. We demonstrate our method in detecting anomalous genes of a particular strain of *Helicobacter Pylori* bacteria.

CHAPTER

2

LITERATURE REVIEW: LARGE SAMPLE PROPERTIES OF MARKOV CHAINS

2.1 Introduction

A stochastic process is defined as a collection $\Phi = \{X_t : t \in \mathcal{T}\}$ of random variables defined on a common probability space, where \mathcal{T} is any index set. Typically, for discrete time stochastic processes, $\mathcal{T} = \{0, 1, 2, \dots\}$. The random variable X_t takes values in a set \mathcal{X} , called the state space. A discrete time Markov chain is a stochastic process $\Phi = \{X_t : t = 0, 1, 2, \dots\}$ such that the evolution of the process at a future time point is independent on past states given the present state. This process has been named after Russian probabilist Andrey Andreyevich Markov, who first summarized the idea in Markov (1906). Most of his literary works on the development of Markov chains were discussed in Basharin et al. (2004).

Analyzing and understanding methodology related to discrete time Markov chains are of im-

mense importance due to their wide applicability. In time series analysis, most known autoregressive or moving average models are examples of Markov processes. Applications of Markov models in spatial data have been explored by Besag (1974, 1978); while Rue and Held (2005) deal with Markov random field models. In DNA sequencing, Kimura (1981), Braun and Muller (1998), and Durrett (2008) use Markov chains to capture complex DNA patterns. Markov models may be used for model fitting in many other areas of potential interests as well, including network analysis in engineering and pattern recognition in text analysis. Hence, a discussion on the available methodology for estimating the unknown parameters as well as large sample behavior of Markov chains highlights statistical inference for various real-life examples.

Probabilities for transitions from one state to the next play a key role in determining the properties of a Markov chain. When the state space \mathcal{X} is countable, we can write $P(X_{n+1} = x_{n+1} | X_n = x_n, X_{n-1} = x_{n-1}, \dots, X_0 = x_0) = P(X_{n+1} = x_{n+1} | X_n = x_n)$ for x_0, x_1, \dots, x_{n+1} taking values in \mathcal{X} . Suppose, $P(X_{n+1} = y | X_n = z) = P(X_1 = y | X_0 = z)$ for all integers $n \geq 1$ and for all $(y, z) \in \mathcal{X}^2$. This type of process is called a time homogeneous Markov chain, where the transition probabilities do not depend on the time n . Throughout this article, we will consider time homogeneous Markov chains only, unless it is otherwise noted. For countable \mathcal{X} , we can organize the one-step transition probability from one state to the next in a matrix. If we denote the states in \mathcal{X} as $\{1, 2, 3, \dots\}$ then let $P_{ij} = P(X_1 = j | X_0 = i)$. If $|\mathcal{X}| = d < \infty$, then $P = ((P_{ij}))$ is a $d \times d$ matrix, where the sum of the entries in each row is 1. If \mathcal{X} is countably infinite, then P is an infinite dimensional matrix, again with each row sum being 1. The matrix P is called the transition probability matrix. The n -step transition probability from state i to state j is the $(i, j)^{th}$ entry of the matrix P^n , and is denoted by $P_{ij}^{(n)} = P(X_n = j | X_0 = i)$.

For a countable state space, we formally define a discrete time Markov chain of order 1 as follows
For a countable state space, we formally define a discrete time Markov chain of order 1 as follows

Definition 1 Suppose $\Phi = \{X_t : t = 0, 1, 2, \dots\}$ is a stochastic process taking values in a countable state space \mathcal{X} with $d = |\mathcal{X}|$. Let $P = ((P_{ij}))$ be a $d \times d$ dimensional matrix such that $\sum_j p_{ij} = 1$ and

$$Pr(X_{n+1} = j | X_n = i, X_{n-1} = i_1, \dots, X_0 = i_n) = Pr(X_{n+1} = j | X_n = i) = P_{ij}.$$

Then we call Φ a time-homogeneous discrete time Markov Chain over a countable state space.

Things become complicated when \mathcal{X} is a general state space and not necessarily countable. The notion of a transition from one state to the next may not work in the same sense as in the countable case and we cannot construct a transition probability matrix as before. Instead, we introduce transition probability kernels, defined in Meyn and Tweedie (2012) as follows.

Definition 2 Let $\mathcal{B}(\mathcal{X})$ be a countably generated σ -field over \mathcal{X} , more precisely a Borel σ -field if \mathcal{X} is a topological space. We call $P = \{P(x, A), x \in \mathcal{X}, A \in \mathcal{B}(\mathcal{X})\}$ a transition probability kernel or Markov transition function if

- (a) for each $A \in \mathcal{B}(\mathcal{X})$, $P(\cdot, A)$ is non-negative measurable function on \mathcal{X} ;
- (b) for each $x \in \mathcal{X}$, $P(x, \cdot)$ is a probability measure on $\mathcal{B}(\mathcal{X})$.

In this case, the n -step transition kernel is computed recursively as

$$P^{(n)}(x, A) = \int_{\mathcal{X}} P(x, dy) P^{(n-1)}(y, A), \quad x \in \mathcal{X}, A \in \mathcal{B}(\mathcal{X}).$$

In the countable case, the transition kernel simplifies to a transition probability matrix in a manner such that for each state i , we have a discrete probability distribution on \mathcal{X} according to the i^{th} row of the transition matrix with P_{ij} as the transition probability from state i to state j . From now on, we let P denote the transition probability matrix if \mathcal{X} is countable, and transition probability kernel if \mathcal{X} is a general state space. It is obvious that any definitions or results related to a general state space are also valid for countable state space Markov chains as well, because a countable state space is a subcategory of a general state space. However, theoretical developments in countable or finite state space models are less abstract and more useful in real-life examples compared to the general state space case.

The remainder of the chapter is organized as follows. In Section 2.2, we introduce the important definitions and notation that will be used throughout the paper. Sections 2.3 and 2.4 deal with two major concepts related to large sample results, namely the stationary distribution and ergodicity,

respectively. The formulation of central limit theorems using various approaches are presented in Section 2.5. We conclude the report in Section 2.6 with some future research ideas on this topic.

2.2 Definitions and Notations

Before going into the intriguing theory of Markov chains, we need to discuss some preliminary ideas and definitions. Just as we have introduced the notions of transition probability matrices and transition probability kernels, we will discuss the definitions separately for countable and general state spaces, respectively.

2.2.1 Countable State Spaces

Let $\Phi = \{X_t : t = 0, 1, 2, \dots\}$ be a Markov chain over countable state space \mathcal{X} with transition probability matrix P . Define, for $j \in \mathcal{X}$,

$$\eta_j = \sum_{n=1}^{\infty} \mathcal{I}\{X_n = j\}; \quad \tau_j = \min\{n \geq 1 : X_n = j\}; \quad \sigma_j = \min\{n \geq 0 : X_n = j\}.$$

If $X_0 = j$, η_j is the number of times the chain returns to the state j , and τ_j is the first return time. Also, σ_j is the first hitting time to state j . The k^{th} return to j is defined successively as $\tau_j(k) = \min\{n > \tau_j(k-1) : X_n = j\}$, $k = 2, 3, \dots$, where $\tau_j(1) = \tau_j$. Now, for $i, j \in \mathcal{X}$, define

$$\begin{aligned} U_{ij} &= \sum_{n=1}^{\infty} P_{ij}^{(n)} = \sum_{n=1}^{\infty} E[\mathcal{I}\{X_n = j\} | X_0 = i] = E\left[\sum_{n=1}^{\infty} \mathcal{I}\{X_n = j\} | X_0 = i\right] \\ &= E[\eta_j | X_0 = i] = E_i[\eta_j], \\ L_{ij} &= P(\tau_j < \infty | X_0 = i) = P_i(\tau_j < \infty) = P_i(\Phi \text{ ever enters state } j). \end{aligned}$$

We say that a state i leads to a state j if $L_{ij} > 0$, in other words if $P_{ij}^{(n)} > 0$ for some $n \geq 1$. We denote this as $i \rightarrow j$. We say that the state i communicates with the state j if $i \rightarrow j$ and $j \rightarrow i$, written as $i \leftrightarrow j$. This relation ' \leftrightarrow ' is an equivalence, leading to formation of an equivalence class $C(i) = \{j : i \leftrightarrow j\}$ for $i \in \mathcal{X}$ with $i \in C(i)$. The state space \mathcal{X} is comprised of the union of such equivalence classes. The period of the class $C(i)$ is defined as $d = \gcd\{n : P_{ii}^{(n)} > 0\}$, and the period is

same for every state $j \in C(i)$. Now, if there exists only one such equivalence class, i.e. if $C(i) = \mathcal{X}$ for $i \in \mathcal{X}$, then we say that the Markov chain Φ is *irreducible*, and every state communicates with every other state. An irreducible chain with period 1 is called *aperiodic*.

A state j is called *recurrent* if $L_{jj} = 1$, otherwise it is called *transient*. L_{ii} is not always easy to compute, however there exists an useful condition that is described in Ross (1996) or Karlin and Taylor (1975). The state j is recurrent if and only if $\sum_{n=1}^{\infty} P_{jj}^{(n)} U_{jj} = \infty$. Thus the main characteristic of a recurrent state is that starting from that state, the chain will eventually return to that state with probability 1, and thus the expected number of returns is ∞ . If the expected number of returns of the state j is finite, then subsequently $L_{jj} < 1$, i.e. j is a transient state. We say that a recurrent state j is positive recurrent if $E[\tau_j | X_0 = j] = E_j[\tau_j] < \infty$, otherwise it is called null recurrent. Throughout the paper, we define $E[Z | X_0 = j] = E_j[Z]$, where Z is a random variable and $j \in \mathcal{X}$.

A Markov chain over a countable state space is called **ergodic** if it is aperiodic, irreducible and positive recurrent. The limit theorems in later sections are constructed mostly for ergodic chains. We will formally define ergodicity in section 2.4.

2.2.2 General State Spaces

Let $\Phi = \{X_t : t = 0, 1, 2, \dots\}$ be a Markov chain over a general state space \mathcal{X} with transition probability kernel P . Take $x \in \mathcal{X}$ and $A \in \mathcal{B}(\mathcal{X})$. Consistent with the terms defined in section 2.2.1, we define the following terms, having similar meaning.

$$\eta_A = \sum_{n=1}^{\infty} \mathcal{I}\{X_n \in A\}; \quad \tau_A = \min\{n \geq 1 : X_n \in A\}; \quad \sigma_A = \min\{n \geq 0 : X_n \in A\}.$$

$$U(x, A) = \sum_{n=1}^{\infty} P^{(n)}(x, A) = E_x[\eta_A]; \quad L(x, A) = P_x(\tau_A < \infty) = P_x(\Phi \text{ ever enters set } A).$$

The set A is called *recurrent* if $E_x[\eta_A] = \infty$ for all $x \in A$, and called *uniformly transient* if $E_x[\eta_A] \leq M < \infty$ for all $x \in A$. Denote $Q(x, A) = P_x(\Phi \in A \text{ infinitely often (i.o.)})$. The set A is called *Harris recurrent* if $Q(x, A) = 1$, in other words if $P_x(\eta_A = \infty) = 1$ (see Meyn and Tweedie (2012)). Alternatively, A is Harris recurrent if $L(x, A) = 1$ for all $x \in A$; this can be established using the fact that $L(x, A) = 1 \implies Q(x, A) = 1$ for all $x \in A$.

So far, the extension of the definitions of recurrence and transience in the general set-up are analogous to the countable case. However, the concept of irreducibility is complicated in the general set-up. Here we cannot define two-way communication between states x and y in \mathcal{X} . For instance, the mass of the distribution $P(x, \cdot)$ is 0 everywhere if $P(x, \cdot)$ is continuous on \mathcal{X} . To overcome this difficulty, we introduce the notion of φ irreducibility. We call $\Phi = \{X_t : t = 0, 1, 2, \dots\}$ φ -irreducible if there exists a probability measure φ on $\mathcal{B}(\mathcal{X})$ such that whenever $\varphi(A) > 0$ for some $A \in \mathcal{B}(\mathcal{X})$, we have $L(x, A) > 0$ for all $x \in \mathcal{X}$.

Many of the basic concepts of irreducibility in general state spaces were developed by German probabilist Wolfgang Doeblin in Doeblin (1937). Later Doob (1953), Harris (1955), Orey et al. (1959), Orey (1971), Tweedie (1974a), Tweedie (1974b), and Nummelin (2004) have enriched the literature.

Proposition 4.2.2 of Meyn and Tweedie (2012) shows that if the Markov chain Φ is φ -irreducible for some probability measure φ , then there exists a measure ψ satisfying the following properties:

1. Φ is ψ -irreducible.
2. For any other measure φ' , the chain is φ' -irreducible iff $\psi \succ \varphi'$, i.e. if $\psi(A) = 0$ for some $A \in \mathcal{B}(\mathcal{X})$, then $\varphi'(A) = 0$.
3. If $\psi(A) = 0$, then $\psi\{y : L(y, A) > 0\} = 0$.

We call this probability measure ψ a *maximal irreducibility* measure, which was first introduced in Tweedie (1974a). Subsequently the chain Φ is called ψ -irreducible. Define $\mathcal{B}^+(\mathcal{X}) = \{A \in \mathcal{B}(\mathcal{X}) : \psi(A) > 0\}$. The whole Markov chain Φ is called *Harris recurrent* if it is ψ -irreducible and every set $A \in \mathcal{B}^+(\mathcal{X})$ is Harris recurrent.

Note that if \mathcal{X} is countable, it is a special case of the general state space. For example, the above notion of existence of a maximal irreducible measure is equivalent to the chain being irreducible in the countable case. Similarly, we can simplify the results for a general state space in the countable case. A set $\alpha \in \mathcal{B}(\mathcal{X})$ is called an atom if there exists a measure ν on $\mathcal{B}(\mathcal{X})$ such that $P(x, A) = \nu(A)$, for all $x \in \alpha$ and $A \in \mathcal{B}(\mathcal{X})$. This α is called an *accessible atom* if Φ is ψ -irreducible and $\psi(\alpha) > 0$. Many of the results for a general state spaces are analogous to the countable case if \mathcal{X} contains an

atom. Clearly, every single point in \mathcal{X} is always an atom. If \mathcal{X} is countable and the corresponding chain Φ is irreducible, every point is an accessible atom.

The assumptions of recurrence, irreducibility, and ψ -irreducibility discussed in this section will be very crucial for most of the limiting results presented in the later sections.

2.3 The Stationary Distribution

So far, we have discussed some preliminary concepts related to the construction of discrete time Markov chains. Next, we focus on the evolution of a Markov process in the long run. Statisticians are often interested in how stable a Markov chain is as n increases. Of course one can achieve maximum stability if the variation of the random variables $\{X_n\}$ remains constant over time. In other words, stability can be achieved if the distribution of X_n remains invariant over n . This leads to the formulation of a *stationary distribution*, which is invariant under translation through time. We will first discuss the definition and properties of the stationary distribution for countable state spaces and then extend it to general state spaces.

Let P be the transition probability matrix for the Markov chain $\Phi = \{X_t : t = 0, 1, 2, \dots\}$ over state space \mathcal{X} , and label the states as $1, 2, 3, \dots$ and so on. Consider a distribution π on \mathcal{X} represented as a row vector $\pi = (\pi_1, \pi_2, \dots)$, such that $\sum_{j \in \mathcal{X}} \pi_j = 1$. If a random variable $Y \sim \pi$, then $P(Y = j) = \pi_j$, $j \in \mathcal{X}$. We call a probability vector π a stationary distribution if it satisfies the following relation:

$$\pi = \pi P. \tag{2.3.1}$$

Suppose $X_0 \sim \pi$. Then $P(X_1 = j) = \sum_{i \in \mathcal{X}} P(X_1 = j | X_0 = i)P(X_0 = i) = \sum_{i \in \mathcal{X}} \pi_i P_{ij} = \pi_j$, using (2.3.1). So, $X_1 \sim \pi$, and by a similar argument, $X_n \sim \pi$ for all $n \geq 0$.

Now the question of interest is under what conditions does a stationary distribution exist? Also, what happens when the distribution of X_0 is not stationary? Obviously in that case we cannot expect that the distribution of each of the random variable X_n will be the same. The following theorem (see Karlin and Taylor (1975) or Ross (1996)) establishes existence of a stationary distribution and of

long run stationarity of a Markov chain under ergodicity.

Theorem 2.3.1 *Let $\Phi = \{X_t : t = 0, 1, 2, \dots\}$ be an aperiodic, irreducible and positive recurrent Markov chain over countable state space $\mathcal{X} = \{1, 2, 3, \dots\}$ with transition probability matrix P . Then for any $i, j \in \mathcal{X}$,*

$$\lim_{n \rightarrow \infty} P_{ij}^{(n)} = \pi_j, \quad (2.3.2)$$

where $\pi = (\pi_1, \pi_2, \dots)$ is the unique stationary distribution, satisfying $\pi = \pi P$.

This result implies that the distribution of X_n converges to the stationary distribution if $n \rightarrow \infty$, irrespective of the initial state. If we drop the aperiodic condition, a unique stationary distribution still exists, however the limiting condition in (2.3.2) is not satisfied if period $d \geq 2$. This is obvious, since for any $n \in \mathbb{N}$ and $j \in \mathcal{X}$, $P_{jj}^{(n)} > 0$ only if $n = kd$ for some $k \in \mathbb{N}$ and $P_{jj}^{(n)} = 0$ if n is not divisible by d . Ross (1996) showed that for an irreducible, positive recurrent and periodic Markov chain of period d with transition probability matrix P , there exists a unique stationary distribution π satisfying $\pi = \pi P$, and for any $j \in \mathcal{X}$, $\lim_{n \rightarrow \infty} P_{jj}^{(nd)} = d\pi_j$.

For general state space \mathcal{X} , we say that a probability measure over $\mathcal{B}(\mathcal{X})$ with transition probability kernel P is *invariant* if

$$\pi(A) = \int_{\mathcal{X}} \pi(dx)P(x, A); \quad A \in \mathcal{B}(\mathcal{X}).$$

We call a Markov chain Φ *positive* if such an invariant or stationary probability measure exists. As the conditions under which the stationary probability measure π exists and for which the transition kernel $P^{(n)}$ converges to π in distribution have been thoroughly discussed in Meyn and Tweedie (2012), we are not going to discuss them for brevity. We call Φ *Harris ergodic* if it is positive, irreducible and Harris recurrent. However, if Φ is ψ -irreducible and possesses an accessible atom $\alpha \in \mathcal{B}^+(\mathcal{X})$, then Φ is positive recurrent if and only if $E_\alpha(\tau_\alpha) < \infty$ and a stationary probability measure π exists with $\pi(\alpha) = 1/E_\alpha(\tau_\alpha)$. Now this result, when simplified for a countable state space \mathcal{X} , implies that the stationary probability of the j^{th} state in \mathcal{X} for an ergodic Markov chain is given by $\pi_j = 1/E_j(\tau_j)$. We will use this fact often while constructing central limit theorems.

2.4 Ergodicity

In the previous section, we have discussed that under some conditions, the transition probability kernel $P^{(n)}$ converges to stationary distribution π as $n \rightarrow \infty$. This property is called ‘ergodicity.’ Now we will discuss different rates of convergence to the stationary distribution, which helps us to build the foundation for central limit theorems. In this section, most of the definitions and results are presented for a general state space \mathcal{X} along with the notion of a general transition probability kernel P . However, Kendall (1959) and Vere-Jones (1962) have proved the following result specifically for countable \mathcal{X} .

Theorem 2.4.1 *Let $\Phi = \{X_t : t = 0, 1, 2, \dots\}$ be an aperiodic, irreducible and positive recurrent Markov chain over countable state space \mathcal{X} with transition probability matrix P . If there exists a state $i \in \mathcal{X}$ with $|P_{ii}^{(n)} - \pi_i| = \mathcal{O}(\rho_i^n)$ for some $\rho_i < 1$, then there exists $\rho < 1$ such that for all $k, j \in \mathcal{X}$, $|P_{kj}^{(n)} - \pi_j| = \mathcal{O}(\rho^n)$, where π is the stationary distribution.*

Thus, if the conditions of the above theorem hold, then $P_{ij}^{(n)}$ converges to π_j geometrically with uniform rate ρ for all $j \in \mathcal{X}$. Now, if we want to extend the idea of convergence to general state spaces, we need a measure of the difference between the two measures $P^{(n)}(x, \cdot)$ and π for $x \in \mathcal{X}$. For that purpose, we use the total variation norm between two measures μ and ν on the same σ -field \mathcal{B} , defined as $\|\mu - \nu\| = \sup_{A \in \mathcal{B}} |\mu(A) - \nu(A)|$. Mathematically, we call a Markov chain Φ on state space \mathcal{X} ergodic if for all $x \in \mathcal{X}$,

$$\lim_{n \rightarrow \infty} \|P^{(n)}(x, \cdot) - \pi(\cdot)\| = 0. \quad (2.4.3)$$

In most cases we deal with the following type of bounds

$$\|P^{(n)}(x, \cdot) - \pi(\cdot)\| \leq M(x)\gamma(n), \quad (2.4.4)$$

where M is some non-negative function and γ is a non-negative and non-increasing function on \mathbb{Z}_+ . If $\gamma(n) = t^n$ for some $t < 1$, then we call the chain **geometrically ergodic**. If $\gamma(n)$ is uniformly bounded by some real number $M_0 < \infty$ for all $x \in \mathcal{X}$, then we call the chain **uniformly ergodic**. If $\gamma(n) = n^{-m}$ for some $m \in \mathbb{N}$, then the rate of convergence is polynomial and hence we call the chain

polynomially ergodic. In practice, establishing the relation (2.4.4) is very difficult for general state spaces. In the 21st century, significant research has been conducted to find sufficient conditions for which the relation (2.4.4) holds for some appropriate function γ . Jones et al. (2004) has accumulated some of these conditions, which we are going to discuss, along with their implications.

Let $V : \mathcal{X} \rightarrow \mathbb{R}$ be a real-valued function on \mathcal{X} . Define

$$PV(x) = \int V(y)P(x, dy); \quad \Delta V(x) = PV(x) - V(x), \quad x \in \mathcal{X}.$$

Suppose that for a function $V : \mathcal{X} \rightarrow [1, \infty)$, there exist constants $0 < d < 1$ and $b < \infty$ such that

$$\Delta V(x) \leq -dV(x) + b\mathcal{I}(x \in C), \quad x \in \mathcal{X} \tag{2.4.5}$$

for some set $C \in \mathcal{X}$, where \mathcal{I} is the indicator function. These kind of relations are called ‘drift conditions’. Apparently, equation (2.4.5) looks like a contraction mapping, which eventually plays a key role in determining the rate of convergence. Tweedie (1975) first used the drift condition (2.4.5) to prove geometric ergodicity. In that case $\gamma(n) = \rho^n$ for some $\rho \in (1 - d, 1)$; details may be found in Meyn and Tweedie (2012).

Some other approaches have also been taken to establish geometric ergodicity. For example, Nummelin and Tweedie (1978) have shown that for a Harris ergodic Markov chain Φ , if there exists an atom α with $\pi(\alpha) > 0$ and $|P^{(n)}(\alpha, \alpha) - \pi(\alpha)| = \mathcal{O}(\rho_\alpha^n)$, then the chain is geometrically ergodic. This is an extension of Theorem (2.4.1), which was specifically designed for countable state spaces. Geometric ergodicity was characterized in terms of hitting time distributions by Nummelin and Tuominen (1982), where they came up with a set of equivalent conditions to the drift condition (2.4.5).

The research on ergodic theory is not limited to geometric ergodicity. In a groundbreaking work, Tuominen and Tweedie (1994) have studied subgeometric convergence (i.e. slower than geometric) of ergodic chains using a set of drift conditions. Using these ideas, Jarner et al. (2002) used a drift condition similar to (2.4.5). Suppose that for a function $V : \mathcal{X} \rightarrow [1, \infty)$, there exist constants $d > 0$,

$b < \infty$ and $0 \leq \tau < 1$ such that

$$\Delta V(x) \leq -d[V(x)]^\tau + b\mathcal{I}(x \in C), \quad x \in \mathcal{X} \quad (2.4.6)$$

for some set C . Jarner et al. (2002) proved that if (2.4.6) holds then the Markov chain Φ is polynomially ergodic of degree $\frac{\tau}{1-\tau}$. Douc et al. (2004) extended some conditions of Jarner et al. (2002) to achieve a simpler one which is conducive to finding a subgeometric convergence rate.

The ideas of ergodicity, rate of convergence and drift conditions will be carried forward to prove central limit theorems for Markov chains in a general state space next.

2.5 Central Limit Theorems

Central limit theorems for dependent random variables have been developed in the last century. The CLT results are important for determining large sample distributions of sample averages of functions of discrete random variables. Meaningful statistical inference may be made using the results. The major goal of this section is to discuss the needed assumptions for CLTs and the types of functions of discrete random variables for which they hold. Before going into the main results, we define some relevant notation.

Suppose $f : \mathcal{X} \rightarrow \mathbb{R}$ is a Borel measurable function over the state space \mathcal{X} , and let $\Phi = \{X_t : t = 0, 1, 2, \dots\}$ be a Markov chain on \mathcal{X} . Define $S_n(f) = \sum_{i=1}^n f(X_i)$, and $\bar{f}_n = S_n(f)/n$. If Φ has stationary distribution π , define $E_\pi f = \int f(x)\pi(dx)$ for general \mathcal{X} and $E_\pi f = \sum_{i \in \mathcal{X}} \pi_i f(i)$ for countable \mathcal{X} . Here, we present results that indicate that under suitable conditions, the asymptotic distribution of $\sqrt{n}(\bar{f}_n - E_\pi f)$ is $\mathcal{N}(0, \sigma_f^2)$, where σ_f^2 is some quantity based on the function f . Interestingly, different approaches lead to seemingly different expressions for σ_f^2 . However in all cases we deduce that σ_f^2 may be finally written as

$$\sigma_f^2 = \text{var}_\pi\{f(X_0)\} + 2 \sum_{i=1}^{\infty} \text{cov}_\pi\{f(X_0), f(X_i)\}.$$

if the sum exists. We will present the CLT related results using different approaches, and how they

are connected with one another.

2.5.1 CLT for a Countable State Space

One popular approach to prove a limit theorem for a Markov chain with a countable state space is to break the sum $S_n(f)$ into i.i.d blocks and apply CLT for iid random variables to the blocks. For that, we fix a state $j \in \mathcal{X}$ and look at the successive visits to j in X_1, \dots, X_n . Let $l_n(j) = \sum_{k=1}^n \mathcal{I}(X_k = j)$ be the number of returns to j by time n . Let $s_k^{(j)}(f) = \sum_{i=\tau_j(k)+1}^{\tau_j(k+1)} f(X_i)$ for $k \geq 1$. By the strong Markov property, $s_k^{(j)}(f)$ are iid random variables, as the Markov process starts afresh after each visit to j . Let $s_0^{(j)}(f) = \sum_{i=1}^{\tau_j} f(X_i)$. Observe that, $s_1^{(j)}(f)$ is identically distributed with $s_0^{(j)}(f)$ given $X_0 = j$, since $s_k^{(j)}(f)$ is the sum of $f(X_i)$'s between two successive visits. Thus $E[s_1^{(j)}(f)] = E_j[s_0^{(j)}(f)]$. We have already discussed that $E_j[\tau_j] = 1/\pi_j$. Define $Z_0^{(j)}(f) = s_0^{(j)}(f) - \tau_j E_\pi f$ and let $\sigma_j^2 = E_j[(Z_0^{(j)}(f))^2]$. Lai (1967) have shown that for an aperiodic, irreducible and positive recurrent chain, $\pi_j E_j[s_0^{(j)}(f)]$ is independent of j , and equals $E_\pi f$. This immediately produces the following CLT, which was proved by Doeblin (1937).

Theorem 2.5.1 *Let $\Phi = \{X_t : t = 0, 1, 2, \dots\}$ be an irreducible, aperiodic and positive recurrent Markov chain on countable state space \mathcal{X} . Suppose $\sigma_j^2 < \infty$ and define $B_j = \pi_j \sigma_j^2$. Then we have*

$$\sqrt{n}(\bar{f}_n - E_\pi f) \xrightarrow{d} \mathcal{N}(0, B_j).$$

provided $B_j > 0$. Moreover B_j is independent of the index j with common value σ_f^2 .

A possible extension of Theorem (2.5.1) to general state spaces will be analogous if the state space \mathcal{X} possesses an accessible atom α . Suppose, $s_0^{(\alpha)}(f) = \sum_{k=1}^{\tau_\alpha} f(X_k)$ for some measurable function f . The following theorem was proved in Orey (1971) and Meyn and Tweedie (2012).

Theorem 2.5.2 *Suppose $\Phi = \{X_t : t = 0, 1, 2, \dots\}$ is an Harris ergodic Markov chain on measurable space $(\mathcal{X}, \mathcal{B}(\mathcal{X}))$ with stationary distribution π . Suppose there exists an accessible atom $\alpha \in \mathcal{B}^+(\mathcal{X})$ such that $E_\alpha[(s_0^{(\alpha)}(|f|))^2] < \infty$ and $E_\alpha(\tau_\alpha^2) < \infty$. Then*

$$\sqrt{n}(\bar{f}_n - E_\pi f) \xrightarrow{d} \mathcal{N}(0, \sigma_f^2),$$

and σ_f^2 can be alternatively represented as

$$\sigma_f^2 = \pi(\alpha) E_\alpha \left[\left(\sum_{k=1}^{\tau_\alpha} [f(X_k) - E_\pi(f)] \right)^2 \right].$$

In the next section we will present CLT in general state spaces not necessarily possessing an accessible atom.

2.5.2 CLT in a General State Spaces

In section (2.4), we have shown how drift conditions are useful in determining the convergence rate for an ergodic chain. Subsequently, these conditions may be utilized to prove CLT for a class of functions, as noted by Jones et al. (2004).

Theorem 2.5.3 *Let Φ be a Harris ergodic Markov chain on a general state space \mathcal{X} having stationary distribution π . Suppose $f : \mathcal{X} \rightarrow \mathbb{R}$ and assume that one of the following conditions hold:*

1. *The drift condition (2.4.5) holds with $f^2(x) \leq V(x)$ for all $x \in \mathcal{X}$.*
2. *The drift condition (2.4.6) holds and $|f(x)| \leq V(x)^{\tau+\eta-1}$ for all $x \in \mathcal{X}$ where $1 - \tau \leq \eta \leq 1$ and $E_\pi V^{2\eta} < \infty$.*

Then $\sigma_f^2 \in [0, \infty)$ and if $\sigma_f^2 > 0$, then for any initial distribution,

$$\sqrt{n}(\bar{f}_n - E_\pi f) \xrightarrow{d} \mathcal{N}(0, \sigma_f^2). \tag{2.5.7}$$

The first part of the theorem was proved in Meyn and Tweedie (2012), and the second part in Jarner et al. (2002). From a practical point of view, several attempts have been made to establish this type of drift condition under the MCMC set up to prove CLT. As references, drift and minorization conditions for a Gibbs sampler have been developed in Roberts and Polson (1994), Rosenthal (1995), and Hobert and Geyer (1998). Similar analysis for the Metropolis Hastings algorithm have been discussed, e.g., by Mengersen et al. (1996), Roberts and Tweedie (1996), and Fort and Moulines (2000), among others.

So far, we have discussed how the drift conditions help us to connect ergodicity and CLT. We now focus on another approach based on mixing conditions to prove CLT. Mixing coefficients give us a quantification of the rate in which the random variables far in the future become asymptotically independent of the present state of a stochastic process. As per our need, we introduce three different mixing conditions, maintaining the same notation and definitions used in Jones et al. (2004).

Suppose $Y = \{Y_t : t = 0, 1, 2, \dots\}$ is a general sequence of random variables on a probability space $(\Omega, \mathcal{F}, \mathcal{P})$, and let $\mathcal{F}_k^m = \sigma(Y_k, \dots, Y_m)$.

Definition 3 *The sequence Y is said to be strongly mixing, or α -mixing if $\alpha(n) \rightarrow 0$ as $n \rightarrow \infty$, where*

$$\alpha(n) = \sup_{k \geq 1} \sup_{A \in \mathcal{F}_1^k, B \in \mathcal{F}_{k+n}^\infty} |\mathcal{P}(A \cap B) - \mathcal{P}(A)\mathcal{P}(B)|.$$

Definition 4 *The sequence Y is said to be asymptotically uncorrelated, or ρ -mixing if $\rho(n) \rightarrow 0$ as $n \rightarrow \infty$, where*

$$\rho(n) = \sup\{\text{corr}(U, V), U \in L_2(\mathcal{F}_1^k), V \in L_2(\mathcal{F}_{k+n}^\infty), k \geq 1\}.$$

Definition 5 *The sequence Y is said to be uniformly mixing, or ϕ -mixing if $\phi(n) \rightarrow 0$ as $n \rightarrow \infty$ where*

$$\phi(n) = \sup_{k \geq 1} \sup_{\substack{A \in \mathcal{F}_1^k, \mathcal{P}(A) > 0, \\ B \in \mathcal{F}_{k+n}^\infty}} |\mathcal{P}(B|A) - \mathcal{P}(B)|.$$

If (2.4.4) holds, i.e. the total variation norm is bounded by $M(x)\gamma(n)$ with $E_\pi M < \infty$, then it can be shown that $\alpha(n) \leq \gamma(n)E_\pi M$, i.e. $\alpha(n) = \mathcal{O}(\gamma(n))$. Hence a Harris ergodic Markov chain Φ satisfying (2.4.4) with $\gamma(n) \rightarrow 0$ as $n \rightarrow \infty$ is strongly mixing. Also, using standard techniques one can prove $4\alpha(n) \leq \rho(n) \leq 2\sqrt{\phi(n)}$. Thus if a sequence of random variables is uniformly mixing or ρ -mixing, then it is strongly mixing too. The dependence between two random variables with lag n in the sequence Y is quantified by the coefficients $\alpha(n)$, $\rho(n)$ or $\phi(n)$. Ibragimov (1975) and Bradley (1986) have shown that Φ is uniformly ergodic if and only if it is uniformly mixing, and in that case $\phi(n) = \mathcal{O}(e^{-\theta n})$ for some $\theta > 0$.

We say that the Markov chain Φ is reversible or satisfies a detailed balance condition if

$$\pi(dx)P(x, dy) = \pi(dy)P(y, dx) \quad \forall x, y \in \mathcal{X}. \quad (2.5.8)$$

The condition (2.5.8) often plays an important role in determining mixing properties. Bradley (1986) proved that if Φ is geometrically ergodic and (2.5.8) holds, then Φ is ρ -mixing, with $\rho(n) = \mathcal{O}(e^{-\theta n})$ for some $\theta > 0$.

Over the years, sufficient conditions for establishing central limit theorems for mixing sequences have been developed. For strongly mixing sequences, these results are discussed in Cogburn (1960), Ibragimov (1962, 1975), Denker et al. (1986) and Doukhan et al. (1994). Since Harris ergodic chains are strongly mixing, the relevant theorems can be applied in ergodic Markov chains. We only need some conditions on the set of functions f and the nature of the ergodicity. Jones et al. (2004) collected all these results from Ibragimov (1962, 1975), Chan and Geyer (1994), citedoukhan1994functional, Tierney (1994), Roberts et al. (1997), and summarized them as follows.

Theorem 2.5.4 *Let Φ be a Harris ergodic Markov chain on a general state space \mathcal{X} with invariant distribution π and let $f : \mathcal{X} \rightarrow \mathbb{R}$ be a Borel function. Assume one of the following conditions:*

1. Φ is polynomially ergodic of order $m > 1$, $E_\pi M < \infty$ and there exists $B < 1$ such that $|f(x)| < B$ almost surely;
2. Φ is polynomially ergodic of order m , $E_\pi M < \infty$ and $E_\pi |f(x)|^{2+\delta} < \infty$ where $m\delta > 2 + \delta$;
3. Φ is geometrically ergodic and $E_\pi [f^2(x)(\log^+ |f(x)|)] < \infty$;
4. Φ is geometrically ergodic, satisfies (2.5.8) and $E_\pi f^2(x) < \infty$; or
5. Φ is uniformly ergodic and $E_\pi f^2(x) < \infty$.

Then for any initial distribution, as $n \rightarrow \infty$, $\sqrt{n}(\bar{f}_n - E_\pi f) \xrightarrow{d} \mathcal{N}(0, \sigma_f^2)$.

We conclude this section with an example, namely a random walk on the positive half line described in Jones et al. (2004) and Meyn and Tweedie (2012). Let $\Phi = \{X_t : t = 0, 1, 2, \dots\}$ be a Markov chain on

$\mathcal{X} = [0, \infty]$ defined by

$$X_{n+1} = (X_n + W_{n+1})^+; \quad n \geq 1$$

with $X_0 = 0$, and W_1, W_2, \dots being iid random variables with cdf Γ . It can be shown that if $E[W_1] < 0$, then the chain is Harris ergodic. Jarner et al. (2002) proved that if $E[(W_1^+)^m] < \infty$ for some $m \geq 2$, then the drift condition (2.4.6) holds with $V(x) = (x+1)^m$, $\tau = (m-1)/m$ and $C = [0, k]$ for some $k \in \mathbb{R}$. Hence, the chain is polynomially ergodic and the CLT in (2.5.7) holds for any function $f : \mathcal{X} \rightarrow \mathbb{R}$ if $|f(x)| \leq (x+1)^{m(\tau+\eta-1)}$ for $x \geq 0$ and $1 - \tau \leq \eta \leq 1$ such that $E_\pi(X+1)^{2m\eta} < \infty$. This and other examples found in Jones et al. (2004) and Meyn and Tweedie (2012) illustrate the applicability of the results discussed in this section.

2.6 Concluding Remarks

In a nutshell, this chapter covers the theoretical developments of the large sample properties of discrete time Markov chains. Note that, finite state sparse Markov models are a special case of a higher order Markov chain, so the theoretical developments on finite state space models will be directly applied to SMM's. A CLT for a triangular array of Markov chains where the order of the chains vary with increasing n would be interesting. Development of some elegant models for fitting SMM's or VLMC's in real life problems will be useful from a practical point of view as well. In the next few chapters, we will discuss the developments in the theory and methods of sparse Markov models.

LARGE SAMPLE PROPERTIES OF SPARSE MARKOV MODELS

3.1 Introduction

Estimation methods and large sample results for Markov chains of order one are very common in practice. However, there are many Markov processes where the present state is dependent on the past m states. If $\Phi = \{X_t : t = 0, 1, 2, \dots\}$ is a sequence of random variables taking values in a finite state space Σ with $|\Sigma| = d$, and for $n \geq m$, $P(X_n = x_n | X_{n-1} = x_{n-1}, \dots, X_0 = x_0) = P(X_n = x_n | X_{n-1} = x_{n-1}, \dots, X_{n-m} = x_{n-m})$, we call Φ a Markov chain of order m . The study of an m^{th} -order Markov chain Φ is not too much different from the order one case, since we can construct a first-order Markov chain from Φ by taking consecutive m -tuples as the states. Let us denote this new Markov chain by $\Phi' = \{Y_n : n = 0, 1, 2, \dots; Y_n = (X_{n+m-1}, \dots, X_n)\}$. The new state space is $\Sigma' = \Sigma^m$, i.e. all possible m -tuples from Σ . Hence the transition matrix P' corresponding to Φ' will be $d^m \times d^m$ dimensional.

In each row of P' , at most d entries can be non-zero, since $P(Y_{n+1} = (j_1, \dots, j_m) | Y_n = (i_1, \dots, i_m)) > 0$ only if $i_k = j_{k+1}$, $k = 1, 2, \dots, m-1$, and in that case $P(Y_{n+1} = (j_1, \dots, j_m) | Y_n = (i_1, \dots, i_m)) = P(X_{n+m} = j_1 | X_{n+m-1} = i_1, \dots, X_n = i_m)$. Hence the total number of entries to be estimated is $d^m(d-1)$, growing exponentially with increasing m . To reduce the number of parameters, Raftery (1985) proposed the following model:

$$P(X_n = j_0 | X_{n-1} = j_1, \dots, X_{n-m} = j_m) = \sum_{i=1}^m \lambda_i q_{j_i j_0},$$

where $\sum_{i=1}^m \lambda_i = 1$ and $Q = ((q_{ij}))$ is a known transition matrix on Σ . The parameters $(\lambda_1, \dots, \lambda_m)$ are estimated by the maximum likelihood method.

To avoid the ‘curse of dimensionality’ in higher-order Markov models, Bühlmann et al. (1999) and Bühlmann (2000) presented some theoretical properties of Variable Length Markov Chains (VLMC), first introduced by Rissanen (1983). In such a set up, $P(X_n = x_n | X_{n-1} = x_{n-1}, X_{n-2} = x_{n-2}, \dots)$ is independent of the past conditional on the past ℓ values. Here ℓ may not be a fixed number, rather a function of the past values $(x_{n-1}, x_{n-2}, \dots)$. The relevant contexts (recent pasts) of variable orders are determined to form a context tree. If the maximum value of ℓ is m , then the VLMC is a Markov chain of order m . However, the number of parameters may reduce significantly, as some of the probability distributions are clumped together. More generally, the overall state space Σ^m of all possible m -tuples can be divided into a partition of size r and we write $\Sigma^m = A_1 \cup \dots \cup A_r$. If (i_1, \dots, i_m) and (j_1, \dots, j_m) both belong to same partition, then for all $x \in \Sigma$, $P(X_n = x | (X_{n-1}, \dots, X_{n-m}) = (i_1, \dots, i_m)) = P(X_n = x | (X_{n-1}, \dots, X_{n-m}) = (j_1, \dots, j_m))$. This generalization was first proposed by Garcia et al. (2011), later extended by Jääskinen et al. (2014) for analyzing sequence data, where each m -tuple is assigned to a particular group, not necessarily following a tree structure. This model is called a Sparse Markov Model (SMM). A Bayesian model fitting approach for SMM is available in Xiong et al. (2016).

As long as Σ is finite, the CLT results or statistical methodologies in Sparse Markov Models will be same as that of Markov chains of order one, with a little more assumptions on the transition probabilities so that the chain remains ergodic. Bühlmann et al. (1999) proved large sample results in the VLMC set up, including a CLT of bootstrap estimators. However, things will be interesting if the order of the Markov chain increases with increasing length of the chain. Thus, the order of the chain can be represented as a function of the chain length n , and we denote the order as m_n .

Clearly, proving CLT type results are more challenging in this set up. In this chapter, we will provide a brief sketch of our main result regarding the CLT of a triangular array of higher-order Markov chains, where the order is increasing with increasing chain length n .

The rest of the chapter is organised as follows. We present our major theoretical results in Section 3.2. In Section 3.3, we simplify the conditions of the results for a binary Markov chain. We conclude this chapter with some important remark in the Section 3.4. The proofs are included in the Appendix A.

3.2 Central Limit Theorem

Consider a triangular array of variable length markov chains as follows. The n^{th} chain is presented as $\Phi_n = \{X_0^{(n)}, \dots, X_n^{(n)}\}$, where $X_j^{(n)} \in \Sigma$. Recall that, Σ is a finite state space with $|\Sigma| = d$. However, the order of the Markov chain Φ_n varies with n , and denote it by m_n . One can easily represent this Markov chain as usual order one chain as follows. Let $Y_t^{(n)} = (X_{t+m_n-1}^{(n)}, \dots, X_t^{(n)})$, $t = 0, 1, \dots, n - m_n + 1$. Then $\Phi'_n = \{Y_t^{(n)}\}$ is an order 1 Markov chain with state space $S_n = \Sigma^{m_n}$. We will work with this Markov chain Φ'_n from now on for our convenience.

Our goal is to establish a CLT type result for some real valued sequence of functions over Σ in this triangular array set up. Before going into the main result, let us define some useful notation. Assume that the chain Φ'_n is aperiodic and irreducible for each n . Let π_n be the stationary probability vector of length d^{m_n} corresponding to the chain Φ'_n . For some state $\alpha \in S_n$, denote $\sigma_\alpha(0) = \inf\{j \geq 0 : Y_j^{(n)} = \alpha\}$. Thus $\sigma_\alpha(0)$ is the first hitting time of α for the chain Φ'_n . In similar fashion, successively define the k^{th} hitting time of α as follows:

$$\sigma_\alpha(k) = \inf\{j \geq \sigma_\alpha(k-1) : Y_j^{(n)} = \alpha\}.$$

Define

$$\ell_{n,\alpha} = \max\{k : \sigma_\alpha(k) \leq n - m_n + 1\} = \sum_{j=0}^{n-m_n+1} \mathcal{I}(Y_j^{(n)} = \alpha) - 1. \quad (3.2.1)$$

Thus $\ell_{n,\alpha}$ is the number of returns of Φ'_n to the state α after the first hitting time.

If the order of the Markov chains in each array is fixed, i.e. $m_n = m$ for some $m \in \mathbb{N}$, then any state $\alpha \in \Sigma^m$ is recurrent, resulting $\ell_{n,\alpha} \rightarrow \infty$ as $n \rightarrow \infty$. However, in our set-up, $m_n \rightarrow \infty$ as $n \rightarrow \infty$. Note that, for $\alpha \in \Sigma^{m_n}$, $E_{\pi_n}(\ell_{n,\alpha}) = (n - m_n + 2)\pi_n(\alpha) - 1$. If there exists a sequence of states $\{\alpha_n : \alpha_n \in \Sigma^{m_n}\}$ such that $n\pi_n(\alpha_n) \rightarrow \infty$ as $n \rightarrow \infty$, then $E_{\pi_n}(\ell_{n,\alpha_n}) \rightarrow \infty$, i.e. the expected number of returns to α_n converges to ∞ .

Next, we present our main result of this chapter, which demonstrate the large sample property of a triangular array of sparse Markov chain.

Theorem 3.2.1 *Consider a triangular array of sparse Markov chains over a finite alphabet Σ of size d as follows. Φ_n , the n^{th} array is represented as $\{X_0^{(n)}, \dots, X_n^{(n)}\}$ and has order m_n . Define $Y_t^{(n)} = (X_{t+m_n-1}^{(n)}, \dots, X_t^{(n)})$, $t = 0, 1, \dots, n - m_n + 1$. Then $\{Y_t^{(n)} : 0 \leq t \leq n - m_n + 1\}$ is an order one MC over Σ^{m_n} . Denote the stationary probability vector of the n^{th} chain as π_n . Suppose, the following assumptions hold.*

- (i) For each $n \in \mathbb{N}$, $\Phi'_n = \{Y_t^{(n)}\}$ is an aperiodic and irreducible MC over Σ^{m_n} .
- (ii) $m_n \rightarrow \infty$ as $n \rightarrow \infty$ with $\frac{m_n}{n} \rightarrow 0$.
- (iii) There exists a sequence of states $\{\alpha_n : \alpha_n \in \Sigma^{m_n}\}$ such that $n\pi_n(\alpha_n) \rightarrow \infty$ as $n \rightarrow \infty$.
- (iv) $\text{Var}_{\pi_n}(l_{n,\alpha_n}/n\pi_n(\alpha_n)) \rightarrow 0$ as $n \rightarrow \infty$.

Consider a sequence of functions $g_n : \Sigma^{m_n} \rightarrow \mathbb{R}$, such that

- (i) For each n , $E_{\alpha_n} \left[\sum_{j=1}^{\tau_{\alpha_n}} \bar{g}_n(Y_j^{(n)}) \right]^2 < \infty$;
- (ii) $\sup_n \frac{E_{\alpha_n} \left[\sum_{j=1}^{\tau_{\alpha_n}} |\bar{g}_n(Y_j^{(n)})| \right]^2}{E_{\alpha_n} \left[\sum_{j=1}^{\tau_{\alpha_n}} \bar{g}_n(Y_j^{(n)}) \right]^2} < \infty$;

where $\bar{g}_n(x) = g_n(x) - E_{\pi_n}(g_n)$. Under the above conditions

$$\frac{1}{\sqrt{n}} \sum_{j=0}^{n-m_n+1} \frac{\bar{g}_n(Y_j^{(n)})}{\sqrt{\frac{E_{\alpha_n}(\tau_{\alpha_n})}{E_{\alpha_n} \left[\sum_{j=1}^{\tau_{\alpha_n}} \bar{g}_n(Y_j^{(n)}) \right]^2}}} \xrightarrow{d} \mathcal{N}(0, 1)$$

The proof mostly follows the proof of the central limit theorem of Harris ergodic Markov chain, described in Meyn and Tweedie (2012). We provide the proof in the Appendix A.

Apart from the above approach to establish the CLT for triangular arrays of variable length Markov chains, we could proceed by imposing some conditions on the mixing coefficients for each array. There have been significant efforts to establish the CLT for the triangular arrays of dependent random variables based on the mixing properties. The most important challenge is to derive sufficient conditions involving the mixing coefficients, not only for a single chain, but for each chain corresponding to a particular array. Withers (1981) established CLT for dependent random variables which satisfy a weak mixing condition, called l -mixing. Eckhard (1996) gives an useful CLT result for triangular arrays where the random variables satisfy the α -mixing property. Suppose $\{Y_{ni}, i = 1, \dots, k_n; n \in \mathbb{N}\}$ is a sequence of α -mixing random variables. Define ${}^n \mathcal{F}_k^m = \sigma(Y_{nk}, \dots, Y_{nm})$. The α -mixing for this triangular array is defined as

$$\alpha(m) = \sup_n \sup_{k:1 \leq k \leq n-m} \{|\mathcal{P}(A \cap B) - \mathcal{P}(A)\mathcal{P}(B)| : A \in {}^n \mathcal{F}_1^k, B \in {}^n \mathcal{F}_{k+m}^n\}.$$

The triangular array is called α -mixing if $\alpha(m) \rightarrow 0$ as $m \rightarrow \infty$. Denote $T_n = \sum_{i=1}^{k_n} Y_{ni}$. Eckhard (1996) derived a CLT result as follows.

Result 3.2.1 *Let $\sum_{k=1}^{\infty} \alpha(k)^{1-2/p} < \infty$ for some $p, 2 < p \leq \infty$. Further, assume that $E Y_{ni} = 0$ and $(E|Y_{ni}|^p)^{1/p} < \infty$. Suppose.*

$$\lim_{n \rightarrow \infty} \sum_{i=1}^{k_n} (E|Y_{ni}|^p I(|Y_{ni}| > \epsilon))^{2/p} = 0$$

for all $\epsilon > 0$. Moreover, assume that

$$\begin{aligned} \lim_{n \rightarrow \infty} E T_n^2 &= 1; \\ \lim_{n \rightarrow \infty} \sum_{i=1}^{k_n} (E|Y_{ni}|^p)^{2/p} &\leq C \end{aligned}$$

for some constant C . Then, as $n \rightarrow \infty$

$$T_n \xrightarrow{d} \mathcal{N}(0, 1).$$

Now we can extend the above result (3.2.1) in our new set-up as well, assuming similar conditions on the mixing coefficients for each of the array. Our result is as follows.

Theorem 3.2.2 *Consider a triangular array of sparse Markov chains over a finite alphabet Σ of size d as follows. The n^{th} array Φ_n is represented as $\{X_1^{(n)}, \dots, X_{N_n}^{(n)}\}$ and has order m_n . Denote $k_n = \lfloor N_n/m_n \rfloor$, where $\lfloor \cdot \rfloor$ is the greatest integer function. Define $W_{t,n} = (X_t^{(n)}, \dots, X_{t+m_n-1}^{(n)})$, $t = 1, \dots, N_n - m_n + 1$; and $Z_{n,j} = (X_{(j-1)m_n+1}^{(n)}, \dots, X_{jm_n}^{(n)})$, $j = 1, \dots, k_n$. Then both $\Phi_{1n} = \{W_{t,n} : 1 \leq t \leq N_n - m_n + 1\}$ and $\Phi_{2n} = \{Z_{n,j} : 1 \leq j \leq k_n\}$ are order one MC over Σ^{m_n} . Denote the transition matrix of dimension $d^{m_n} \times d^{m_n}$ for Φ_{1n} to be P_n and the transition matrix for Φ_{2n} to be Q_n . $P_n(a, b)$ and $Q_n(a, b)$ be the transition probabilities of the chains Φ_{1n} and Φ_{2n} respectively for $a, b \in \Sigma^{m_n}$. Let the stationary distribution of the chain Φ_{2n} is π_n . Consider a sequence of functions $f_n : \Sigma^{m_n} \rightarrow \mathbb{R}$, and define $Y_{n,j} = f_n(Z_{n,j})$, $T_n = \sum_{j=1}^{k_n} Y_{n,j}$.*

Suppose, the following assumptions hold.

(i) $P(X_{m_n+1}^{(n)} = u_{m_n+1} | X_{m_n}^{(n)} = u_{m_n}, \dots, X_1^{(n)} = u_1) > 0$ for $u_1, \dots, u_{m_n} \in \Sigma$. This will ensure the chain will be aperiodic and irreducible.

(ii) If

$$\delta_n = \sup_{a, b \in \Sigma^{m_n}} \sum_{c \in \Sigma^{m_n}} [Q_n(a, c) - Q_n(b, c)]^+$$

then

$$\delta := \sup_n \delta_n < 1.$$

(iii) $E(Y_{n,j}) = 0$ and $(E(|Y_{n,j}|^p))^{1/p} < \infty$ for some $p > 2$. Here the expectation is taken under the stationary distribution.

(iv) $k_n \rightarrow \infty$ as $n \rightarrow \infty$, and

$$\lim_{n \rightarrow \infty} \sum_{j=1}^{k_n} (E(|Y_{n,j}|^p \mathbb{I}(|Y_{n,j}| > \epsilon)))^{2/p} = 0.$$

(v)

$$\lim_{n \rightarrow \infty} E(T_n^2) = 1; \quad \lim_{n \rightarrow \infty} (E(|Y_{n,j}|^p))^{2/p} \leq C$$

for some constant C .

Under the above assumptions, as $n \rightarrow \infty$

$$T_n \xrightarrow{d} \mathcal{N}(0, 1).$$

Note that, the above result is really important in the sense that the mixing coefficient directly depends on the transition probabilities. However, the first result (3.2.1) requires more investigation on how the conditions are satisfied in terms of transition probabilities for certain examples of SMM. In the next section, we provide such an example.

3.3 Example

In the previous section, we have discussed some requirements so that large sample results are established. One such condition was $n\pi_n(\alpha_n) \rightarrow \infty$ as $n \rightarrow \infty$. It is easy to observe that there is always a state $\alpha_n \in \Sigma^{m_n}$ with $\pi_n(\alpha_n) \geq 1/d^{m_n}$, hence assuming $n/d^{m_n} \rightarrow \infty$ will suffice. However, this bound is the most obvious one. If we can make the bound tighter, probably we will get $m_n \rightarrow \infty$ in more faster rate. One such example is given below. Suppose $\Sigma = \{0, 1\}$. Consider a VLMC of length n and order m with the contexts $\{0, 10, 110, \dots, 1^{m-1}0, 1^m\}$. Hence there are $m + 1$ many leaves in the context tree. Suppose, for a fixed m , $P(0|1^j0) = 1 - p_j$ and $P(1|1^j0) = p_j$, $j = 0, 1, \dots, m - 1$; and $P(0|1^m) = 1 - p_m$. In this set-up, we find that

$$\frac{1}{\pi_n(1^{m-1}0)} = \prod_{j=0}^{m-2} \frac{1}{p_j} + \frac{p_{m-1} + 1 - p_m}{1 - p_m} + \sum_{j=1}^{m-2} \prod_{k=j}^{m-2} \frac{1}{p_j} = q(p, m);$$

using the recursive relations for stationary probabilities. We leave the proof in the Appendix section.

If the transition probabilities satisfy $q(p, m_n) = o(n)$, we get our desired result. For example, if $p_j = (j + 1)/(j + 2)$, then $q(p, m) \leq 2m + m \sum_{j=1}^{m-2} \frac{1}{j+1} = \mathcal{O}(m \log(m))$. Hence, the required condition simplifies to $m_n \log(m_n)/n \rightarrow 0$ as $n \rightarrow \infty$.

3.4 Conclusion

Overall, the theoretical results guarantee CLT for certain functional of an higher order Markov chain, whose order changes with the length of the chain. We have also derived the CLT in terms of mixing coefficients, which will give us an idea about the value of h for which X_n and X_{n+h} will be nearly independent, i.e. how quickly we can achieve the stationarity. We also simplify the conditions of the theorem (3.2.1) by a particular type of VLMC in binary chains, and determined how the order m_n varies with n . In future, this might lead to simplifying the conditions in more general set-up.

CHAPTER

4

FITTING SPARSE MARKOV MODELS USING REGULARIZATION

4.1 Introduction

Let $\{X_t\}$ be a categorical time series in discrete time, with a finite state space Σ . We suppose that the evolution of the time series follows an m -th order Markov structure where

$$\mathcal{L}(X_{t+1}|X_s, s \leq t) = \mathcal{L}(X_{t+1}|X_s, t-m < s \leq t) \quad (4.1.1)$$

for some $m \geq 1$, where for any random vectors X, Y defined on a common probability space, we write $\mathcal{L}(Y|X)$ to denote the probability distribution of Y given X . Even when the alphabet Σ is small, such as $\Sigma = \{0, 1\}$ in applications involving binary chains or $\Sigma = \{A, G, T, C\}$ in genetics applications, complexity of the model (4.1.1) increases fairly quickly and parameter estimation may be difficult

even for moderately large m . Indeed, in the absence of a parametric model specification, the number of free parameters associated with (4.1.1) is given by $|\Sigma|^m(|\Sigma| - 1)$, which grows geometrically fast in the order m , where $|\Sigma|$ denotes the size of the alphabet, that is the number of elements in Σ .

Different dimension reduction strategies have been applied to reduce the model complexity in (4.1.1), such as Variable Length Markov Chains (VLMC) based on tree-structured conditioning sets. This idea was first introduced by Rissanen (1983), where relevant contexts (recent pasts) of variable orders are determined to form a context tree. In VLMC, $P(X_{t+1} = x_{t+1} | X_t = x_t, \dots, X_1 = x_1) = P(X_{t+1} = x_{t+1} | \tilde{X}_t^{(\ell)} = \tilde{x}_t^{(\ell)})$, where $\tilde{X}_t^{(\ell)} = (X_t, X_{t-1}, \dots, X_{t-\ell+1})$, $\tilde{x}_t^{(\ell)}$ is the observed value of $\tilde{X}_t^{(\ell)}$ and the tree length ℓ may not be a fixed number, but rather is a function of the past values (x_t, \dots, x_1) . In general, context tree models have $L(|\Sigma| - 1)$ parameters, where L is the number of leaves in the context tree. That L can take on arbitrary positive integer values for general context trees highlights the flexibility of a model with variable length contexts, and the fact that such models can lead to huge reductions in the number of parameters, especially when there is a long context in a single direction. A model of a variable order allows for a better trade-off between bias that arises through using contexts that are too short, and variance that increases with having many parameters, thus improving statistical inference. Bühlmann et al. (1999) and Bühlmann (2000) developed model selection strategies and studied asymptotic behaviour of Variable Length Markov Chains (VLMC). Recently, Kontoyiannis et al. (2020) and Papageorgiou and Kontoyiannis (2022) have developed inference and posterior representations for Bayesian Context Trees (BCT) for discrete time series analysis. These two papers also illustrate the prediction in the BCT set-up using a posterior predictive distribution.

Roos and Yu (2009b) and Roos and Yu (2009a) pointed out that there can be relevant contexts that don't have the hierarchical structure of a context tree. Although they have discussed the possibility of more general models, the analyses of those papers were limited to the case where $\Sigma = \{0, 1\}$. Recently, researchers began studying sparse models posed in terms of a general partition of the set of all m -tuples Σ^m , where m is the maximal order of Markovian dependence. Such models are called Sparse Markov Models (SMM), and introduce a sparse parametrization based on an unknown grouping of all possible m th order histories Σ^m . This generalization was first proposed by Garcia

et al. (2011), who called it Minimal Markov Models. Later on, Jääskinen et al. (2014) developed Bayesian predictive methods to analyze sequence data using SMMs. Xiong et al. (2016) extended the previous paper, introducing a recursive algorithm for optimizing the partition for an SMM. Following a similar approach, Bennett et al. (2022) developed a Sparse Markov Model using a collapsed Gibbs sampler. In this paper, we also consider SMMs in full generality, allowing an arbitrary and unknown number of groups. Specifically, let $\mathcal{C}_1, \dots, \mathcal{C}_{k_0}$ be a partition of Σ^m . Then, the Markov Chain $\{X_t\}$ in (4.1.1) is an SMM with groups $\{\mathcal{C}_1, \dots, \mathcal{C}_{k_0}\}$ if it satisfies the following sparse representation:

$$P(X_{t+1} \in \cdot | X_t = a_{-1}, \dots, X_{t-m+1} = a_{-m}) \text{ is the same for all } (a_{-m}, \dots, a_{-1}) \in \mathcal{C}_i, \quad (4.1.2)$$

for each $i = 1, \dots, k_0$. Thus, for each i , the transition probability remains unchanged over all m -step histories lying in the set \mathcal{C}_i . This reduces the number of unknown probability parameters to $k_0(|\Sigma| - 1)$. However, both the number k_0 of the sets in the partition and the sets \mathcal{C}_i themselves are unknown and must be estimated from the data.

To illustrate VLMC and SMM, we provide a very simple example of both models using DNA sequences with $\Sigma = \{A, G, T, C\}$. In figure (4.1), we present the context tree of a VLMC of order $m = 3$, with Level 0 representing the current time t . The tree structure indicates that for all 16 histories of order 3 with the most recent history being $x_{t-1} = A$, the transition matrices are the same. A similar structure holds true for $x_{t-1} = C$. If the two recent histories are $x_{t-1} = G$ and $x_{t-2} = A$, then the transition probability matrix for all 4 possible triplets (x_{t-3}, A, G) are the same; and so on. Hence the given context tree corresponds to a partition of the 64 triplets from Σ^3 into 12 different groups, with contexts represented by the leaf nodes of the graph. However, for a SMM, the grouping can be arbitrary and does not necessarily have to follow a tree structure. One such example is portrayed in figure (4.2), where we enumerated the histories of a third-order Markov model as $1, 2, 3, \dots, 64$. These histories are partitioned arbitrarily into 5 groups, where all histories in a given group have the same transition probabilities. Thus, VLMC forms a special subclass of SMM.

The generalization to SMM introduces additional challenges for model fitting. Indeed, the task of identifying the true partition is a difficult problem even for moderately large m . To appreciate

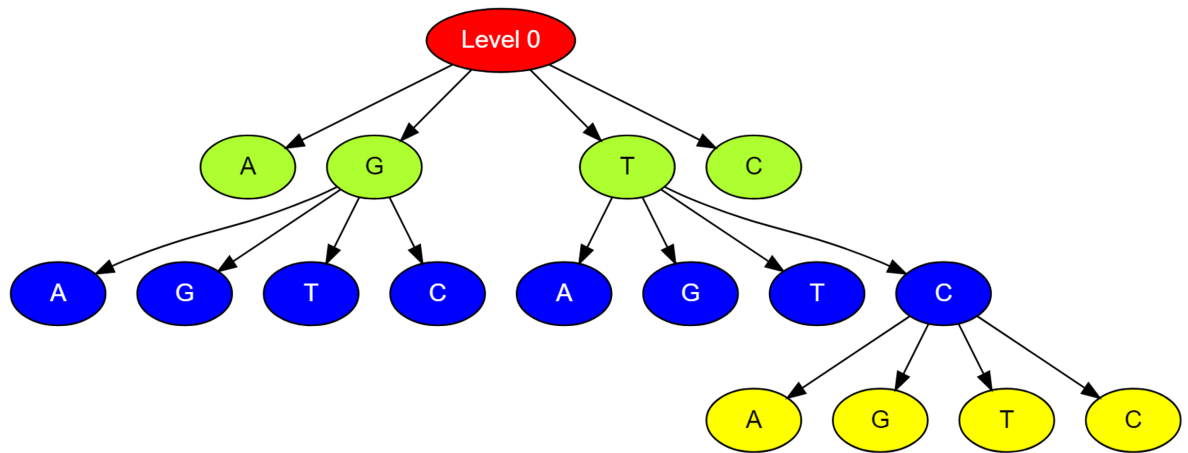


Figure 4.1: Context Tree for a VLMC of Order 3

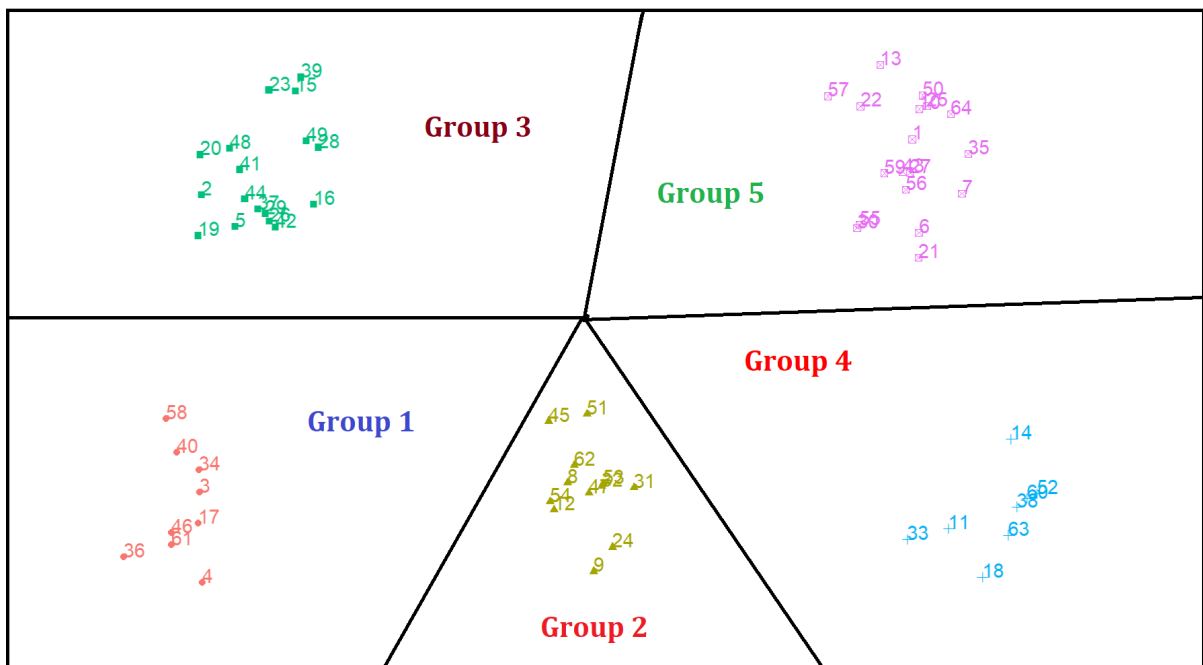


Figure 4.2: Partition of Triplets for SMM of Order 3

why, note that the total number of partitions of Σ^m , given by the well-known Bell number $B(|\Sigma|^m)$, grows at a very fast rate with the order m (cf. De Bruijn (1981)):

$$B(|\Sigma|^m) \approx \exp\left(m|\Sigma|^m \log|\Sigma|\right).$$

For example, with $\Sigma = \{0, 1\}$, $B(|\Sigma|^3) = 4140$, while $B(|\Sigma|^4) = 10480142147$. As a result, selecting the true partition from such a large collection of partitions is very difficult. Here we propose a novel methodology for selecting the true SMM model order as well as the true partition using a suitably defined penalized criterion function. The key feature of the criterion function is to discourage the distance between any two probability vectors from the *same* partition to be different from zero. By choosing the distance measure on the probability vectors suitably, we can ensure that the minimization of the penalized criterion function reduces to a convex optimization problem. Thus, our approach provides a scalable algorithm that is computationally feasible even for large m . The performance of the method depends on the choice of a regularization parameter that we select based on the data. We provide theoretical guarantees on the performance of the method by showing that it selects the true partition with probability tending to one. We also present results from an extensive simulation study that indicate encouraging finite sample performance of the proposed method in a variety of settings.

The rest of the chapter is organised as follows. In Section 4.2, we describe in detail the methodology for fitting SMMs using regularization. Section 4.3 proves theoretical properties that, in particular, ensure the model selection consistency of our method. In Section 4.4, we numerically illustrate our methodology by an extensive simulation study. A real data analysis involving virus classification has been presented in Section 4.5. We conclude this paper by summarizing our findings in Section (4.6). The proofs of the theoretical results are provided in the Appendix B.

4.2 Methodology

4.2.1 Notation

Let $\mathbb{N} = \{1, 2, \dots\}$ be the set of all positive integers, $\mathcal{X}_n = (X_1, \dots, X_n)$, and $\tilde{X}_t^{(m)} = (X_t, X_{t-1}, \dots, X_{t-m+1})$, for $m \geq 1$, $t \in \mathbb{N}$. Write w for an ordered (finite) sequence of Σ -elements of length $|w|$. Let wu denote the (ordered) concatenation of w and u . Write $|\Sigma| = d$ and w.l.o.g., set $\Sigma = \{1, \dots, d\}$. Let $\Sigma^m = \{\sigma_1, \dots, \sigma_p\}$ so that $p = |\Sigma|^m$. Let $N_w = \sum_{t=|w|}^{n-1} \mathbb{1}(\tilde{X}_t^{(|w|)} = w)$ where $\mathbb{1}(\cdot)$ denotes the indicator function. For any $S \subset \Sigma^m$ and $a \in \Sigma$, define $N_S = \sum_{t=m}^{n-1} \mathbb{1}(\tilde{X}_t^{(m)} \in S)$, $N_{S,a} = \sum_{t=m}^{n-1} \mathbb{1}(\tilde{X}_t^{(m)} \in S, X_{t+1} = a)$. In particular, N_{σ_j} denotes the number of times the chain $\tilde{X}_t^{(m)}$ hits the m -tuple σ_j , and $N_{\sigma_j,a}$ is the number of transitions from σ_j to a . Note that $n - m + 1$ denotes the total number of m -th order histories in the observed variables \mathcal{X}_n .

Next we define the probabilities associated with the SMM (4.1.2). For $j = 1, \dots, p$ and $a \in \Sigma$, let

$$\pi_{j,a} = P(X_{t+1} = a \mid \tilde{X}_t^{(m)} = \sigma_j);$$

and $\pi_j = (\pi_{j,a})_{a=1, \dots, d}$ be the corresponding transition probability vector. Note that by the SMM property, for any $a \in \Sigma$, the transition probability $\pi_{j,a}$ is a constant over all j such that $\sigma_j \in \mathcal{C}_i$. However, we do not know the sets \mathcal{C}_i and determining them is one of the challenges of fitting an SMM to a data set. To that end, define non-parametric estimators of $\pi_{j,a}$ using their empirical versions:

$$\hat{\pi}_{j,a} = N_{\sigma_j,a} / N_{\sigma_j},$$

and let $\hat{\pi}_j$ be the transition probability vectors consisting of the elements $\hat{\pi}_{j,a}$.

Here we propose a new approach to fitting the SMM based on regularization.

4.2.2 Description of the Method

Consider the penalized criterion function

$$\frac{1}{2} \sum_{j=1}^p \|\hat{\pi}_j - \mathbf{b}_j\|_2^2 + \lambda \sum_{1 \leq i < j \leq p} w_{i,j} \rho(\mathbf{b}_i, \mathbf{b}_j) \quad (4.2.3)$$

over $\mathbf{b}_j = (b_{j,1}, \dots, b_{j,d})^T \in \Pi_d$ for $j = 1, \dots, p$, where $\lambda > 0$ is a penalty parameter, $w_{i,j}$ are suitable nonnegative weights, Π_d is the d -dimensional simplex $\Pi_d = \{(u_1, \dots, u_d) \in [0, 1]^d : u_1 + \dots + u_d = 1\}$ and where $\rho(\cdot, \cdot)$ is a distance measure between two d -dimensional probability vectors. Thus, (4.2.3) treats the estimators $\hat{\pi}_{j,a}$ as (correlated) “observations” and penalizes the distance between all distinct pairs of probability vectors in order to identify the identical probability vectors. In particular, the number of parameters grows at a rate proportional to the size of the true partition in the SMM and with a suitable choice of the penalization term, one can identify the identical probability vectors. When $\rho(\mathbf{b}_i, \mathbf{b}_j)^2 = \sum_{a=1}^d (b_{i,a} - b_{j,a})^2$, (4.2.3) gives a version of the Group LASSO of Yuan and Lin (2006) that is designed for selecting pairs of full vectors that are close, and we have a convex optimization problem that can be solved for large p . On the other hand, if we use the ℓ_1 distance $\rho(\mathbf{b}_i, \mathbf{b}_j) = \sum_{a=1}^d |b_{i,a} - b_{j,a}|$, then only component-wise zero differences can be identified.

Once we minimize the criterion function in (4.2.3), it is a relatively easy task to find estimates of k_0 and the sets \mathcal{C}_i . Specifically, we start with a pair with the smallest i and seek all $j > i$ such that the distance between the solutions \mathbf{b}_i^* and \mathbf{b}_j^* is zero. Then, we set $\hat{\mathcal{C}}_1$ to be the set consisting of i and all such j . In the next step, we consider all pairs that are not in $\hat{\mathcal{C}}_1$ and repeat the procedure until all pairs with estimated zero distances have been grouped. In case there are indices j for which none of the estimated paired distances are zero, we keep them as singletons, that is groups consisting of single elements. This gives the estimated groups $\hat{\mathcal{C}}_i : i = 1, \dots, \hat{k}$, with \hat{k} giving an estimate of k_0 .

In comparison, traditional clustering methodologies like K -means have many limitations. In most cases, we have to pre-specify the number of clusters, along with the possibility that we end up with a local minima instead of the global one. The advantage of clustering by solving equation (4.2.3) for a range of λ is that we get a solution path from at most p many singleton clusters to only one cluster consisting of all the elements. Subsequently, we can fix some criterion function which will enable us to find the optimum cluster assignment among all possible models in the solution path. Hence, not only do we not need to fix the number of clusters beforehand, but we also avoid the problem of being stuck at local minima. This particular approach will be broadly referred to as “Convex Clustering”.

4.2.3 Computational considerations

Several efficient algorithms have been developed in recent years to solve equation (4.2.3) when the penalty function ρ is convex; e.g. $\rho(\mathbf{b}_i, \mathbf{b}_j) = \|\mathbf{b}_i - \mathbf{b}_j\|_p$ for some $p \geq 1$. Pelckmans et al. (2005), Lindsten et al. (2011), Hocking et al. (2011) and others recently proposed this convex clustering approach and established it to be more robust and scalable in comparison to the traditional approaches. Lindsten et al. (2011) used an off-the-shelf convex solver CVX to solve the convex clustering problem, which suffers from scalability issues. Theoretical perfect cluster recovery conditions have been derived by Zhu et al. (2014) only for two clusters, while Panahi et al. (2017) derived perfect recovery conditions for general k clusters, but under the assumption of uniform weights. Sun et al. (2021) provided sufficient conditions for theoretical recovery conditions under more general weight choices. They have also developed a faster algorithm called semismooth Newton based augmented Lagrangian method (SS-NAL), and derived the convergence criteria for their algorithm. Recently, Wang and Allen (2021) have introduced the Integrative Generalized Convex Clustering Optimization (iGecco) method for solving the convex clustering problem for more general loss functions, including non-differentiable ones. The major difference of our set-up from previous developments is that we cluster empirical transition probability vectors as opposed to the original data points.

We now introduce the specific computational algorithm that we choose to implement to minimize our objective function in (4.2.3). In a recent work, Chi and Lange (2015) developed an elegant method of solving convex clustering problems by augmented Lagrangian methods that we will exploit to fit the SMM models. For $\rho(\mathbf{x}) = \|\mathbf{x}\|_2$, we first view solving equation (4.2.3) as the following constrained optimization problem

$$\begin{aligned} \min & \frac{1}{2} \sum_{j=1}^p \|\hat{\pi}_j - \mathbf{b}_j\|_2^2 + \lambda \sum_{l \in \mathcal{E}} w_l \|\mathbf{v}_l\|_2 \\ \text{subject to} & \mathbf{b}_{l_1} - \mathbf{b}_{l_2} - \mathbf{v}_l = 0; \end{aligned} \tag{4.2.4}$$

where \mathcal{E} is the set of all distinct edges $\{l : l = (l_1, l_2), l_1 < l_2, w_l > 0\}$. Here, a new splitting variable \mathbf{v}_l has been introduced to capture the difference between the group centroids, which makes the optimization procedure much easier. Two algorithms have been developed for solving this constrained

optimization problem, namely ADMM and AMA. For both these algorithms, first we incorporate an augmented Lagrangian as follows:

$$\begin{aligned} \mathcal{L}_\nu(\mathbf{B}, \mathbf{V}, \mathbf{\Gamma}) = & \frac{1}{2} \sum_{j=1}^p \|\hat{\pi}_j - \mathbf{b}_j\|_2^2 + \lambda \sum_{l \in \mathcal{E}} w_l \|\mathbf{v}_l\|_2 \\ & + \sum_{l \in \mathcal{E}} \langle \gamma_l, \mathbf{v}_l - \mathbf{b}_{l_1} + \mathbf{b}_{l_2} \rangle + \frac{\nu}{2} \sum_{l \in \mathcal{E}} \|\mathbf{v}_l - \mathbf{b}_{l_1} + \mathbf{b}_{l_2}\|_2^2, \end{aligned} \quad (4.2.5)$$

where \mathbf{B}, \mathbf{V} and $\mathbf{\Gamma}$ are the matrices with $\mathbf{b}_j, \mathbf{v}_l$ and γ_l for $j = 1, \dots, p$ and $l \in \mathcal{E}$ in their columns respectively. Splitting the variables in such fashion would allow us to update \mathbf{B}, \mathbf{V} and $\mathbf{\Gamma}$ sequentially, given the other variables. The convergence of ADMM does not depend on the choice of ν ; it is known to converge for any $\nu > 0$. On the other hand, AMA converges for any $0 < \nu < 2/p$. The performance of both these algorithms have been compared, and it has been established that AMA is much faster than ADMM, especially when the weights are sparse. Since AMA provides much faster results, we will use this algorithm in numerical implementation of our methodology. Suppose, $\mathbf{B}^{(t)}$ and $\mathbf{\Gamma}^{(t)}$ be the parameter values in the t^{th} step. The updates in the next step are computed using the following relations:

$$\begin{aligned} \mathbf{b}_j^{(t+1)} &= \hat{\pi}_j + \sum_{l_1=j} \gamma_l^{(t)} - \sum_{l_2=j} \gamma_l^{(t)} \\ \gamma_l^{(t+1)} &= \mathcal{P}_{C_l}(\gamma_l^{(t)} - \nu \mathbf{g}_l^{(t+1)}) \end{aligned}$$

where $\mathbf{g}_l^{(t+1)} = \mathbf{b}_{l_1}^{(t+1)} - \mathbf{b}_{l_2}^{(t+1)}$, $C_l = \{\gamma_l : \|\gamma_l\|_2 \leq \lambda w_l\}$, and $\mathcal{P}_A(\mathbf{x})$ is the projection of \mathbf{x} onto the set A . We continue until convergence, and the convergence criterion can be formulated using the dual problem and duality gap.

4.2.4 Selection of the Tuning Parameter

So far, we have discussed the numerical methods to solve (4.2.3) for a given λ . But it is important to choose an optimum value of λ for the optimization problem. In this section, we propose a data driven method to select this tuning parameter using the BIC criterion. For a given λ , denote the obtained clusters as $\mathcal{C}_1(\lambda), \dots, \mathcal{C}_{k_\lambda}(\lambda)$, where k_λ is the number of clusters. Define the common

Algorithm 1 AMA

Initialize $\Gamma^{(0)}$

```
1: for  $t = 1, 2, 3, \dots$  do
2:   for  $j = 1, 2, 3, \dots, p$  do
3:      $\Delta_j^{(t)} = \sum_{l_1=j} \gamma_{l_1}^{(t-1)} - \sum_{l_2=j} \gamma_{l_2}^{(t-1)}$ 
4:   end for
5:   for all  $l$  do
6:      $\mathbf{g}_l^{(t)} = \hat{\pi}_{l_1} - \hat{\pi}_{l_2} + \Delta_{l_1}^{(t)} - \Delta_{l_2}^{(t)}$ 
7:      $\gamma_l^{(t)} = \mathcal{P}_{C_l}(\gamma_l^{(t-1)} - \nu \mathbf{g}_l^{(t)})$ 
8:   end for
9: end for
```

transition probability for the m -tuples in the estimated group $\hat{\mathcal{C}}_a(\lambda)$ as

$$\hat{R}_{a,a}^{(\lambda)} = \frac{\sum_{\sigma_j \in \hat{\mathcal{C}}_a(\lambda)} N_{\sigma_j, a}}{\sum_{\sigma_j \in \hat{\mathcal{C}}_a(\lambda)} N_{\sigma_j}} = \frac{N_{\hat{\mathcal{C}}_a(\lambda), a}}{N_{\hat{\mathcal{C}}_a(\lambda)}} \quad \forall a = 1, \dots, k_\lambda; a \in \Sigma.$$

The log-likelihood of the observations under the obtained cluster assignment for a particular λ is given by

$$\ell_n(\lambda) = \sum_{a=1}^{k_\lambda} \sum_{a \in \Sigma} N_{\hat{\mathcal{C}}_a(\lambda), a} \log \hat{R}_{a,a}^{(\lambda)}.$$

Hence, the BIC score corresponding to the obtained model is

$$BIC_n(\lambda) = -2\ell_n(\lambda) + k_\lambda(|\Sigma| - 1) \log n.$$

By a grid search over a range of possible λ values, we select the λ for which BIC is minimized. The solution of equation (4.2.3) corresponding to that λ is considered as the estimated cluster assignment. The novelty of our method is that we are able to select the optimum tuning parameter from the data itself. For general convex clustering scenarios, one may not be able to compute the BIC criterion since the distributional properties of the data points in a cluster are unknown. The assumption of Markovian structure is useful in our set-up to formulate the likelihood function. Moreover, the CLT-type results provide us the asymptotic distributions of the estimated transition probabilities. In the next section, we provide new theoretical results to demonstrate that for a range of λ values, we will be able to perfectly recover the true clusters for large n .

4.3 Conditions and Theoretical Results

4.3.1 Conditions

We consider equation (4.2.3) with $\rho(\mathbf{b}_i, \mathbf{b}_j) = \|\mathbf{b}_i - \mathbf{b}_j\|_2$. Let the optimum solution be denoted by $\mathbf{b}_i^*(\lambda)$, for $i = 1, 2, \dots, p$. Also, let the true partition of the state space Σ^m be $\{\mathcal{C}_1, \dots, \mathcal{C}_{k_0}\}$, with the corresponding transition probability vectors being $\mathbf{R}_1, \dots, \mathbf{R}_{k_0}$. Thus, $\mathbf{R}_{\alpha, \alpha} = P(X_{t+1} = a | Y_t = \sigma_\alpha)$. Set $p_\alpha = |\mathcal{C}_\alpha|$, the size of the α^{th} partition. Following the notation of Sun et al. (2021), define

$$\begin{aligned} w_i^{(\beta)} &= \sum_{j \in \mathcal{C}_\beta} w_{i,j} \quad \forall i = 1, 2, \dots, p; & \mu_{i,j}^{(\alpha)} &= \sum_{\ell \neq \alpha} |w_i^{(\ell)} - w_j^{(\ell)}| \quad \forall \alpha = 1, 2, \dots, k_0; \\ w^{\alpha, \beta} &= \sum_{i \in \mathcal{C}_\alpha} \sum_{j \in \mathcal{C}_\beta} w_{i,j} \quad \forall \alpha \neq \beta, \alpha, \beta \in \{1, 2, \dots, k_0\}; & \hat{\pi}^{(\alpha)} &= \frac{1}{p_\alpha} \sum_{i \in \mathcal{C}_\alpha} \hat{\pi}_i; \\ \lambda_{\min}^{(n)} &= \max_{1 \leq \alpha \leq k_0} \max_{i, j \in \mathcal{C}_\alpha} \left\{ \frac{\|\hat{\pi}_i - \hat{\pi}_j\|_2}{p_\alpha w_{i,j} - \mu_{i,j}^{(\alpha)}} \right\}; \\ \lambda_{\max}^{(n)} &= \min_{1 \leq \alpha < \beta \leq k_0} \left\{ \frac{\|\hat{\pi}^{(\alpha)} - \hat{\pi}^{(\beta)}\|_2}{\frac{1}{p_\alpha} \sum_{l \neq \alpha} w^{(\alpha, l)} + \frac{1}{p_\beta} \sum_{l \neq \beta} w^{(\beta, l)}} \right\}. \end{aligned}$$

We shall suppose that the following conditions hold.

(A1) $w_{i,j} = w_{j,i}$ and $w_{i,j} > 0$ for any $i, j \in \mathcal{C}_\ell$, $\ell = 1, 2, \dots, k_0$.

(A2) $p_\alpha w_{i,j} > \mu_{i,j}^{(\alpha)}$, $\forall i, j \in \mathcal{C}_\alpha$ and $\forall \alpha = 1, 2, \dots, k_0$.

In (A1) we assume symmetry, and that the weight is positive between two m -tuples belonging to the same partition. (A2) gives a lower bound for the weight between two m -tuple in a particular group. Similar conditions have been used by Sun et al. (2021) to prove perfect recovery results. In Theorem 4.3.6 below, we provide some simple sufficient conditions on weight choices to satisfy these conditions.

Before going into the main results, we state two auxiliary results that will be used for the subsequent results.

Proposition 4.3.1 *Suppose $\{X_n\}_{n \geq 1}$ be an aperiodic and irreducible SMM of order m , with the true*

partition $\{\mathcal{C}_1, \dots, \mathcal{C}_{k_0}\}$. Then, as $n \rightarrow \infty$,

- (a) $\hat{\pi}_j \xrightarrow{p} \mathbf{R}_\alpha$ for $j \in \mathcal{C}_\alpha$;
- (b) $\frac{N_{\sigma_j}}{N} \xrightarrow{p} q_j$, where q_j is the stationary probability of the state σ_j ;
- (c) With $\Sigma_\alpha = \text{diag}(\mathbf{R}_\alpha) - \mathbf{R}_\alpha \mathbf{R}_\alpha^{(T)}$,

$$\sqrt{N_{\sigma_j}}(\hat{\pi}_j - \mathbf{R}_\alpha) \xrightarrow{d} \mathcal{N}(\mathbf{0}, \Sigma_\alpha).$$

Since Σ_α is of rank $|\Sigma| - 1$, the asymptotic Normal distribution is singular.

Thus, Proposition (4.3.1) asserts weak consistency and asymptotic normality of the estimated transition probability vectors, which can be proved using existing results on Markov chains in Billingsley (1961) or Meyn and Tweedie (2012). The next result deals with perfect recovery under general weight choices under Conditions (A.1) and (A.2), which is a direct consequence of the results in Sun et al. (2021).

Proposition 4.3.2 *Suppose the above conditions (A1) and (A2) hold and $\lambda_{\min}^{(n)} < \lambda_{\max}^{(n)}$. Then for any $\lambda \in (\lambda_{\min}^{(n)}, \lambda_{\max}^{(n)})$, $\mathbf{b}_i^*(\lambda) = \mathbf{b}_j^*(\lambda)$ for $i, j \in \mathcal{C}_\alpha$; $\alpha = 1, \dots, k_0$ and $\mathbf{b}_i^*(\lambda) \neq \mathbf{b}_j^*(\lambda)$ for any $i \in \mathcal{C}_\alpha, j \in \mathcal{C}_\beta, \alpha \neq \beta$. In other words, for any $\lambda \in (\lambda_{\min}^{(n)}, \lambda_{\max}^{(n)})$, we recover the true partition of the state space.*

These propositions will be among the key tools used for proving our results. The CLT result will be useful for determining probability bounds for perfect recovery under the conditions of the proposition (4.3.2). In the next subsection, we state our major theoretical findings.

4.3.2 Main results

Though it may not appear that the resulting solution of the objective function in (4.2.4) will produce a valid probability distribution over Σ , according to the theorem (4.3.3), which is proven in the appendix, that must be the case. However, we do not consider the solution as our estimated transition probabilities for the partitions, since the resulting vectors do not represent the true cluster centers in most of the cases.

Theorem 4.3.3 For any $\lambda > 0$, the optimal solution $\mathbf{b}_i^*(\lambda)$ is a valid probability distribution for $i = 1, 2, \dots, p$; i.e.

$$(a) \ b_{i,a}^*(\lambda) \geq 0 \text{ for } a = 1, \dots, d,$$

$$(b) \ \sum_{a=1}^d b_{i,a}^*(\lambda) = 1.$$

Next, we would like to derive the probability of true cluster recovery. There are two steps involved in this process. First, we need the true model in the solution path over varying λ . This implies the conditions of Proposition 4.3.2 must be satisfied, i.e. $\lambda_{\min}^{(n)} < \lambda_{\max}^{(n)}$. From Theorem 4.3.4, we get a lower bound of the probability of the true model being present in the solution path. Note that Sun et al. (2021) have derived these perfect recovery conditions for a given fixed data set when the data points in a particular cluster are close to each other. Our approach is significantly different from that, as we cluster the estimated transition probability vectors, which are random variables. This leads to the facts that $\lambda_{\min}^{(n)}$ and $\lambda_{\max}^{(n)}$ are random variables as well. We provide theoretical bounds to ensure that the probability of the event $\{\lambda_{\min}^{(n)} < \lambda_{\max}^{(n)}\}$ is $1 - \mathcal{O}_p(e^{-n})$.

Theorem 4.3.4 Define

$$\begin{aligned} \delta &= \min_{1 \leq \alpha < \beta \leq k_0} \|\mathbf{R}_\alpha - \mathbf{R}_\beta\|_2; & \delta_1 &= \min_{1 \leq \alpha \leq k_0} \min_{i, j \in \mathcal{C}_\alpha} (p_\alpha w_{i,j} - \mu_{i,j}^{(\alpha)}) \\ \delta_2 &= \max_{1 \leq \alpha < \beta \leq k_0} \left(\frac{1}{p_\alpha} \sum_{l \neq \alpha} w^{(\alpha,l)} + \frac{1}{p_\beta} \sum_{l \neq \beta} w^{(\beta,l)} \right). \end{aligned}$$

Then, under Conditions (A1) and (A2), as $n \rightarrow \infty$,

$$\begin{aligned} P(\lambda_{\min}^{(n)} < \lambda_{\max}^{(n)}) &\geq P(\|\hat{\pi}_j - \mathbf{R}_\alpha\|_2 < \frac{\epsilon}{2} \forall j \in \mathcal{C}_\alpha, \forall \alpha = 1, \dots, k_0) \\ &\geq 1 - \sum_{\alpha=1}^{k_0} C_1^{(\alpha)} \sum_{j \in \mathcal{C}_\alpha} \exp[-(n-m)\epsilon^2 C_{2,j}] \end{aligned}$$

for $0 < \epsilon < \frac{\delta \delta_1}{\delta_1 + \delta_2}$, and for some constants $C_1^{(\alpha)}, C_{2,j} > 0$.

Looking at the expressions for $\lambda_{\min}^{(n)}$ and $\lambda_{\max}^{(n)}$, it is evident that $\lambda_{\min}^{(n)}$ shrinks towards 0 as n increases as the estimated transition probability vectors $\hat{\pi}_i$ and $\hat{\pi}_j$ belonging to the same cluster

\mathcal{C}_α become closer to each other. On the other hand, the different group means $\hat{\pi}^{(\alpha)}$ and $\hat{\pi}^{(\beta)}$ tend to get separated from each other, making $\lambda_{\max}^{(n)}$ converge to a positive number, so that eventually we get $\lambda_{\min}^{(n)} < \lambda_{\max}^{(n)}$. These expressions also tell us that in order to have perfect recovery of the clusters, a scaled version of the maximum within-group deviation of the transition probabilities should be less than a scaled version of the minimum between-group variation. These scales are heavily dependent on the choice of the weights $w_{i,j}$. Note that, if we choose the weights in a way so that $w_{i,j}$ is higher if $\hat{\pi}_i$ and $\hat{\pi}_j$ are closer (and potentially belong to the same cluster), and lower if they are far from each other (potentially belonging to different clusters), the denominator of the term $\lambda_{\min}^{(n)}$ will be higher, and the denominator of $\lambda_{\max}^{(n)}$ will be lower in the ideal scenario. Hence, these particular choice of the weights will enhance separating $\lambda_{\min}^{(n)}$ and $\lambda_{\max}^{(n)}$, increasing the chance of recovering the true cluster assignment. Once we have the true model in the solution path, the next step is to establish that the probability of selecting that model through the BIC criterion converges to 1 as $n \rightarrow \infty$. The next theorem gives the precise statement of this result.

Theorem 4.3.5 *Suppose the conditions of the Theorem (4.3.4) holds, and $\lambda_{\min}^{(n)} < \lambda_{\max}^{(n)}$. For any λ , denote the clustering assignment obtained by minimizing the equation (4.2.3) as $M_\lambda = \{\hat{\mathcal{C}}_1(\lambda), \dots, \hat{\mathcal{C}}_{k_\lambda}(\lambda)\}$; where k_λ is the associated number of clusters. Suppose $\ell_n(\lambda)$ is the log-likelihood of the observations corresponding to the cluster assignment M_λ , and the corresponding BIC score is $BIC_n(\lambda) = -2\ell_n(\lambda) + k_\lambda(d-1)\log n$. Choose some $\lambda_0 \in (\lambda_{\min}^{(n)}, \lambda_{\max}^{(n)})$. Then, for any λ such that $M_\lambda \neq M_{\lambda_0}$,*

$$P\left(BIC_n(\lambda_0) < BIC_n(\lambda)\right) \rightarrow 1$$

as $n \rightarrow \infty$.

Theorem 4.3.5 asserts the consistency of the model selection using BIC criterion and is one of the most important conclusions of this paper. Variable selection consistency results in Zhang et al. (2010) use a similar BIC-criterion for the LASSO penalty in multiple linear regression. But our set-up is very different from the regression set-up (with independent observations). Further, the penalty function is quite different as well. As a result, the key steps for proving our result are very different. The consequence of this theoretical result is extremely important when we apply our method in

the simulation study or in real data applications. Even for moderately large sample sizes, we can achieve good clustering performance by this BIC optimization criterion. We will demonstrate these properties in a finite sample situation in the simulation study.

Although we have stated all of our results under conditions (A.1) and (A.2), we still need to check whether these conditions are feasible in practice. The next result provides sufficient conditions for perfect cluster recovery under a particular weight choice involving Gaussian kernels that also produce good clustering results in finite samples.

Theorem 4.3.6 *Define $p_{min} = \min_{\alpha} p_{\alpha}$, $p_{max} = \max_{\alpha} p_{\alpha}$, and assume the cluster sizes are different.*

Suppose

(a) $w_{i,j} = e^{-\phi \|\hat{\pi}_i - \hat{\pi}_j\|_2^2} l_{i,j}^k$, where $l_{i,j}^k$ is the indicator function that $\hat{\pi}_i$ is one of the k nearest neighbours of $\hat{\pi}_j$ or vice versa, for some $\phi > 0$;

(b) $k \geq p_{max} - 1$;

(c) for some $\epsilon < \epsilon_{max} = \frac{\delta}{2} - \frac{1}{2\phi\delta} \log\left(2\left(\frac{k'+1}{p_{min}} - 1\right)\right)$, $\|\hat{\pi}_j - \mathbf{R}_{\alpha}\|_2 < \frac{\epsilon}{2}$; $\forall j \in \mathcal{C}_{\alpha}$, $\forall \alpha = 1, \dots, k_0$, where

$$k' = \max_i \sum_{j=1}^p l_{i,j}^k.$$

Then conditions (A1) and (A2) are satisfied. Moreover, $\delta_1 \geq p_{min} e^{-\phi \epsilon_{max}^2 - 2(k'+1-p_{min})} e^{-\phi(\delta - \epsilon_{max})^2} = \delta_1^{(min)}$, $\delta_2 \leq 2(k'+1-p_{min}) e^{-\phi(\delta - \epsilon_{max})^2} = \delta_2^{(max)}$.

Theorem 4.3.6 simplifies Conditions (A1) and (A2) for a special choice of weights, which we will use later to demonstrate our simulation studies. The intuition behind this choice is that $w_{i,j}$ should be a decreasing function of $\|\hat{\pi}_i - \hat{\pi}_j\|_2$ which enforces less penalization for well separated points. The following results are direct consequences of Theorem 4.3.6:

Corollary 4.3.6.1 *Under the assumptions of Theorem 4.3.4,*

$$(a) \lambda_{min}^{(n)} \leq \frac{\epsilon}{\delta_1^{(min)}}, \lambda_{max}^{(n)} \geq \frac{\delta - \epsilon}{\delta_2^{(max)}}.$$

$$(b) \quad \epsilon < \min \left\{ \epsilon_{max}, \frac{\delta \delta_1}{\delta_1 + \delta_2} \right\} \implies \lambda_{min}^{(n)} < \lambda_{max}^{(n)}.$$

Corollary 4.3.6.2 *For a balanced design, i.e. when $p_\alpha = p/k_0$ are the same for all groups \mathcal{C}_α , $\delta_2^{(max)} = 0$ if $k = p/k_0 - 1$. Hence, for any $\epsilon < \frac{\delta}{2}$, perfect recovery is possible for $\lambda \in (\lambda_{min}^{(n)}, \infty)$.*

We present all proofs in the appendix section. Corollary 4.3.6.1 gives us an idea about how close the empirical transition probabilities for each m -tuple are to the true probability vectors. Corollary 4.3.6.2 considers a special case when the design is balanced. In that scenario, for large n and for the correct choice of the nearest neighbour, the true model can be retrieved for a wide range of tuning parameters λ , thereby providing very accurate clustering results. In the next section, we will explore the impact of weight choices on clustering accuracy through simulations of finite samples.

4.4 Simulation study

In this section, we will numerically demonstrate the performance of the convex clustering methodology described in the previous section in terms of recovering the true cluster assignments. We compare the clustering performance for different choices of the weights $w_{i,j}$. We consider SMM of various orders, lengths and $|\Sigma|$ values. Note that we don't pre-specify the number of clusters or the labels of the clusters in our method, so it is not feasible to compute the straightforward miss-classification rate to compare the outcome with the true one. Instead, Rand Index (RI) and Adjusted Rand Index (ARI) are used to measure the similarity between the true cluster and the obtained cluster assignment.

Rand Index computes the proportion of (i, j) pairs that are correctly identified as belonging to same cluster or different clusters. Mathematically, for any two cluster assignments $X = (X_1, \dots, X_r)$ and $Y = (Y_1, \dots, Y_s)$ of the elements $(\sigma_1, \dots, \sigma_p)$, the Rand Index is defined by

$$RI = \frac{a + b}{a + b + c + d} = \frac{a + b}{\binom{p}{2}}$$

where a is the number of pairs that are in the same cluster in both X and Y , b is the number of pairs that are in the different clusters in both X and Y , c is the number of pairs that are in same

cluster of X , but in different clusters of Y , and d is number of pairs which are in same cluster of Y , but in different clusters of X . Values of RI vary between 0 and 1. If two clusters are identical, RI should be 1. Higher RI values indicate more similarity among two given clusters.

However, Rand Index has some limitations. For example, if the number of clusters increases, and the cluster sizes are not large, RI will be close to 1 even for two completely different cluster assignments. To address this issue, usage of Adjusted Rand Index (ARI) is preferred. ARI uses the expected similarity of all pairwise comparisons between clusterings specified by a random model. If $a_i = |X_i|$, $b_j = |Y_j|$, and $p_{ij} = |X_i \cap Y_j|$, then ARI is computed by the following formula:

$$ARI = \frac{\sum_{i,j} \binom{p_{ij}}{2} - [\sum_i \binom{a_i}{2} \sum_j \binom{b_j}{2}] / \binom{p}{2}}{\frac{1}{2} [\sum_i \binom{a_i}{2} + \sum_j \binom{b_j}{2}] - [\sum_i \binom{a_i}{2} \sum_j \binom{b_j}{2}] / \binom{p}{2}}.$$

For each of our simulation study, we compare the similarity between the estimated cluster assignment by solving the equation (4.2.3) for the appropriate regularization parameter λ of the true clustering using both RI and ARI . We focus on the choice of weights that result in higher ARI values. Chi and Lange (2015) and Sun et al. (2021) have shown, both numerically and theoretically, that choosing sparse weights substantially improves the clustering quality, and also makes the algorithm much faster. In our study, we perform clustering under different weight choices, both sparse and dense, and compare how the ARI values depend on that choice. In each set-up, we replicate the experiment 1000 times to obtain the mean RI and ARI and their standard error. We also compute the proportion of times ARI or RI is 1, i.e. we empirically compute the probability of perfect cluster recovery.

4.4.1 Simulation Set-up 1

Here, we take $|\Sigma| = 4$, the usual scenario when analyzing the DNA sequences. The order of the chain m is taken to be both 2 or 3. For $m = 2$, we equally divide the 16 tuples into 4 groups of 4 elements; and for $m = 3$, we divide the 64 triplets into 8 groups of equal size 8. For a particular group C_i , we generate $Z_{C_i,\ell}$ independently from $Unif(0, 1)$, for $\ell = 1, 2, 3, 4$. The transition probability of that

group is generated from a Dirichlet distribution with parameter $(e^{Z_{C_i,1}}, \dots, e^{Z_{C_i,4}})$. As weights, we first take $w_{i,j} = 1$ for all $i, j = 1, 2, \dots, p, i < j$. Next we choose some sparse weights depending on the distance between the estimated transition probabilities $\hat{\pi}_i$ and $\hat{\pi}_j$. We have used the k -nearest neighbour based weights as proposed in Chi and Lange (2015), such that $w_{i,j} = \exp^{-\phi \|\hat{\pi}_i - \hat{\pi}_j\|_2^2} l_{i,j}^k$, where $l_{i,j}^k$ is the indicator function that $\hat{\pi}_i$ belongs to k nearest neighbour of $\hat{\pi}_j$ or vice versa, for some $\phi > 0$. In this example we use $\phi = 100$. We also incorporate a third choice of weight, namely $w_{i,j} = \exp^{-\phi \|\hat{\pi}_i - \hat{\pi}_j\|_\infty^2} l_{i,j}^{k(\infty)}$ where $l_{i,j}^{k(\infty)}$ is the similar indicator function, but the nearest neighbour is computed w.r.t l_∞ distance. We use two different values, $k = 5$ and $k = 3$. The results are provided in the following tables.

From table (4.1), it is clear that for $m = 3$, uniform weights, i.e. $w_{i,j} = 1$ for all pairs (i, j) , results

Table 4.1: Clustering Performance for $m = 3, w_{i,j} = 1$.

Sample Size (n)	5000	10000	15000	20000	25000
RI (s.e)	0.866 (0.03)	0.901 (0.04)	0.933 (0.04)	0.954 (0.03)	0.969 (0.03)
ARI (s.e)	0.199 (0.18)	0.469 (0.26)	0.651 (0.22)	0.777 (0.17)	0.854 (0.15)
Prob. of True Recovery	0	0.003	0.017	0.042	0.128

Table 4.2: Clustering Performance for $m = 3, w_{i,j} = \exp^{-\phi \|\hat{\pi}_i - \hat{\pi}_j\|_2^2} l_{i,j}^k$.

k nearest neighbour= 5					
Sample Size (n)	5000	10000	15000	20000	25000
RI (s.e)	0.982 (0.024)	0.995 (0.007)	0.997 (0.005)	0.999 (0.002)	1 (0.001)
ARI (s.e)	0.935 (0.081)	0.980 (0.027)	0.990 (0.019)	0.997 (0.009)	0.999 (0.004)
Prob. of True Recovery	0.264	0.466	0.714	0.907	0.981
k nearest neighbour= 3					
Sample Size (n)	5000	10000	15000	20000	25000
RI (s.e)	0.984 (0.024)	0.994 (0.007)	0.997 (0.005)	0.999 (0.002)	1 (0.001)
ARI (s.e)	0.940 (0.074)	0.979 (0.026)	0.990 (0.019)	0.997 (0.009)	0.999 (0.004)
Prob. of True Recovery	0.223	0.408	0.713	0.908	0.981

in really poor performance in terms of recovering the true cluster. Although the *ARI* increases with increasing n , we may need really large sample size to get good results. On the other hand, for $m = 2$, choosing the weights using $k = 3$ performs much better than that with $k = 5$ in terms of both *ARI* and perfect recovery, especially for lower sample sizes. Note that the model is balanced in this example, and the optimum choice of k is 3 (by Corollary 4.3.6.1). We can justify that fact using our

Table 4.3: Clustering Performance for $m = 3$, $w_{i,j} = \exp^{-\phi \|\hat{\pi}_i - \hat{\pi}_j\|_2^2} I_{i,j}^{k(\infty)}$.

k nearest neighbour= 5					
Sample Size (n)	5000	10000	15000	20000	25000
RI (s.e)	0.981 (0.025)	0.995 (0.009)	0.998 (0.005)	0.999 (0.002)	1 (0.001)
ARI (s.e)	0.931 (0.085)	0.980 (0.033)	0.990 (0.018)	0.997 (0.009)	0.999 (0.004)
Prob. of True Recovery	0.294	0.508	0.735	0.904	0.980
k nearest neighbour= 3					
Sample Size (n)	5000	10000	15000	20000	25000
RI (s.e)	0.983 (0.021)	0.994 (0.007)	0.997 (0.005)	0.999 (0.002)	1 (0.001)
ARI (s.e)	0.937 (0.076)	0.977 (0.029)	0.990 (0.019)	0.997 (0.009)	0.999 (0.004)
Prob. of True Recovery	0.206	0.387	0.709	0.907	0.981

Table 4.4: Clustering Performance for $m = 2$, $w_{i,j} = \exp^{-\phi \|\hat{\pi}_i - \hat{\pi}_j\|_2^2} I_{i,j}^k$.

k nearest neighbour= 5					
Sample Size (n)	5000	10000	15000	20000	25000
RI (s.e)	0.900 (0.09)	0.981 (0.04)	0.994 (0.02)	0.997 (0.01)	0.999 (0.01)
ARI (s.e)	0.745 (0.19)	0.946 (0.10)	0.982 (0.05)	0.992 (0.04)	0.997 (0.02)
Prob. of True Recovery	0.223	0.708	0.876	0.951	0.977
k nearest neighbour= 3					
Sample Size (n)	5000	10000	15000	20000	25000
RI (s.e)	0.945 (0.07)	0.994 (0.02)	0.999 (0.01)	0.999 (0.005)	1 (0.003)
ARI (s.e)	0.851 (0.17)	0.983 (0.06)	0.995 (0.03)	0.998 (0.02)	0.999 (0.01)
Prob. of True Recovery	0.480	0.908	0.972	0.991	0.996

simulation study. On the other hand, the optimum choice of k is 7 for $m = 3$, but the choice $k = 5$ is reliable as well. $k = 3$ makes the weights too sparse in this case, which results in a small degradation of the clustering accuracy. For both $m = 2$ and $m = 3$, the probability of true recovery increases with increasing n for all weight choices.

This experiment gives very good results for large n , mostly for $n \geq 10000$. It is worth of investigating under what circumstances we will be able to get very good recovery for smaller n , such as $n = 1000$ or $n = 2000$. The theoretical results suggest that if the cluster centroids are well separated, clustering performance gets better even for smaller sample sizes. In this experiment, we have 4 groups for $m = 2$, with the minimum centroid difference 0.123 in terms of l_2 distance and 0.108 in terms of l_∞ distance; these values are respectively 0.105 and 0.067 for $m = 3$. In the next simulation study, we will demonstrate how the clustering accuracy improves for well-separated centroids.

4.4.2 Simulation Set-up 2

Here we take $|\Sigma| = 4$ and $m = 3$. We divide this 64 triplets into four groups of sizes 18, 18, 15 and 13. For the α^{th} group, $R_{\alpha,\alpha} = 0.7, R_{\alpha,\beta} = 0.1, \alpha = 1, 2, 3, 4, \beta = 1, 2, 3, 4, \alpha \neq \beta$. As the choice of weight, we first used the $k = 15$ nearest neighbours w.r.t. the l_2 distance in the Gaussian kernel, and $\phi = 100$.

Table 4.5: Clustering Performance for $m = 2, w_{i,j} = \exp^{-\phi \|\hat{\pi}_i - \hat{\pi}_j\|_2^2} l_{i,j}^{k(\infty)}$.

k nearest neighbour= 5					
Sample Size (n)	5000	10000	15000	20000	25000
RI (s.e)	0.889 (0.10)	0.975 (0.04)	0.992 (0.02)	0.996 (0.01)	0.998 (0.01)
ARI (s.e)	0.720 (0.20)	0.928 (0.12)	0.974 (0.06)	0.989 (0.04)	0.994 (0.03)
Prob. of True Recovery	0.187	0.644	0.821	0.922	0.960
k nearest neighbour= 3					
Sample Size (n)	5000	10000	15000	20000	25000
RI (s.e)	0.949 (0.07)	0.995 (0.02)	0.999 (0.01)	0.999 (0.005)	1 (0.004)
ARI (s.e)	0.860 (0.17)	0.984 (0.05)	0.995 (0.03)	0.998 (0.02)	0.999 (0.01)
Prob. of True Recovery	0.501	0.908	0.973	0.990	0.996

Table 4.6: Clustering Performance for Different Weight Choice and Sample Size for Simulation 2.

Weight Choice	Sample Size (n)	RI	ARI	Prob. of True Recovery
l_2 Distance, Gaussian Kernel	1000	0.940 (0.022)	0.816 (0.073)	0
	2000	0.984 (0.012)	0.954 (0.034)	0.14
l_∞ Distance, Exponential Kernel	1000	0.969 (0.020)	0.908 (0.059)	0.104
	2000	0.994 (0.009)	0.983 (0.025)	0.638
l_1 Distance, Exponential Kernel	1000	0.965 (0.020)	0.893 (0.060)	0.03
	2000	0.993 (0.009)	0.979 (0.025)	0.468

The second choice of weight is $w_{i,j} = \exp^{-\phi \|\hat{\pi}_i - \hat{\pi}_j\|_\infty} l_{i,j}^{k(\infty)}$, with $\phi = 10$. Note that here we have used the l_∞ distance to find the nearest neighbours, but instead of incorporating the Gaussian kernel, we have used the natural exponential decay. The third weight is similar to the second one, using l_1 distance instead of l_∞ . Here we used relatively smaller samples, $n = 1000$ and $n = 2000$, respectively. The results are displayed in table (4.6).

From the experiment, we can infer that the weight involving l_2 distance in the Gaussian kernel performs poorly compared to l_∞ or l_1 distance. Using l_∞ distance is especially effective in such scenarios, as it measures the maximum element-wise distance between two estimated transition probability vectors. We are then able to separate out two vectors that are not likely to be in the same cluster.

From this study, we can conclude that when the centroids are well-separated in in a small number of co-ordinates, use of l_∞ distance can provide the best possible result.

4.5 Real Data Analysis

Over time, deadly pandemics have been prevalent in different parts of the world. The most recent one is the outbreak of a novel coronavirus disease Covid-19 (SARS-COV-2), started in the Wuhan

province of China in November 2019. Later it was spread all over the world, and hit hard over the whole world. As of April 2022, there are 510 million Covid cases worldwide and more than 6.2 million fatal cases. The patients affected with Covid suffer from fever, headache, fatigue, respiratory problems, shortness of breath etc., these characteristics were first analyzed by Wu and McGoogan (2020). Most of the patients exhibit very mild to moderate symptoms, only 8–10% patients need hospitalization and the fatality rate is close to 1.5%. This is much lower than 10% fatality rate in the reported 8000 cases during SARS-COV-1 outbreak in 2002 in China. Another novel coronavirus outbreak, the Middle Eastern Respiratory Coronavirus (MERS-Covid) has affected the Arabian peninsula in 2012. There were only 2500 reported cases of MERS, with astonishingly high fatality rate of 35%. However, SARS-COV-2, or the Covid-19 is much more contagious than the other ones, as it has very mild symptoms and low fatality rate which make it easier for the virus to be transmitted through air.

After the emergence of Covid-19, people are familiar with the RT-PCR, rapid antigen and rapid PCR testing procedures. These are all lab based processes to detect the presence of coronavirus from saliva or mucus, collected from human. The RT-PCR test ends up in a Positive result if the sample contains the DNA of the pathogen. The collected samples, if fully available, are almost equal to the reference sequence. In that case, there is no need to fit models, we can easily match the collected sample with the available genome sequence, and identify the virus whose sequence is almost similar to the reference genome. But in practice, there are many occasions where only a part of the full sequence is available. For example, a covid patient might have co-morbidity, i.e. suffering from any other critical disease. When the sample is collected from that patient for Covid test, different pathogens are mixed to form a mutation of the original virus. Hence, we end up with a corrupted sample, and need to trim a large part of the full DNA sequence to get rid of that corrupted part. The challenge lies in that scenario, where we have to detect the nature of a virus from a partially retrieved sample.

In the time of covid outbreak, it is necessary to distinguish the covid patients from the other patients. Often the symptoms for many diseases are similar as covid. For example, in the Indian subcontinent, Dengue is very much prevalent in the summer and fall season. The symptoms include

high fever, fatigue, headache, vomiting etc, very similar to covid. Same goes for another disease Hepatitis B. For all these diseases, if the DNA sequences in the collected samples are not fully available, it is extremely difficult to classify them to their correct class of virus. Wrong detection of disease for the critical patients can lead to very serious problems including fatality. In this paper, we consider four different viruses in our study: SARS-CoV-2, MERS, Dengue and Hepatitis B for the classification purpose. In the data set, we have full DNA sequences of 500 individuals, suffering from any one of this four diseases. Then we trim a big part of the full sequence and use them as our partially available information. The main objective is to come up with a methodology so that the miss-classification rate is lower, even when very low proportion of the sequences are retained.

While analyzing DNA sequences, we use the sparse Markov model (SMM) developed in this chapter. The method is simple to understand and computationally efficient at the same time, as we demonstrated with the simulation study. This also helps us interpreting the results confidently, to be unfolded when we discuss the set-up and the results from our study.

4.5.1 Data Description

We have collected the sample for 500 individuals from NCBI database, 200 affected from SARS-CoV-2, 50 from MERS, 100 from Dengue and 150 from Hepatitis B over different time periods and different locations. NCBI database also contains reference genome sequence of every virus. These reference sequences represents an ideal genome structure of any particular virus species. Note that, a very minimal changes in the neucleotide sequence can lead to a very different strain of the same disease. We use this reference sequence for training purposes. The lengths of the reference genome sequences for SARS, MERS, Dengue and Hepatitis B are 29903, 30119, 10735 and 3542 respectively.

The pandemic MERS was mostly prevalent in the Arab countries in 2012, hence all the samples are collected in that time. For Dengue and Hepatitis B, the samples are distributed across the world in the last thirty years. We have been particularly careful in collecting the samples from COVID-19 disease so that we are able to incorporate different strains of that disease. To ensure that, we have used 50 samples each from four different time-frames: April 2020, September 2020, January 2021 and April 2021. These time-frames are selected based on the spread of a certain strain, or peak in

the Covid cases worldwide. For example, in April 2020, Covid cases were significantly increasing in the whole world, and the lockdown was imposed for the first time. In September and October 2020, south east Asian countries, and most importantly India reached its peak in the first wave of Covid cases. In January 2021, USA and a major part of the world have experienced the peak of the Beta strain. In April 2021, wave of the deadliest Delta strain hit hard on the Asian countries, and began to spread in the rest of the world. In this method of sample collection, we have tried to represent the Covid samples in a time-homogeneous manner.

4.5.2 Method

We have already mentioned that we want to classify the viruses from the partially retrieved genome sequences. For that, we first model the 4 reference genome sequences using our proposed SMM method. Next, we randomly select a continuous segment of the genome sequence for each sample, and then compute the likelihood of that segment under each of the 4 reference models. Suppose the i^{th} model is denoted by \hat{P}_i , $i = 1, 2, 3, 4$. For any given sequence $x = x_1 x_2 \dots x_n$, likelihood of x for each model is

$$L_i(x) = \hat{P}_i(\tilde{X}_m^{(m)} = \tilde{x}_m^{(m)}) \prod_{t=m+1}^n \hat{P}_i(X_{t+1} = x_{t+1} \mid \tilde{X}_t^{(m)} = \tilde{x}_t^{(m)}).$$

We then classify x to $\arg \max_{i=1, \dots, 4} L_i(x)$. Note that the transition probabilities inside the product term are estimated from the fitted model, along with $\hat{P}_i(\tilde{X}_m^{(m)} = \tilde{x}_m^{(m)})$. Thus, we can expect we can classify the true virus from a moderately large segment of the full RNA sequence of the respective viruses.

4.5.3 Results

In our analysis, we fit two different models. In the first model, we fit SMM-s of order $m = 4$ for SARS and MERS, while for the other two viruses, we use $m = 3$. The form of the weights are $w_{i,j} = \exp^{-\phi \|\hat{\pi}_i - \hat{\pi}_j\|_\infty^2} l_{i,j}^k$ where $l_{i,j}^k$ is the indicator function that $\hat{\pi}_i$ belongs to k nearest neighbour of $\hat{\pi}_j$ or vice versa in terms of l_∞ distance, for some $\phi > 0$. For the first two viruses, we take the number of

nearest neighbors (nn) to be 20, and for the later two cases they are 5. The value of $\phi = 100$ is fixed for all four cases. In the second model, we fit SMM-s of order $m = 3$ for all four viruses, with $\phi = 100$ and number of nearest neighbours to be 5.

Model 1

The number of clusters and its sizes for each model is presented in the following table (4.7). We also present the top twenty m -tuples in terms of the occurrences in the reference sequence for each 4 virus in the figure (4.3).

Table 4.7: Number of clusters and size of each cluster obtained in Model 1.

Virus	Number of Clusters	Cluster Size
Covid 19	28	155, 50, 10, 7, 5, 4, 2 (3 times), 1 (19 times)
MERS	30	141, 60, 6, 5 (3 times), 3 (3 times), 2 (4 times), 1 (17 times)
Dengue	7	24, 16, 14, 4, 3, 2, 1
Hepatitis B	14	41, 8, 4, 1 (11 times)

From the samples, we randomly choose segments of length $100\epsilon\%$, and compute the likelihoods under 4 models to classify it to the most likely class of virus. Three different values of ϵ have been used; 0.05, 0.1 and 0.25. We compute the overall mis-classification rates for all three cases, i.e. the proportion of virus that are wrongly classified. Apart from that, we also present the class-wise counts of the samples from a particular species in the 4×4 confusion matrices in the following table (4.8).

Model 2

Now we present the clustering performance for the second model, where all fitted SMM have order $m = 3$ in the table (4.9).

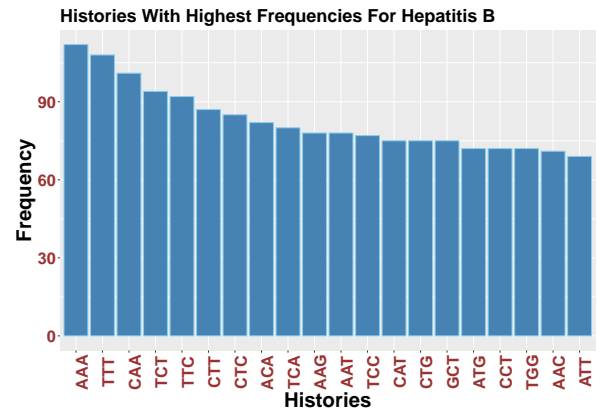
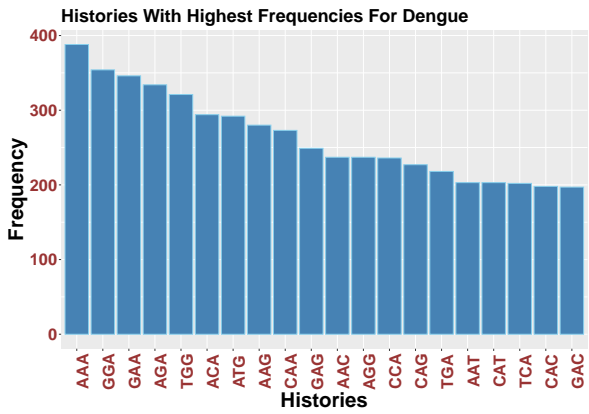
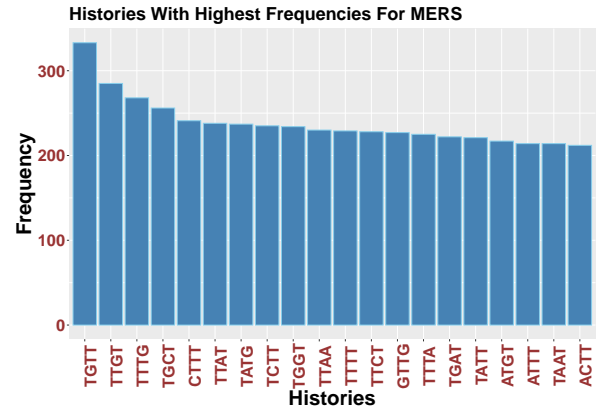
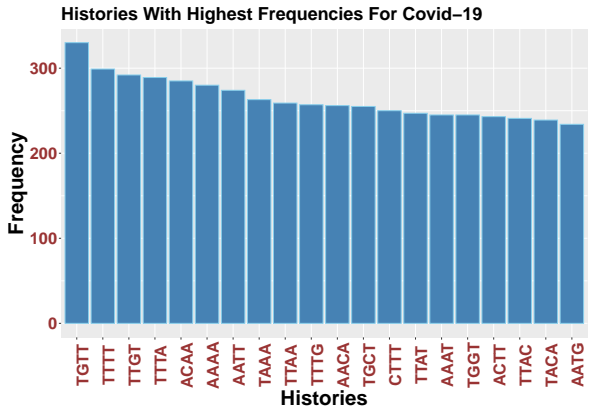


Figure 4.3: Histories with higher frequency in the reference sequence for each virus

Table 4.8: Confusion Matrices for $\epsilon = 0.05, 0.1$ and 0.25 respectively with mis-classification rates 22.6%, 15.2% and 2.8% in **Model 1**.

Observed \ Fitted	SARS-Cov-2	MERS	Dengue	Hepatitis B	Total
SARS-Cov-2	185	0	4	11	200
MERS	0	50	0	0	50
Dengue	0	0	100	0	100
Hepatitis B	17	44	37	52	150

Observed \ Fitted	SARS-Cov-2	MERS	Dengue	Hepatitis B	Total
SARS-Cov-2	193	0	0	7	200
MERS	0	50	0	0	50
Dengue	0	0	100	0	100
Hepatitis B	6	39	24	81	150

Observed \ Fitted	SARS-Cov-2	MERS	Dengue	Hepatitis B	Total
SARS-Cov-2	194	0	0	6	200
MERS	0	50	0	0	50
Dengue	0	0	100	0	100
Hepatitis B	0	8	0	142	150

Table 4.9: Confusion Matrices for $\epsilon = 0.05, 0.1$ and 0.25 respectively with mis-classification rates 22%, 16% and 4% in **Model 2**.

Observed \ Fitted	SARS-Cov-2	MERS	Dengue	Hepatitis B	Total
SARS-Cov-2	177	8	4	11	200
MERS	1	49	0	0	50
Dengue	0	0	100	0	100
Hepatitis B	11	30	45	64	150

Observed \ Fitted	SARS-Cov-2	MERS	Dengue	Hepatitis B	Total
SARS-Cov-2	187	6	0	7	200
MERS	0	50	0	0	50
Dengue	0	0	100	0	100
Hepatitis B	4	37	27	82	150

Observed \ Fitted	SARS-Cov-2	MERS	Dengue	Hepatitis B	Total
SARS-Cov-2	190	2	0	8	200
MERS	0	50	0	0	50
Dengue	0	0	100	0	100
Hepatitis B	0	10	0	140	150

4.5.4 Discussion

In any statistical analysis, more sample size helps us designing a sensible methodology. In our experiment as well, if the length of a sample segment from the full sequence is very short, the conclusions will not be reliable. From the tables, it is clear that $\epsilon = 0.05$, i.e. when only 5% of the original sample sequences are retained, the miss-classification rate among the Hepatitis B samples are high. This is completely justified, since for Hepatitis B samples, the length of selected segments are about 170, whereas that lengths are about 500 for Dengue and 1500 for MERS and SARS. As we increase the proportion, the performance naturally improves. The overall miss-classification rates are 0.228, 0.152 and 0.032 for $\epsilon = 0.05$, 0.10 and 0.25 respectively. So, with only 25% of the sequences, we can correctly identify the true virus for more than 95% of the cases. Even within Hepatitis B, the miss-classification error reduces drastically once we have fairly long sequence so that meaningful inference could be made. For the other three viruses, the samples are rarely miss-classified.

The most important question in our modelling of SMM is that how we choose the order m of the Markov chains. Clearly, higher the m is, we get more detailed information from the model. On the other hand, it is not hard to think that if we choose m too large, say $m = 10$, we won't get much information from the histories. Even for most of the histories, we won't have even a single observation. Hence, choosing the orders based on the lengths of the reference sequences would be ideal, so that every m -tuple has significant number of observations. The lengths of the reference sequences of SARS and MERS are close to 30000, hence we use $m = 4$ in these cases. The lengths of reference sequences of Hepatitis B and Dengue are relatively shorter, so we think $m = 3$ would be a better fit for the modelling purpose. Apart from the numerical point of view, there is a biological significance for using $m \geq 3$ as well. If one looks into the structure of any DNA or RNA sequence, three consecutive DNA bases form a "codon", which translates a genetic code into a sequence of amino acids or a stop codon. Although there are 64 many possible codons, the number of amino acids is 20, where each amino acid can correspond to multiple codons. Rest of the codons are named as stop codons, which signals the end of protein synthesis. These amino acids molecules combine to form proteins, which are the basic building blocks of life. So, it is fair to assume that SMM of order 3 or more will be able to explain the structure of a virus.

While sampling, we have selected the snippets from any part of the full sample sequences. All these viral infections have similar symptoms. So it is obvious that there are some similarity in the sequence structure in some parts of these viruses. There might be a particular sequence of amino acids in the genome structure which construct the spike proteins. Very short length of the chains thus be misleading from this point of view as well. We might end up with a short segment of one virus which is similar to a segment of another one. In the case when we have a sample sequence of moderate length so that the sequence is not overlapping with other viruses, the classification will be really good, as we can see from the results. MERS and Dengue samples are never miss-classified to other classes, even with $\epsilon = 0.05$. This phenomenon can be justified by the fact that our SMM method utilizes the information from the reference genome sequence in a compact manner, so that it can capture the diversity of genome structure from different parts of the samples. For example, the frequency of an m -tuple can be spread over the whole sequence, or may be concentrated in a particular region. In either case, the estimated transition probabilities will help us calculating the likelihoods under each model, and selecting the true virus with high probability. Overall, this method is successful in such classification problems, which opens up a scope for a broader research in this area.

4.6 Summary

The proposed method of fitting sparse Markov model can be utilized in many different areas. Our methodology provides great computational efficiency and clustering accuracy. The novelty of our approach is that we have built a completely data dependent methodology for fitting SMM, without using any prior knowledge of the number of clusters or the transition probabilities. The theoretical results guarantee that for large sample sizes, we are able to reconstruct the true clusters with probability tending to 1, which reduces the number of parameters in an efficient manner. As a real data application, we study the structure of the RNA sequence of a virus to classify an unknown partial sample to the most likely class. Even for two very similar type of viruses, our method can separate them, even for moderate sequence length. Even for different variants of same disease, we

can use a single reference sequence for training purpose, reflecting the flexibility of our approach.
The methodology then has wide applicability.

FITTING SPARSE MARKOV MODEL BY
GENERALIZED CONVEX CLUSTERING
ALGORITHM

5.1 Introduction

Among all the statistical tools in unsupervised learning, clustering is probably the most fundamental problem. There are many traditional methods of clustering including K -means, mean-shift algorithm, hierarchical clustering or spectral clustering . The major problem of these algorithms is that they suffer from finding the global minimum for the objective functions. The solution depends on the initial values which often gets stuck in a local minima. Apart from that, the total number of clusters need to be fixed beforehand for executing these algorithms. Hence, one needs to pre-specify

the number of clusters and execute the clustering methods for different cluster numbers. Needless to say this makes the problem computationally challenging, especially when there are a large number of data points.

To overcome these problems, Pelckmans et al. (2005), Lindsten et al. (2011) and Hocking et al. (2011) have transformed the clustering problem as a convex optimization problem. In such formulation given the data points $\mathbf{x}_1, \dots, \mathbf{x}_n \in \mathbb{R}^p$ they have proposed to minimize the following convex optimization criteria

$$F_\lambda(\mathbf{B}) = \frac{1}{2} \sum_{i=1}^n \|\mathbf{x}_i - \mathbf{b}_i\|_2^2 + \lambda \sum_{i < j} w_{i,j} \|\mathbf{b}_i - \mathbf{b}_j\| \quad (5.1.1)$$

for some tuning parameter $\lambda > 0$ and non-negative weights $w_{i,j} = w_{j,i}$ where \mathbf{B} is the matrix containing the vectors \mathbf{b}_i -s. The rationale behind this approach is that if the data points \mathbf{x}_i and \mathbf{x}_j belong to the same cluster, their corresponding centers \mathbf{b}_i and \mathbf{b}_j should be the same. The second term acts as a regularization quantity which enforces the difference $\mathbf{b}_i - \mathbf{b}_j$ to be zero depending on the type of the norm $\|\cdot\|$ and the tuning parameter λ . The biggest advantage of this method is it gives us a solution paths to identify the clusters over a different range of λ , from which we can choose the most suitable model based on some model selection criteria, e.g. AIC or BIC. This formulation not only handles the problem of finding the global minimum, it doesn't require us to fix the number of clusters beforehand.

Many different algorithms have been proposed to minimize the objective function (5.1.1). Lindsten et al. (2011) have used an off-the-shelf convex solver, named CVX to solve the clustering problem. Hocking et al. (2011) have used the idea of fused lasso to develop a few algorithms for different types of norms in the penalization term, including ℓ_1 and ℓ_2 norms. Chi and Lange (2015) have viewed this convex optimization problem as a constrained optimization problem, and hence used popular variable splitting algorithms like alternating direction method of multipliers (ADMM) or alternating minimization algorithm (AMA) for solving the convex clustering problem. Sun et al. (2021) have introduced a semismooth Newton based augmented Lagrangian method (SS-NAL) to solve the convex optimization problem. Apart from different computational approaches of solving the problem, some theoretical properties of this method regarding cluster recovery conditions have been developed by Zhu et al. (2014), Panahi et al. (2017) and Sun et al. (2021).

The computational benefits and easy interpretation of convex clustering method are useful in many practical contexts beyond the standard clustering problems. Following the similar ideas, Majumder et al. (2022) have developed a data based method of fitting sparse Markov models for modelling higher-order Markov models. Suppose X_1, \dots, X_n be a categorical time series over a state space Σ with $|\Sigma| = d$. If one wants to fit a Markov chain of order m , there are total $|\Sigma|^m(|\Sigma| - 1)$ many parameters that need to be estimated. Even for moderate m and Σ , this task is computationally challenging. There are many model reduction techniques have been proposed in literature, e.g. variable length Markov models (VLMC) by Rissanen (1983), Bühlmann et al. (1999) or sparse Markov models by Garcia et al. (2011), Xiong et al. (2016), Jääskinen et al. (2014) or Majumder et al. (2022). For a sparse Markov model, the set of all m -th order histories Σ^m is partitioned into a certain number of groups k_0 , where the transition probabilities are same for two histories belonging to the same partition. This reduces the total number of parameters to $k_0(|\Sigma| - 1)$ which is feasible to deal with.

Majumder et al. (2022) have utilized the convex clustering algorithm by Chi and Lange (2015) to estimate the partitions of Σ^m by clustering the empirical transition probability vectors $\hat{\pi}_i$ -s for the histories $i = 1, \dots, |\Sigma|^m$ as the data points. If $p = |\Sigma|^m$, then Majumder et al. (2022) proposed to minimize the objective function

$$F_\lambda(\mathbf{B}) = \frac{1}{2} \sum_{i=1}^n \|\hat{\pi}_i - \mathbf{b}_i\|_2^2 + \lambda \sum_{i < j} w_{i,j} \|\mathbf{b}_i - \mathbf{b}_j\| \quad (5.1.2)$$

over $\mathbf{b}_1, \dots, \mathbf{b}_p \in \mathbb{S}^d$, where \mathbb{S}^d is the d -dimensional simplex. Using the intrinsic computational steps, Majumder et al. (2022) have shown that minimizing $F_\lambda(\mathbf{B})$ without restricting \mathbf{b}_i to \mathbb{S}^d indeed produces the optimum solution in \mathbb{S}^d itself, unaltered the convex clustering approach by Chi and Lange (2015). Once the partitions are identified, it is easy to estimate the common transition probabilities. Each value of λ will lead to a specific model, Majumder et al. (2022) proposed to choose the optimum model which minimizes the BIC criteria.

So far, the developments of convex clustering algorithms consider a squared error loss between the data points and the parameters \mathbf{b}_i . However, there are many scenarios where some other loss functions could be useful. For example, a common disparity measure between two probability

distribution is the Kullback-Leibler Divergence (KLD), which possesses some nice properties. This type of loss functions could be useful in our SMM set-up where we eventually have to deal with disparities between probability vectors. Another such measure of divergence is the density power divergence (DPD), introduced by Basu et al. (1998) which produces robust estimators under data contamination. Hence it will be worthwhile to develop the convex clustering algorithms for other loss functions as well. For feature selection in mixed multi-view data, Wang and Allen (2021) have developed an integrative generalized convex clustering method, namely “iGecco+” which deals with different type of differentiable and non-differentiable loss functions for various data types; including Bernoulli log-likelihood or multinomial log-likelihood in binary and categorical data, or Poisson log-likelihood for analyzing count data. Thus, it will be worthwhile to explore how changing the loss functions for fitting SMM make an impact on the computational aspects.

In this chapter, we propose several SMM fitting algorithms for convex loss functions and adjust our method depending on the strong convexity. The major challenge in such cases is that we may not always have a closed-form solution in every steps of the algorithms. Not only that, extra computational burden arise from restricting $\mathbf{b}_i \in \mathbb{S}^d$ as the optimum solution might not be a transition probability vector unlike the minimizer of equation (5.1.2). In our method, we incorporate such extra constraints, and update the computation accordingly. Extensive simulation studies have been conducted to compare the performances and the time complexity under different loss functions. We also proposed a more relaxed approach of solving the original convex clustering problem (5.1.1), which performs comparably in terms of finding the true partition in much less time.

The rest of the chapter is organized as follows. In Section 5.2, we develop the ADMM and AMA algorithm for general loss functions. In Section 5.3, we apply our algorithm to the density power divergence. Theoretical cluster recovery conditions are explicitly presented in Section 5.4. We propose a more relaxed method of solving the original convex clustering problem in Section 5.5. Extensive simulation studies have been performed in Section 5.6, both the time complexity and clustering accuracy have been compared across different loss functions. We finish this chapter in Section 5.7, summarizing the key findings and possible future applications of our method.

5.2 Fitting SMM Using Generalized Loss Function

Before going into the details of the algorithm, let us define some notations. Let X_1, \dots, X_n be a categorical time series over the state space Σ with $|\Sigma| = d$. The order of the fitted Markov chain over this sequence is m , and denote the elements of $\Sigma^m = \{\sigma_1, \dots, \sigma_p\}$ where $p = d^m$. For any set $S \subset \Sigma^m$, $N_S = \sum_{t=m}^{n-1} \mathcal{I}(\tilde{X}_t^{(m)} \in S)$, $N_{S,a} = \sum_{t=m}^{n-1} \mathcal{I}(\tilde{X}_t^{(m)} \in S, X_{t+1} = a)$ for $a \in \Sigma$, where $\tilde{X}_t^{(m)} = (X_t, X_{t-1}, \dots, X_{t-m+1})$. The elements of empirical transition probability vector $\hat{\pi}_i$ are defined by $\hat{\pi}_{i,a} = N_{\sigma_i,a} / N_{\sigma_i}$ for $i = 1, \dots, p$.

The equation (5.1.2) deals with squared error loss function. However as we discussed earlier, there are other loss functions which can be useful in our context. Hence we can generalize the above set-up as follows.

$$\sum_{j=1}^p \delta_j \rho(\hat{\pi}_j, \mathbf{b}_j) + \lambda \sum_{i < j} w_{i,j} \|\mathbf{b}_i - \mathbf{b}_j\|_2. \quad (5.2.3)$$

for $\mathbf{b}_j \in \mathbb{S}^d$, \mathbb{S}^d being the d dimensional simplex, and for some constants $\delta_j > 0$. These constants δ_j will determine the amount of penalization we should allow for each observation specific loss $\rho(\hat{\pi}_j, \mathbf{b}_j)$. The major challenge for such general loss function ρ is to develop similar ADMM or AMA updates. Even though AMA is a much faster algorithm, it is only applicable when ρ is a strongly convex function. In addition to that, we may not have a closed form solution for each parameter in every step, as noticed by Wang and Allen (2021). In such cases we may have to deal with some kind of approximation that simplifies the algorithm. In this section, we first discuss the ADMM algorithm for a general loss function ρ , then modify the algorithm to AMA when ρ is strongly convex.

Like all other convex clustering algorithms, we need to view the optimization problem (5.2.3) as the following constrained optimization problem

$$\begin{aligned} & \min \sum_{j=1}^p \delta_j \rho(\hat{\pi}_j, \mathbf{b}_j) + \lambda \sum_{l \in \mathcal{E}} w_l \|\mathbf{v}_l\|_2 \\ & \text{subject to } \mathbf{b}_{l_1} - \mathbf{b}_{l_2} - \mathbf{v}_l = 0; \quad l \in \mathcal{E} \\ & \mathbf{1}^T \mathbf{b}_j = 1; \quad j = 1, \dots, p \end{aligned} \quad (5.2.4)$$

and the corresponding augmented Lagrangian is given by

$$\begin{aligned} \mathcal{L}_\nu(\mathbf{B}, \mathbf{V}, \mathbf{\Gamma}, \mathbf{\Phi}) = & \sum_{j=1}^p \delta_j \rho(\hat{\boldsymbol{\pi}}_j, \mathbf{b}_j) + \lambda \sum_{l \in \mathcal{E}} w_l \|\mathbf{v}_l\|_2 + \sum_{l \in \mathcal{E}} \langle \gamma_l, \mathbf{v}_l - \mathbf{b}_{l_1} + \mathbf{b}_{l_2} \rangle + \sum_{j=1}^p \phi_j (1 - \mathbf{1}^T \mathbf{b}_j) \\ & + \frac{\nu}{2} \sum_{l \in \mathcal{E}} \|\mathbf{v}_l - \mathbf{b}_{l_1} + \mathbf{b}_{l_2}\|_2^2 + \frac{\nu}{2} \sum_{j=1}^p (1 - \mathbf{1}^T \mathbf{b}_j)^2. \end{aligned} \quad (5.2.5)$$

5.2.1 ADMM Updates

Setting the differential of the objective function in (5.2.5) w.r.t \mathbf{b}_j for all $j = 1, \dots, p$ to 0 would give us the updates of \mathbf{b}_j , given other variables. Hence, we want to solve the following equation (5.2.6) to update the variable \mathbf{b}_j , for all $j = 1, \dots, p$; given the variables $\mathbf{V}, \mathbf{\Gamma}$.

$$\begin{aligned} \nabla_j \mathcal{L}_\nu(\mathbf{B}, \mathbf{V}, \mathbf{\Gamma}, \mathbf{\Phi}) = & \delta_j \nabla \rho(\hat{\boldsymbol{\pi}}_j, \mathbf{b}_j) - \sum_{l_1=j} \gamma_{l_1} + \sum_{l_2=j} \gamma_{l_2} - \phi_j \mathbf{1} \\ & - \nu \sum_{l_1=j} (\mathbf{v}_{l_1} - \mathbf{b}_{l_1} + \mathbf{b}_{l_2}) + \nu \sum_{l_2=j} (\mathbf{v}_{l_2} - \mathbf{b}_{l_2} + \mathbf{b}_{l_1}) - \nu (1 - \mathbf{1}^T \mathbf{b}_j) \mathbf{1} = 0 \end{aligned} \quad (5.2.6)$$

Note that, the closed form of \mathbf{b}_j while solving (5.2.6) may not be available in most of the cases. If $\rho(\hat{\boldsymbol{\pi}}_j, \mathbf{b}_j) = \frac{1}{2} \|\hat{\boldsymbol{\pi}}_j - \mathbf{b}_j\|_2^2$ and $\delta_j = 1$, the case reduces to the usual convex clustering method developed in Chi and Lange (2015). In this scenario, the closed form for the solution of \mathbf{b}_j in each step can be determined for solving the equation (5.2.6), which boils down to solving a set of linear equations in this particular case. It was shown by Majumder et al. (2022) that for such squared error loss, we do not even need to incorporate the extra Lagrangian terms for the constraints $\mathbf{1}^T \mathbf{b}_j = 1$, as the resulting solution exhibit this property by its own. However, for general loss function, this may not be true, and hence the extra computational burden involving extra linear constraints should be executed.

In case when the closed form solution is not available, one may want to solve the equation (5.2.6) iteratively in each step. One popular approach is the use of Newton-Raphson method, which is really fast to find root of an equation especially when there exists an unique root. After some basic

calculations, the update of \mathbf{b}_j in the $(t+1)^{th}$ step is the root of the following equation:

$$\begin{aligned} & \delta_j \nabla \rho(\hat{\pi}_j, \mathbf{b}_j) + \nu(p-1)\mathbf{b}_j + \nu(\mathbf{1}^T \mathbf{b}_j)\mathbf{1} \\ & - \sum_{l_1=j} \gamma_{l_1}^{(t)} + \sum_{l_2=j} \gamma_{l_2}^{(t)} - \nu \sum_{l_1=j} \mathbf{v}_{l_1}^{(t)} + \nu \sum_{l_2=j} \mathbf{v}_{l_2}^{(t)} - (\phi_j^{(t)} + \nu)\mathbf{1} - \nu \sum_{i \neq j} \mathbf{b}_i^{(t)} = 0 \end{aligned} \quad (5.2.7)$$

The Newton-Raphson method for solving the above equation (5.2.7) is outlined in the following algorithm (2). Note that, for \mathbf{b}_j to be a valid discrete probability distribution, $b_{j,a} \geq 0$. However, we have not included that constraint in our optimization problem. For that, we are going to state the following assumption:

(C1) There is always an unique solution $\mathbf{b}_j^{(t+1)} \geq \mathbf{0}$ to the equation (5.2.7).

Later on, we will show that these conditions are indeed satisfied with some common class of divergence measure ρ .

Algorithm 2 Newton-Raphson for Updating \mathbf{b}_j

Input: $\gamma_l^{(t)}, \mathbf{v}_l^{(t)}, \mathbf{b}_i^{(t)}$ for $l \in \mathcal{E}, i = 1, \dots, p$

Initialize: $\mathbf{y} = \mathbf{b}_j^{(t)}$

1: **while** Not Converged **do**

$$\begin{aligned} \mathbf{y} = & \mathbf{y} - \left(\delta_j \nabla^2 \rho(\hat{\pi}_j, \mathbf{y}) + \nu(p-1)\mathbf{I} + \nu \mathbf{1} \mathbf{1}^T \right)^{-1} \\ & \left(\delta_j \nabla \rho(\hat{\pi}_j, \mathbf{y}) - \nu \sum_{l_1=j} \mathbf{v}_{l_1}^{(t)} + \nu \sum_{l_2=j} \mathbf{v}_{l_2}^{(t)} - \nu \sum_{i \neq j} \mathbf{b}_i^{(t)} - \sum_{l_1=j} \gamma_{l_1}^{(t)} + \sum_{l_2=j} \gamma_{l_2}^{(t)} - (\phi_j^{(t)} + \nu)\mathbf{1} \right) \end{aligned}$$

2: **end while**

3: $\mathbf{b}_j^{(t+1)} = \mathbf{y}$

The major disadvantage of using ADMM in such problems is that we need to carry out a inner nested loop to update certain variables in each step, which significantly increases the computational burden. In literature, many different versions of ADMM have been proposed to address this problem. These updated algorithms provide some kind of inexact or one-step update of the variables without fully solving the objective function, making them a lot faster. Such modifications include generalized

ADMM by Deng and Yin (2016) and proximal ADMM by Shefi and Teboulle (2014) or Banert et al. (2016). In the convex clustering scenario, a similar approach was taken by Wang and Allen (2021) for clustering multi view data using general loss functions. The objective function in their problem was different, and they had to use one-step proximal gradient updates for updating certain variables. In our case, we propose to use the one-step Newton-Raphson update of solving the sub-problem of updating \mathbf{b}_j in every iteration. The partial differentiability of $\mathcal{L}_\nu(\mathbf{B}, \mathbf{V}, \mathbf{\Gamma})$ w.r.t \mathbf{b}_j enables us to use the Newton-Raphson method, which is much faster than proximal gradient or gradient descent methods. Most importantly, this algorithm is guaranteed to converge, which we will prove later. We name the original ADMM algorithm, which try to compute the updates of the variables exactly through inner nested loop as **Sparse Markov Model by Generalized Convex Clustering Optimization** (SMMGECCO). Consecutively, the inexact ADMM update using the one-step Newton-Raphson update is named as SMMGECCO+. The complete steps for this optimization problem providing the updates of \mathbf{B}, \mathbf{V} and $\mathbf{\Gamma}$ are outlined in the following algorithms (3) and (4).

Algorithm 3 SMMGECCO Algorithm for General Convex Function

Initialize: $\mathbf{B}^{(0)}, \mathbf{V}^{(0)}, \mathbf{\Gamma}^{(0)}, \Phi^{(0)}, t = 0$.

- 1: **while** Not Converged **do**
- 2: $\mathbf{b}_j^{(t+1)}$ = solution of equation (5.2.7) using Algorithm (2).
- 3: $\mathbf{v}_l^{(t+1)} = \text{prox}_{\sigma_l \|\cdot\|_2} \left(\mathbf{b}_{l_1}^{(t+1)} - \mathbf{b}_{l_2}^{(t+1)} - \frac{1}{\nu} \gamma_l^{(t)} \right)$;
- 4: $\gamma_l^{(t+1)} = \gamma_l^{(t)} + \nu \left(\mathbf{v}_l^{(t+1)} - \mathbf{b}_{l_1}^{(t+1)} + \mathbf{b}_{l_2}^{(t+1)} \right)$;
- 5: $\phi_j^{(t+1)} = \phi_j^{(t)} + \nu \left(\mathbf{1} - \mathbf{1}^T \mathbf{b}_j^{(t+1)} \right)$;
- 6: $t = t + 1$.
- 7: **end while**

If ρ is a separable function of the elements of \mathbf{b}_j , whose Hessian is a diagonal matrix, it is really cheap in terms of computational complexity to invert the Hessian in each step. For example, if $\rho(\hat{\pi}_j, \mathbf{b}_j) = -\sum_{a=1}^d \hat{\pi}_{j,a} \log(\hat{\pi}_{j,a}/b_{j,a})$, i.e. the Kullback-Leibler divergence, then

$$\nabla^2 \rho(\hat{\pi}_j, \mathbf{b}_j) = \text{diag} \left(\frac{\hat{\pi}_{j,1}}{b_{j,1}^2}, \dots, \frac{\hat{\pi}_{j,d}}{b_{j,d}^2} \right).$$

Algorithm 4 SMMGECCO+ Algorithm for General Convex Function

Initialize: $\mathbf{B}^{(0)}, \mathbf{V}^{(0)}, \mathbf{\Gamma}^{(0)}, \Phi^{(0)}, t = 0$.

1: **while** Not Converged **do**

2:

$$\begin{aligned} \mathbf{b}_j^{(t+1)} &= \mathbf{b}_j^{(t)} - \left(\nabla_{j,j}^2 \mathcal{L}_\nu(\mathbf{B}^{(t)}, \mathbf{V}^{(t)}, \mathbf{\Gamma}^{(t)}, \Phi^{(t)}) \right)^{-1} \nabla_j \mathcal{L}_\nu(\mathbf{B}^{(t)}, \mathbf{V}^{(t)}, \mathbf{\Gamma}^{(t)}, \Phi^{(t)}) \\ &= \mathbf{b}_j^{(t)} - \left(\delta_j \nabla^2 \rho(\hat{\pi}_j, \mathbf{b}_j^{(t)}) + \nu(p-1)\mathbf{I} + \nu \mathbf{1} \mathbf{1}^T \right)^{-1} \\ &\quad \left(\delta_j \nabla \rho(\hat{\pi}_j, \mathbf{b}_j^{(t)}) - \nu \sum_{l_1=j} \mathbf{v}_l^{(t)} + \nu \sum_{l_2=j} \mathbf{v}_l^{(t)} - \nu \sum_{i \neq j} \mathbf{b}_i^{(t)} - \sum_{l_1=j} \gamma_l^{(t)} + \sum_{l_2=j} \gamma_l^{(t)} - (\phi_j^{(t)} + \nu) \mathbf{1} \right) \end{aligned}$$

3: $\mathbf{v}_l^{(t+1)} = \text{prox}_{\sigma_l \|\cdot\|_2} \left(\mathbf{b}_{l_1}^{(t+1)} - \mathbf{b}_{l_2}^{(t+1)} - \frac{1}{\nu} \gamma_l^{(t)} \right);$

4: $\gamma_l^{(t+1)} = \gamma_l^{(t)} + \nu \left(\mathbf{v}_l^{(t+1)} - \mathbf{b}_{l_1}^{(t+1)} + \mathbf{b}_{l_2}^{(t+1)} \right);$

5: $\phi_j^{(t+1)} = \phi_j^{(t)} + \nu \left(\mathbf{1} - \mathbf{1}^T \mathbf{b}_j^{(t+1)} \right);$

6: $t = t + 1.$

7: **end while**

In such cases, the inverse of the Hessian matrix of \mathcal{L}_ν w.r.t. \mathbf{b}_j is

$$\left(\delta_j \nabla^2 \rho(\hat{\pi}_j, \mathbf{b}_j) + \nu(p-1)\mathbf{I} + \nu \mathbf{1} \mathbf{1}^T \right)^{-1} = \left(D_j + \nu \mathbf{1} \mathbf{1}^T \right)^{-1} = D_j^{-1} - \nu \frac{D_j^{-1} \mathbf{1} \mathbf{1}^T D_j^{-1}}{1 + \nu \mathbf{1}^T D_j^{-1} \mathbf{1}}$$

using Sherman-Morrisson-Woodbury formula, where $D_j = \delta_j \nabla^2 \rho(\hat{\pi}_j, \mathbf{b}_j) + \nu(p-1)\mathbf{I}$. Clearly, D_j is a diagonal matrix, and hence

$$D_j^{-1} = \frac{1}{\delta_j} \left(\nabla^2 \rho(\hat{\pi}_j, \mathbf{b}_j) \right)^{-1} + \frac{1}{\nu(p-1)} \mathbf{I}.$$

Hence, inverting the $d \times d$ Diagonal matrix can be carried out in $\mathcal{O}(d)$ many operations, and the computation of the Hessian matrix would require $\mathcal{O}(d^2)$ many operations. This is computationally feasible in each step when d is not very large, and we expect good convergence result.

5.2.2 AMA Updates

When ρ is strongly convex, one can use the AMA algorithm for solving the original optimization problem (5.2.4). In this algorithm, the variables \mathbf{b}_j -s are updated by letting the tuning parameter

$\nu = 0$, and hence we need to solve the following equation

$$\delta_j \nabla \rho(\hat{\boldsymbol{\pi}}_j, \mathbf{b}_j) - \sum_{l_1=j} \gamma_{l_1}^{(t)} + \sum_{l_2=j} \gamma_{l_2}^{(t)} - \phi_j^{(t)} \mathbf{1} = \mathbf{0}. \quad (5.2.8)$$

We can find the root of the equation (5.2.8) using the Newton-Raphson algorithm outlined in the algorithm (2), with $\nu = 0$. Just like in ADMM case, there is no guarantee that the solution of (5.2.8) satisfies $\mathbf{b}_j^{(t+1)} \geq \mathbf{0}$. However, if we observe carefully, $\mathbf{b}_j^{(t+1)}$ is the minimizer of the function $f(\mathbf{b}_j) = \delta_j \rho(\hat{\boldsymbol{\pi}}_j, \mathbf{b}_j) - \left(\sum_{l_1=j} \gamma_{l_1}^{(t)} - \sum_{l_2=j} \gamma_{l_2}^{(t)} + \phi_j^{(t)} \mathbf{1} \right)^T \mathbf{b}_j$, and for any separable convex function ρ , $b_{j,a}^{(t+1)}$ is the minimizer of $f_a(b_{j,a}) = \delta_j \rho^{(a)}(\hat{\boldsymbol{\pi}}_{j,a}, b_{j,a}) - \left(\sum_{l_1=j} \gamma_{l_1,a}^{(t)} - \sum_{l_2=j} \gamma_{l_2,a}^{(t)} + \phi_j^{(t)} \right) b_{j,a}$. Since ρ is strongly convex, $f_a''(b_{j,a}) > 0$, making $f_a'(b_{j,a})$ a strictly increasing function. Also, $f_a'(b_{j,a}) \rightarrow \infty$ if $b_{j,a} \rightarrow \infty$, hence the only way $f_a'(b_{j,a}) = 0$ won't have a solution for $b_{j,a} > 0$ if $f_a'(b_{j,a}) > 0$, which implies f_a is an increasing function on $[0, \infty)$. Hence $f_a(b_{j,a})$ is minimized for $b_{j,a} = 0$, proving that the equation (5.2.8) will have a solution in $[0, \infty)$.

Once we update \mathbf{b}_j , we seek to update \mathbf{v}_l and γ_l sequentially. However, as shown by Chi and Lange (2015), we can bypass updating \mathbf{v}_l , and directly update γ_l using the linear relation among themselves, which saves much time. The sequential updates of γ_l are as follows:

$$\gamma_l^{(t+1)} = \mathcal{P}_{C_l}(\gamma_l^{(t)} - \nu \mathbf{g}_l^{(t+1)})$$

where $\mathbf{g}_l^{(t+1)} = \mathbf{b}_{l_1}^{(t+1)} - \mathbf{b}_{l_2}^{(t+1)}$, $C_l = \{\gamma_l : \|\gamma_l\|_2 \leq \lambda w_l\}$, and $\mathcal{P}_A(\mathbf{x})$ is the projection of \mathbf{x} onto the set A . The full SMMGECCO algorithm for strongly convex ρ is outlined in (5).

Algorithm 5 SMMGECCO Algorithm for Strongly Convex Function

Initialize: $\mathbf{B}^{(0)}, \mathbf{V}^{(0)}, \boldsymbol{\Gamma}^{(0)}, \boldsymbol{\Phi}^{(0)}$, $t = 0$.

1: **while** Not Converged **do**

2: $\mathbf{b}_j^{(t+1)}$ = solution of the equation (5.2.8), determined by the Newton-Raphson in algorithm (2) with $\nu = 0$.

3: $\gamma_l^{(t+1)} = \mathcal{P}_{C_l}(\gamma_l^{(t)} - \nu \mathbf{g}_l^{(t+1)})$;

4: $\phi_j^{(t+1)} = \phi_j^{(t)} + \nu(1 - \mathbf{1}^T \mathbf{b}_j^{(t+1)})$;

5: $t = t + 1$.

6: **end while**

Similarly, as we did in ADMM case, we may want to extend the SMMGECCO+ in this case by taking one-step Newton-Raphson update for \mathbf{b}_j . We describe the steps in the algorithm (6). However, we might not use that process much because in practice, the loss functions with which we deal with are nice smooth functions. Since Newton-Raphson converges very fast, it won't take too many inner iterations to solve the corresponding equation (5.2.8) and hence the convergence of the algorithm could be established as well. In the next section, we will demonstrate all these algorithms for a certain class of loss functions in details.

Algorithm 6 SMMGECCO+ Algorithm Algorithm for Strongly Convex Loss Function

Initialize: $\mathbf{B}^{(0)}, \mathbf{V}^{(0)}, \mathbf{\Gamma}^{(0)}, \Phi^{(0)}, \Phi^{(0)}, t = 0.$

1: **while** Not Converged **do**

2: $\mathbf{b}_j^{(t+1)} = \mathbf{b}_j^{(t)} - \left(\delta_j \nabla^2 \rho(\hat{\pi}_j, \mathbf{b}_j^{(t)}) \right)^{-1} \left(\delta_j \nabla \rho(\hat{\pi}_j, \mathbf{b}_j^{(t)}) - \sum_{l_1=j} \gamma_{l_1}^{(t)} + \sum_{l_2=j} \gamma_{l_2}^{(t)} - \phi_j^{(t)} \mathbf{1} \right)$

3: $\gamma_{l_1}^{(t+1)} = \mathcal{P}_{C_l}(\gamma_{l_1}^{(t)} - \nu \mathbf{g}_{l_1}^{(t+1)});$

4: $\phi_j^{(t+1)} = \phi_j^{(t)} + \nu(1 - \mathbf{1}^T \mathbf{b}_j^{(t+1)});$

5: $t = t + 1.$

6: **end while**

5.3 Application of SMMGECCO to Density Power Divergence

So far, we have described the ADMM or AMA algorithm in the context of general convex clustering problem. In this section, we will explicitly simplify these methods for a common class of divergence measure called the density power divergence (dpd), introduced by Basu et al. (1998). For two discrete probability vectors $\mathbf{x}, \mathbf{y} \in \mathbb{S}^d$, the dpd is defined as

$$\Omega_\mu(\mathbf{x}, \mathbf{y}) = \sum_{a=1}^d \left\{ y_a^{1+\mu} - \left(1 + \frac{1}{\mu}\right) y_a^\mu x_a + \frac{1}{\mu} x_a^{1+\mu} \right\}$$

for some $\mu > 0$. This divergence measure is widely used as a robust loss function as an alternative to maximum likelihood estimation, exhibiting robust properties under data contamination. Note that for $\mu \rightarrow 0$, $\Omega_\mu(\mathbf{x}, \mathbf{y})$ converges to $-\sum_{a=1}^d x_a \log(y_a/x_a)$, which is the Kullback-Leibler divergence. For $\mu = 1$, $\Omega_1(\mathbf{x}, \mathbf{y}) = \|\mathbf{x} - \mathbf{y}\|_2^2$, which is the squared error loss. Hence $\mu = 0$ and $\mu = 1$ are the two special

cases of the density power divergence. Usually, in the context of likelihood estimation, the value of μ is varied in the interval $[0, 1]$ to achieve robustness by sacrificing the efficiency by some amount. The higher the value of μ , the outliers are more down-weighted, resulting in a more robust solution under data contamination. Here, in the convex clustering set-up, we can use this loss function to capture the divergence between $\hat{\pi}_j$ and \mathbf{b}_j , and in that case we set

$$\rho(\hat{\pi}_j, \mathbf{b}_j) = \frac{1}{1+\mu} \sum_{a=1}^d \left\{ b_{j,a}^{1+\mu} - \left(1 + \frac{1}{\mu}\right) b_{j,a}^{\mu} \hat{\pi}_{j,a} + \frac{1}{\mu} \hat{\pi}_{j,a}^{1+\mu} \right\}$$

for some $\mu > 0$. For $\mu = 0$, i.e. for KL divergence, the loss function looks likes

$$\rho(\hat{\pi}_j, \mathbf{b}_j) = - \sum_{a=1}^d \hat{\pi}_{j,a} \log(b_{j,a} / \hat{\pi}_{j,a}),$$

which is not strongly convex. Hence we rely on the ADMM algorithm developed earlier in the paper, and cannot apply the supposedly faster AMA algorithm. On the other hand, for $\mu = 1$, we can make the algorithm faster exploiting the strong convexity of ρ . The main question is for which values of μ , the loss will be strongly convex. Also, we want to investigate how much simplification one could make while updating the parameters sequentially, and whether there is any closed form solution for the updates in each step. The next theorem ensures that for any DPD measure with $\mu \in (0, 1]$, ρ is strongly convex.

Theorem 5.3.1 *Suppose $\mathbf{b}, \mathbf{y} \in \mathcal{S}_d$, where \mathcal{S}_d is the d -dimensional simplex. Define $g_{\mu}(\mathbf{b}) = \frac{1}{1+\mu} \Omega_{\mu}(\mathbf{y}, \mathbf{b})$. Then, g_{μ} is a strongly convex function with modulus μ .*

Proof: Recall that

$$g_{\mu}(\mathbf{b}) = \frac{1}{1+\mu} \sum_{a=1}^d \left\{ b_a^{1+\mu} - \left(1 + \frac{1}{\mu}\right) b_a^{\mu} y_a + \frac{1}{\mu} y_a^{1+\mu} \right\} = \sum_{a=1}^d h_a(b_a).$$

Now, $h'_a(b_a) = (b_a^\mu - b_a^{\mu-1} y_a)$. Hence, for $\mathbf{b}_1 \neq \mathbf{b}_2$,

$$\begin{aligned} h'_a(b_{1,a}) - h'_a(b_{2,a}) &= (\mu b_{*,a}^{\mu-1} + (1-\mu)b_{*,a}^{\mu-2} y_a)(b_{1,a} - b_{2,a}) \\ \implies (b_{1,a} - b_{2,a})(h'_a(b_{1,a}) - h'_a(b_{2,a})) &= (\mu b_{*,a}^{\mu-1} + (1-\mu)b_{*,a}^{\mu-2} y_a)(b_{1,a} - b_{2,a})^2 \\ &\geq \mu(b_{1,a} - b_{2,a})^2, \quad \text{since } b_{*,a} \in [0, 1]; \\ \implies (\mathbf{b}_1 - \mathbf{b}_2)^T (\nabla g_\mu(\mathbf{b}_1) - \nabla g_\mu(\mathbf{b}_2)) &\geq \mu \|\mathbf{b}_1 - \mathbf{b}_2\|_2^2 \end{aligned}$$

Hence, g_μ is strongly convex with modulus μ . Immediately following this result, we have the following corollary.

Corollary 5.3.1.1 *Let $\mathbf{B} = [\mathbf{b}_1, \dots, \mathbf{b}_p]$, and $f(\mathbf{B}) = \frac{1}{1+\mu} \sum_{j=1}^p \delta_j \sum_{a=1}^d \{b_{j,a}^{1+\mu} - (1 + \frac{1}{\mu})b_{j,a}^\mu \hat{\pi}_{j,a} + \frac{1}{\mu} \hat{\pi}_{j,a}^{1+\mu}\}$. Assume $d_j > 0$ for $j = 1, \dots, p$. Then $f(\mathbf{B})$ is strongly convex with modulus $(\min_j \delta_j) \mu$.*

The theorem has a very strong implication, that is we can apply the AMA algorithm suitably to speed up our clustering experiment for $\mu > 0$. Following this, it would be worthwhile to derive how the ADMM and AMA algorithm simplifies for $\mu = 0$ and $\mu > 0$ respectively.

5.3.1 ADMM for Kullback-Leibler Loss

Here we will derive the ADMM updates for the Kullback-Leibler divergence, especially the updates of \mathbf{b}_j while solving the equation (5.2.7). In this case

$$\nabla \rho(\hat{\pi}_j, \mathbf{b}_j) = (-\hat{\pi}_{j,1}/b_{j,1}, \dots, -\hat{\pi}_{j,d}/b_{j,d}), \quad \nabla^2 \rho(\hat{\pi}_j, \mathbf{b}_j) = \text{diag}(\hat{\pi}_{j,1}/b_{j,1}^2, \dots, \hat{\pi}_{j,d}/b_{j,d}^2).$$

Hence, we need to solve the equation

$$-\delta_j \frac{\hat{\pi}_{j,a}}{b_{j,a}} + \nu(p-1)b_{j,a} + \nu(\mathbf{1}^T \mathbf{b}_j) = \psi_{j,a}^{(t)}, \quad j = 1, \dots, p; a = 1, \dots, d$$

where $\psi_j^{(t)} = \sum_{l_1=j} \gamma_{l_1}^{(t)} - \sum_{l_2=j} \gamma_{l_2}^{(t)} + \nu \sum_{l_1=j} \mathbf{v}_{l_1}^{(t)} - \nu \sum_{l_2=j} \mathbf{v}_{l_2}^{(t)} + (\phi_j^{(t)} + \nu) \mathbf{1} + \nu \sum_{i \neq j} \mathbf{b}_i^{(t)}$. We can substitute $\mathbf{1}^T \mathbf{b}_j = 1$ in the above equation, since we want to enforce that condition anyway in our problem.

This will make our life a lot easier, since we have an exact solution of the above equation, which

turns out to be a quadratic equation as follows:

$$\begin{aligned} \nu(p-1)b_{j,a}^2 - (\psi_{j,a}^{(t)} - \nu)b_{j,a} - \delta_j \hat{\pi}_{j,a} &= 0 \\ \Rightarrow b_{j,a}^{(t+1)} &= \frac{(\psi_{j,a}^{(t)} - \nu) + \sqrt{(\psi_{j,a}^{(t)} - \nu)^2 + 4\nu(p-1)\delta_j \hat{\pi}_{j,a}}}{2\nu(p-1)}. \end{aligned}$$

Hence, we can directly update \mathbf{b}_j without the exact or one-step inexact Newton-Raphson method to solve the SMMGECCO problem. Consecutively, the other variables are updated following the algorithm (3).

5.3.2 AMA for DPD Loss

We first start with the easy case when $\mu = 1$. If all δ_j are equal, then this is exactly the same convex clustering problem developed by Chi and Lange (2015). Majumder et al. (2022) have shown that in that case, we don't have to even add the extra constraint $\mathbf{b}_j^T \mathbf{1} = 1$ for $j = 1, \dots, p$. Instead, the intermediate steps to execute the AMA themselves ensure that the optimum solution will be a valid probability vector. This property also holds when δ_j are different, which we will show immediately. Let us recall the augmented Lagrangian without the constraint $\mathbf{b}_j^T \mathbf{1} = 1$ for $\mu = 1$ will be as follows:

$$\begin{aligned} \mathcal{L}_\nu(\mathbf{B}, \mathbf{V}, \mathbf{\Gamma}) &= \frac{1}{2} \sum_{j=1}^p \delta_j \|\hat{\pi}_j - \mathbf{b}_j\|_2^2 + \lambda \sum_{l \in \mathcal{E}} w_l \|\mathbf{v}_l\|_2 \\ &+ \sum_{l \in \mathcal{E}} \langle \gamma_l, \mathbf{v}_l - \mathbf{b}_{l_1} + \mathbf{b}_{l_2} \rangle + \frac{\nu}{2} \sum_{l \in \mathcal{E}} \|\mathbf{v}_l - \mathbf{b}_{l_1} + \mathbf{b}_{l_2}\|_2^2. \end{aligned} \tag{5.3.9}$$

The related AMA algorithm for minimizing the above equation (5.3.9) is described in the algorithm (7).

We will now discuss the AMA updates explicitly for dpd loss function for $\mu > 0$. Recall that, we want to minimize the following objective function

$$\mathcal{L}(\mathbf{B}) = \sum_{j=1}^p \rho_\mu(\hat{\pi}_j, \mathbf{b}_j) + \lambda \sum_{1 \leq i < j \leq p} w_{i,j} \|\mathbf{b}_i - \mathbf{b}_j\|_2$$

for $\mathbf{b}_j \in \mathbb{S}^d$, the d -dimensional simplex. The corresponding augmented Lagrangian is given by

Algorithm 7 AMA for $\mu = 1$

Initialize $\Gamma^{(0)}$

```

1: for  $t = 1, 2, 3, \dots$  do
2:   for  $j = 1, 2, 3, \dots, p$  do
3:      $\Delta_j^{(t)} = \frac{1}{\delta_j} \left( \sum_{l_1=j} \gamma_l^{(t-1)} - \sum_{l_2=j} \gamma_l^{(t-1)} \right)$ 
4:   end for
5:   for all  $l$  do
6:      $\mathbf{g}_l^{(t)} = \hat{\pi}_{l_1} - \hat{\pi}_{l_2} + \Delta_{l_1}^{(t)} - \Delta_{l_2}^{(t)}$ 
7:      $\gamma_l^{(t)} = \mathcal{P}_{C_l}(\gamma_l^{(t-1)} - \nu \mathbf{g}_l^{(t)})$ 
8:   end for
9: end for

```

$$\begin{aligned}
\mathcal{L}_\nu(\mathbf{B}, \mathbf{V}, \Gamma, \Phi) &= \frac{1}{1+\mu} \sum_{j=1}^p \sum_{a=1}^d \left\{ b_{j,a}^{1+\mu} - \left(1 + \frac{1}{\mu}\right) b_{j,a}^\mu \hat{\pi}_{j,a} + \frac{1}{\mu} \hat{\pi}_{j,a}^{1+\mu} \right\} + \lambda \sum_{l \in \mathcal{E}} w_l \|\mathbf{v}_l\|_2 \\
&+ \sum_{l \in \mathcal{E}} \langle \gamma_l, \mathbf{v}_l - \mathbf{b}_{l_1} + \mathbf{b}_{l_2} \rangle + \sum_{j=1}^p \phi_j (1 - \mathbf{1}^T \mathbf{b}_j) \\
&+ \frac{\nu}{2} \sum_{l \in \mathcal{E}} \|\mathbf{v}_l - \mathbf{b}_{l_1} + \mathbf{b}_{l_2}\|_2^2 + \frac{\nu}{2} \sum_{j=1}^p (1 - \mathbf{1}^T \mathbf{b}_j)^2.
\end{aligned} \tag{5.3.10}$$

Let denote the updates of the parameters at the t^{th} iteration as $\mathbf{B}^{(t)}, \mathbf{V}^{(t)}, \mathbf{X}^{(t)}, \Gamma^{(t)}, \Phi^{(t)}$. In the next iteration, we first update $b_{j,a}$, keeping all other variables fixed. The AMA update of $b_{j,a}$ in the $(t+1)^{\text{th}}$ step, keeping $\nu = 0$ is the following:

$$\begin{aligned}
b_{j,a}^{(t+1)} &= \arg \min_{b_{j,a}} \left\{ \frac{1}{1+\mu} b_{j,a}^{1+\mu} - \frac{1}{\mu} b_{j,a}^\mu \hat{\pi}_{j,a} - \left(\sum_{l_1=j} \gamma_{l_1,a}^{(t)} - \sum_{l_2=j} \gamma_{l_2,a}^{(t)} + \phi_j^{(t)} \right) b_{j,a} \right\} \\
&= \arg \min_{b_{j,a}} \left\{ \frac{1}{1+\mu} b_{j,a}^{1+\mu} - \frac{1}{\mu} b_{j,a}^\mu \hat{\pi}_{j,a} - \psi_{j,a}^{(t)} b_{j,a} \right\}.
\end{aligned}$$

The major question here whether there exists a minimizer of the function $h_{j,a}(y) = \frac{1}{1+\mu} y^{1+\mu} - \frac{1}{\mu} y^\mu \hat{\pi}_{j,a} - \psi_{j,a} y$ for $y \in [0, \infty)$. The next lemma will guarantee that we indeed come up with an unique minimizer.

Lemma 5.3.2 *The function $h_{j,a}(y) = \frac{1}{1+\mu} y^{1+\mu} - \frac{1}{\mu} y^\mu \hat{\pi}_{j,a} - \psi_{j,a} y$ has an unique minimizer in the domain $[0, \infty)$.*

Proof: We prove our result for three different cases.

- (a) $\hat{\pi}_{j,a} = 0, \psi_{j,a} > 0$. In this case the minimizer is the solution of the equation $h'_{j,a}(y) = 0$, i.e. $y^\mu = \psi_{j,a}$. Hence the solution is $y^* = \psi_{j,a}^{1/\mu}$.
- (b) $\hat{\pi}_{j,a} = 0, \psi_{j,a} \leq 0$. In this case, $y^* = 0$ is the optimum solution.
- (c) $\hat{\pi}_{j,a} \neq 0$. In this case also, the minimizer is the solution of $h'_{j,a}(y) = y^\mu - y^{\mu-1}\hat{\pi}_{j,a} - \psi_{j,a} = 0$. Since $\mu \in (0, 1)$, $h'_{j,a}(0) = -\infty$, and $h'_{j,a}(M) > 0$ for sufficiently large M . Also, $h''_{j,a}(y) = \mu y^{\mu-1} + (1-\mu)y^{\mu-2}\hat{\pi}_{j,a} > 0$ for any $y > 0$. So, $h'_{j,a}(y)$ is strictly increasing in the domain $(0, \infty)$. Hence, by intermediate value theorem, $h'_{j,a}(y) = 0$ has an unique root in $(0, \infty)$.

However, there is no simple form of the minimizer, if we need to solve $h'_{j,a}(y) = 0$ for general μ . To obtain that solution, we need to perform Newton-Raphson algorithm in each iteration. It seems that the computational complexity will be heavily penalized, however in practice Newton-Raphson algorithm converges very fast in these scenarios where we have to find roots of the polynomials. Since there is an unique solution to the equation, the algorithm will be efficient enough to update the variable $b_{j,a}$. The next algorithm (8) will describe the Newton-Raphson for a single step update for $b_{j,a}$ by minimizing $h_{j,a}(y)$ over $[0, \infty)$.

Note that, in the Newton-Raphson step, we update $y_r = y_{r-1}/2$, if $y_{r-1} - \frac{h'_{j,a}(y_{r-1})}{h''_{j,a}(y_{r-1})} < 0$. This technique is adapted to relocate the starting point of the Newton-Raphson method. We know that the solution cannot be negative. So, in any intermediate step, the negative value of y_r will impact computing $h'_{j,a}(y_r)$ which will involve computing the fractional power of a negative number which interrupts the computation. We can also use the one-step update of N-R method in every step for executing the SMMGECCO+. In that case, given all other variables, the inexact update of $b_{j,a}$ at the $(t+1)^{th}$ iteration is the following:

$$b_{j,a}^{(t+1)} = b_{j,a}^{(t)} - \frac{(b_{j,a}^{(t)})^\mu - \hat{\pi}_{j,a}(b_{j,a}^{(t)})^{\mu-1} - \psi_{j,a}^{(t)}}{\mu(b_{j,a}^{(t)})^{\mu-1} + (1-\mu)\hat{\pi}_{j,a}(b_{j,a}^{(t)})^{\mu-2}}$$

Similarly, we update $b_{j,a}^{(t+1)} = b_{j,a}^{(t)}/2$ if the original update is negative. The updates of \mathbf{v}_l, γ_l and ϕ_j are unchanged. The exact and inexact AMA steps are summarized in the following algorithm (9).

Algorithm 8 Newton-Raphson for Updating $b_{j,a}$

Input: $\psi_{j,a}^{(t)}$, *tolerance*.

- 1: **if** $\hat{\pi}_{j,a} = 0$ **then**
- 2: $b_{j,a}^{(t+1)} = \left(\max\{0, \psi_{j,a}^{(t)}\} \right)^{1/\mu}$
- 3: **else**
- 4: $y_0 = \hat{\pi}_{j,0}$
- 5: **for** $r = 1, 2, \dots$ **do**
- 6: $y_r = y_{r-1} - \frac{h'_{j,a}(y_{r-1})}{h''_{j,a}(y_{r-1})}$
- 7: **if** $y_r < 0$ **then**
- 8: $y_r = y_{r-1}/2$
- 9: **end if**
- 10: $err = |y_r - y_{r-1}|$
- 11: **if** $err < tolerance$ **then**
- 12: $b_{j,a}^{(t+1)} = y_r$
- 13: STOP
- 14: **end if**
- 15: **end for**
- 16: **end if**

Algorithm 9 AMA Algorithm for DPD Loss

Initialize: $\mathbf{B}^{(0)}, \mathbf{V}^{(0)}, \mathbf{\Gamma}^{(0)}, \mathbf{\Phi}^{(0)}$, $t = 0$.

- 1: **while** Not Converged **do**
- 2: **if** SMMGECCO **then**
- 3: $\mathbf{b}_j^{(t+1)} =$ Full update using the Newton-Raphson in algorithm (8)
- 4: **end if**
- 5: **if** SMMGECCO+ **then**
- 6: **for** $a = 1, \dots, p$ **do**
- 7:
$$b_{j,a}^{(t+1)} = b_{j,a}^{(t)} - \frac{(b_{j,a}^{(t)})^\mu - \hat{\pi}_{j,a}(b_{j,a}^{(t)})^{\mu-1} - \psi_{j,a}^{(t)}}{\mu(b_{j,a}^{(t)})^{\mu-1} + (1-\mu)\hat{\pi}_{j,a}(b_{j,a}^{(t)})^{\mu-2}}$$
- 8: **if** $b_{j,a}^{(t+1)} < 0$ **then**
- 9: $b_{j,a}^{(t+1)} = b_{j,a}^{(t)}/2$
- 10: **end if**
- 11: **end for**
- 12: **end if**
- 13: $\gamma_l^{(t+1)} = \mathcal{P}_{C_l}(\gamma_l^{(t)} - \nu \mathbf{g}_l^{(t+1)});$
- 14: $\phi_j^{(t+1)} = \phi_j^{(t)} + \nu(1 - \mathbf{1}^T \mathbf{b}_j^{(t+1)});$
- 15: $t = t + 1$.
- 16: **end while**

5.3.3 Stopping Criteria

The convergence of the ADMM and AMA algorithms are discussed in Boyd et al. (2011) and Tseng (1991). The stopping criteria for these algorithms are computed by tracking the duality gap.

ADMM for Kullback-Leibler Loss

Following the similar computations by Boyd et al. (2011) and Chi and Lange (2015), the dual residual vectors after $(t+1)^{th}$ iteration are given by

$$\mathbf{s}_i^{(t+1)} = -\nu \left[\sum_{l_1=i} (\mathbf{v}_l^{(t+1)} - \mathbf{v}_l^{(t)}) - \sum_{l_2=i} (\mathbf{v}_l^{(t+1)} - \mathbf{v}_l^{(t)}) \right].$$

for $i = 1, \dots, p$. The primal residuals are defined by

$$\begin{aligned} \mathbf{r}_{1,l}^{(t+1)} &= \mathbf{b}_{l_1}^{(t+1)} - \mathbf{b}_{l_2}^{(t+1)} - \mathbf{v}_l^{(t+1)}, \quad l \in \mathcal{E} \\ \mathbf{r}_{2,i}^{(t+1)} &= (1 - \mathbf{1}^T \mathbf{b}_i^{(t+1)}), \quad i = 1, \dots, p. \end{aligned}$$

If \mathbf{r} be the vector of all the primal residuals and \mathbf{s} be the vector containing all the dual residuals, then we stop if $\|\mathbf{r}^{(t)}\|_2 \leq \epsilon_{\text{pri}}$ and $\|\mathbf{s}^{(t)}\|_2 \leq \epsilon_{\text{dual}}$ for some small numbers ϵ_{pri} and ϵ_{dual} . Using the proposal of Boyd et al. (2011), we find that the optimum stopping criteria would be

$$\begin{aligned} \epsilon_{\text{pri}} &= \sqrt{p} \epsilon_{\text{abs}} + \epsilon_{\text{rel}} \max \left\{ \sqrt{\sum_{i < j} \|\mathbf{b}_{l_1}^{(t)} - \mathbf{b}_{l_2}^{(t)}\|_2^2 + \sum_{i=1}^p (\mathbf{1}^T \mathbf{b}_i^{(t)})^2}, \|\mathbf{V}^{(t)}\|_F, \sqrt{p} \right\}; \\ \epsilon_{\text{dual}} &= \sqrt{d} \epsilon_{\text{abs}} + \epsilon_{\text{rel}} \sqrt{\sum_{i=1}^p \left\| \sum_{l_1=i} \gamma_i^{(t)} - \sum_{l_2=i} \gamma_i^{(t)} \right\|_2^2 + d \sum_{i=1}^p (\phi_j^{(t)})^2} \end{aligned}$$

where ϵ_{abs} and ϵ_{rel} control the absolute and relative precision. We leave the details of this calculation in the appendix.

AMA for DPD Loss

While solving the convex clustering for general DPD loss using AMA, we perform proximal gradient ascent to maximize the dual objective function. Note that the dual objective is defined by

$$\begin{aligned} D_\lambda(\mathbf{\Gamma}, \mathbf{\Phi}) &= \inf_{\mathbf{B}, \mathbf{V}} \mathcal{L}_0(\mathbf{B}, \mathbf{V}, \mathbf{\Gamma}, \mathbf{\Phi}) \\ &= \inf_{\mathbf{B}} \left\{ \sum_{j=1}^p \delta_j \rho(\hat{\pi}_j, \mathbf{b}_j) - \sum_{l \in \mathcal{E}} \langle \gamma_l, \mathbf{b}_{l_1} - \mathbf{b}_{l_2} \rangle + \sum_{j=1}^p \phi_j (1 - \mathbf{1}^T \mathbf{b}_j) \right\} \\ &\quad + \inf_{\mathbf{V}} \left\{ \lambda \sum_{l \in \mathcal{E}} w_l \|\mathbf{v}_l\| + \sum_{l \in \mathcal{E}} \langle \gamma_l, \mathbf{v}_l \rangle \right\} \end{aligned}$$

for $\rho(\hat{\pi}_j, \mathbf{b}_j) = \frac{1}{1+\mu} \sum_{a=1}^d \left\{ b_{j,a}^{1+\mu} - \left(1 + \frac{1}{\mu}\right) b_{j,a}^\mu \hat{\pi}_{j,a} + \frac{1}{\mu} \hat{\pi}_{j,a}^{1+\mu} \right\}$. Clearly, there is no simplified form of the above dual function for general μ for the first part of this dual. However if we observe carefully, for given values of $\mathbf{\Gamma}^{(t)}$ and $\mathbf{\Phi}^{(t)}$,

$$\arg \min_{\mathbf{B}} \left\{ \sum_{j=1}^p \delta_j \rho(\hat{\pi}_j, \mathbf{b}_j) - \sum_{l \in \mathcal{E}} \langle \gamma_l^{(t)}, \mathbf{b}_{l_1} - \mathbf{b}_{l_2} \rangle + \sum_{j=1}^p \phi_j^{(t)} (1 - \mathbf{1}^T \mathbf{b}_j) \right\} = \mathbf{B}^{(t+1)}$$

where $\mathbf{B}^{(t+1)}$ is the update of the variables \mathbf{B} at the $(t+1)^{th}$ iteration following the algorithm (9). Hence, we replace the value of $\mathbf{B}^{(t+1)}$ in the argument of the above function to obtain the value of the first part of the dual objective function. Finding the second part is easier as noted by Chi and Lange (2015) as

$$\inf_{\mathbf{V}} \left\{ \lambda \sum_{l \in \mathcal{E}} w_l \|\mathbf{v}_l\| + \sum_{l \in \mathcal{E}} \langle \gamma_l, \mathbf{v}_l \rangle \right\} = - \sum_{l \in \mathcal{E}} \mathcal{H}_{C_l}(\gamma_l)$$

where $C_l = \{\gamma_l : \|\gamma_l\|_2 \leq \lambda w_l\}$ and

$$\mathcal{H}_{C_l}(\mathbf{x}) = \begin{cases} 0 & \text{if } \mathbf{x} \in C_l, \\ \infty & \text{o.w.} \end{cases}$$

The proximal gradient ascent of $D_\lambda(\mathbf{\Gamma}, \mathbf{\Phi})$ with step size ν will produce the exact similar steps as outlined in the AMA algorithm (9).

For stopping criteria, we compute the duality gap the difference between the primal value and

the dual objective function, quantified by $F_\lambda(\mathbf{B}^{(t)}) - D_\lambda(\mathbf{\Gamma}^{(t)}, \mathbf{\Phi}^{(t)})$. We stop when

$$F_\lambda(\mathbf{B}^{(t)}) - D_\lambda(\mathbf{\Gamma}^{(t)}, \mathbf{\Phi}^{(t)}) < \tau$$

for some pre-specified small number τ . We know that, under the optimality the duality gap should be zero. Hence, this quantity can serve as a reliable measure for convergence.

5.4 Perfect Recovery Conditions for General Loss

In this section, we will provide the range of λ values, for which the true clusters are identified under certain conditions. Note that, for the usual convex-clustering problem using the squared error loss, which is equivalent to solving the equation (5.2.5) with $\delta_j = 1$, the range of λ values for perfect recovery have been first developed by Zhu et al. (2014). Their works were limited to the case with $w_{i,j} = 1$ and with only two clusters. Panahi et al. (2017) later extended that result for general k clusters. Sun et al. (2021) provides more general limits for any general weights $w_{i,j}$, with some extra assumptions. In this paper, we will show how these cluster recovery conditions change with the change of the loss-function, especially with the dpd loss discussed earlier. We will provide three different results, one for $\mu = 0$, i.e. the Kullback-Leibler divergence; $\mu = 1$, the squared error loss with different δ_j values and for general $0 < \mu < 1$.

Before going into the main results, we define some notations and provide two assumptions, exactly same as of Sun et al. (2021) and Majumder et al. (2022), which will play a key role in proving our result. Let the order of the Markov chain is m , and the true partitions of the state space Σ^m be

$\mathcal{C}_1, \dots, \mathcal{C}_{k_0}$. Define

$$\begin{aligned}
w_i^{(\beta)} &= \sum_{j \in \mathcal{C}_\beta} w_{i,j} \quad \forall i = 1, 2, \dots, p; & \eta_{i,j}^{(\alpha)} &= \sum_{\ell \neq \alpha} |\delta_j w_i^{(\ell)} - \delta_i w_j^{(\ell)}| \quad \forall \alpha = 1, 2, \dots, k_0; \\
w^{(\alpha, \beta)} &= \sum_{i \in \mathcal{C}_\alpha} \sum_{j \in \mathcal{C}_\beta} w_{i,j} \quad \forall \alpha \neq \beta, \alpha, \beta \in \{1, 2, \dots, k_0\}; & \hat{\boldsymbol{\pi}}^{(\alpha)} &= \frac{1}{\delta^{(\alpha)}} \sum_{i \in \mathcal{C}_\alpha} \delta_i \hat{\boldsymbol{\pi}}_i; & \delta^{(\alpha)} &= \sum_{i \in \mathcal{C}_\alpha} \delta_i \\
s_{i,j}^{(\alpha)} &= \delta_i \delta_j \left\| \frac{\hat{\boldsymbol{\pi}}_i - \hat{\boldsymbol{\pi}}_j}{\hat{\boldsymbol{\pi}}^{(\alpha)}} \right\|_2; & \text{where, for 2 vectors } \mathbf{x}, \mathbf{y} \in \mathbb{R}^d, & \frac{\mathbf{x}}{\mathbf{y}} &= \left(\frac{x_1}{y_1} \mathcal{I}(y_1 \neq 0), \dots, \frac{x_d}{y_d} \mathcal{I}(y_d \neq 0) \right); \\
\lambda_{\min}^{(n)} &= \max_{1 \leq \alpha \leq k_0} \max_{i,j \in \mathcal{C}_\alpha} \left\{ \frac{s_{i,j}^{(\alpha)}}{\delta^{(\alpha)} w_{i,j} - \eta_{i,j}^{(\alpha)} - 2s_{i,j}^{(\alpha)} \frac{\sum_{l \neq \alpha} w^{(\alpha,l)}}{\delta^{(\alpha)}}} \right\}; \\
\lambda_{\max}^{(n)} &= \min_{1 \leq \alpha < \beta \leq k_0} \left\{ \frac{\frac{1}{2} \|\hat{\boldsymbol{\pi}}^{(\alpha)} - \hat{\boldsymbol{\pi}}^{(\beta)}\|_2}{\frac{1}{\delta^{(\alpha)}} \sum_{l \neq \alpha} w^{(\alpha,l)} + \frac{1}{\delta^{(\beta)}} \sum_{l \neq \beta} w^{(\beta,l)}}} \right\}.
\end{aligned}$$

We suppose that the following conditions hold.

(A1) $w_{i,j} = w_{j,i}$ and $w_{i,j} > 0$ for any $i, j \in \mathcal{C}_\ell, \ell = 1, 2, \dots, k_0$.

(A2) $\delta^{(\alpha)} w_{i,j} > \eta_{i,j}^{(\alpha)} + 2s_{i,j}^{(\alpha)} \frac{\sum_{l \neq \alpha} w^{(\alpha,l)}}{\delta^{(\alpha)}}$, $\forall i, j \in \mathcal{C}_\alpha$ and $\forall \alpha = 1, 2, \dots, k_0$.

Next, we present our main result for cluster recovery.

Theorem 5.4.1 *Consider solving the equation (5.2.3), with ρ being the density power divergence loss function for some $0 \leq \mu < 1$; and let the above two conditions (A1) and (A2) hold. Then, for any $\lambda \in (\lambda_{\min}^{(n)}, \lambda_{\max}^{(n)})$, the clusters are recovered perfectly.*

Proof:

Case 1: First we prove it for $\mu = 0$, i.e. for the Kullback-Leibler divergence. Here, we want to minimize the following objective function

$$-\sum_{j=1}^p \delta_j \sum_{a=1}^d \hat{\boldsymbol{\pi}}_{j,a} \log(b_{j,a}) + \lambda \sum_{i < j} w_{i,j} \|\mathbf{b}_i - \mathbf{b}_j\|_2 + \sum_{j=1}^p \phi_j (1 - \mathbf{1}^T \mathbf{b}_j). \quad (5.4.11)$$

The corresponding centroid optimization problem is given by

$$\begin{aligned} \min_{\mathbf{X}^{(1)}, \dots, \mathbf{X}^{(k_0)}; \in \mathbb{S}^d} \left\{ - \sum_{\alpha=1}^{k_0} \delta^{(\alpha)} \hat{\pi}_a^{(\alpha)} \log(X_a^{(\alpha)}) + \lambda \sum_{\alpha=1}^{k_0-1} \sum_{\beta=\alpha+1}^{k_0} w^{(\alpha, \beta)} \|\mathbf{X}^{(\alpha)} - \mathbf{X}^{(\beta)}\|_2 \right. \\ \left. + \sum_{\alpha=1}^{k_0} \phi^{(\alpha)} (1 - \mathbf{1}^T \mathbf{X}^{(\alpha)}) \right\} \end{aligned} \quad (5.4.12)$$

By taking sub-differential of the above equation, we get

$$- \frac{\delta^{(\alpha)} \hat{\pi}_a^{(\alpha)}}{X_a^{(\alpha)}} + \lambda \sum_{\beta \neq \alpha} w^{(\alpha, \beta)} z_a^{(\alpha, \beta)} - \phi^{(\alpha)} = 0 \quad (5.4.13)$$

where $\mathbf{z}^{(\alpha, \beta)} \in \partial h(\mathbf{X}^{(\alpha)} - \mathbf{X}^{(\beta)})$, $h(x) = \|x\|_2$, and $\mathbf{z}^{(\alpha, \beta)} = -\mathbf{z}^{(\beta, \alpha)}$ for $\alpha \neq \beta$. From equation (5.4.13), we get

$$\begin{aligned} & - \delta^{(\alpha)} \hat{\pi}_a^{(\alpha)} + \lambda \sum_{\beta \neq \alpha} w^{(\alpha, \beta)} z_a^{(\alpha, \beta)} X_a^{(\alpha)} - \phi^{(\alpha)} X_a^{(\alpha)} = 0 \\ \implies & - \delta^{(\alpha)} \sum_{a=1}^d \hat{\pi}_a^{(\alpha)} + \lambda \sum_{\beta \neq \alpha} w^{(\alpha, \beta)} \sum_{a=1}^d z_a^{(\alpha, \beta)} X_a^{(\alpha)} = \phi^{(\alpha)} \sum_{a=1}^d X_a^{(\alpha)} \\ \implies & \phi^{(\alpha)} + \delta^{(\alpha)} = \lambda \sum_{\beta \neq \alpha} w^{(\alpha, \beta)} \sum_{a=1}^d z_a^{(\alpha, \beta)} X_a^{(\alpha)} \implies |\phi^{(\alpha)} + \delta^{(\alpha)}| \leq \lambda \sum_{\beta \neq \alpha} w^{(\alpha, \beta)} \left| \sum_{a=1}^d z_a^{(\alpha, \beta)} X_a^{(\alpha)} \right| \\ \implies & |\phi^{(\alpha)} + \delta^{(\alpha)}| \leq \lambda \sum_{\beta \neq \alpha} w^{(\alpha, \beta)} \|\mathbf{z}^{(\alpha, \beta)}\|_2 \|\mathbf{X}^{(\alpha)}\|_2 \leq \lambda \sum_{\beta \neq \alpha} w^{(\alpha, \beta)}. \end{aligned}$$

Hence, after some rearrangements, for all $\alpha = 1, \dots, k_0$, we get

$$\begin{aligned} (\phi^{(\alpha)} + \delta^{(\alpha)}) \mathbf{X}^{(\alpha)} &= \delta^{(\alpha)} (\mathbf{X}^{(\alpha)} - \hat{\pi}^{(\alpha)}) + \lambda \sum_{\alpha \neq \ell} w^{(\alpha, \ell)} \mathbf{z}^{(\alpha, \ell)} * \mathbf{X}^{(\alpha)} \\ (\mathbf{X}^{(\alpha)} - \mathbf{X}^{(\beta)}) &= (\hat{\pi}^{(\alpha)} - \hat{\pi}^{(\beta)}) + \left(\frac{\phi^{(\alpha)} + \delta^{(\alpha)}}{\delta^{(\alpha)}} \right) \mathbf{X}^{(\alpha)} - \left(\frac{\phi^{(\beta)} + \delta^{(\beta)}}{\delta^{(\beta)}} \right) \mathbf{X}^{(\beta)} \\ &+ \lambda \left(\frac{\sum_{\alpha \neq \ell} w^{(\alpha, \ell)} \mathbf{z}^{(\alpha, \ell)} * \mathbf{X}^{(\alpha)}}{\delta^{(\alpha)}} - \frac{\sum_{\beta \neq \ell} w^{(\beta, \ell)} \mathbf{z}^{(\beta, \ell)} * \mathbf{X}^{(\beta)}}{\delta^{(\beta)}} \right) \end{aligned}$$

$$\begin{aligned}
\Rightarrow \|\mathbf{X}^{(\alpha)} - \mathbf{X}^{(\beta)}\|_2 &\geq \|\hat{\boldsymbol{\pi}}^{(\alpha)} - \hat{\boldsymbol{\pi}}^{(\beta)}\|_2 - \left(\frac{\phi^{(\alpha)} + \delta^{(\alpha)}}{\delta^{(\alpha)}} \right) - \left(\frac{\phi^{(\beta)} + \delta^{(\beta)}}{\delta^{(\beta)}} \right) \\
&\quad - \lambda \left(\frac{\sum_{\alpha \neq \ell} w^{(\alpha, \ell)}}{\delta^{(\alpha)}} + \frac{\sum_{\beta \neq \ell} w^{(\beta, \ell)}}{\delta^{(\beta)}} \right) \\
&\geq \|\hat{\boldsymbol{\pi}}^{(\alpha)} - \hat{\boldsymbol{\pi}}^{(\beta)}\|_2 - 2\lambda \left(\frac{\sum_{\alpha \neq \ell} w^{(\alpha, \ell)}}{\delta^{(\alpha)}} + \frac{\sum_{\beta \neq \ell} w^{(\beta, \ell)}}{\delta^{(\beta)}} \right) > 0
\end{aligned}$$

if $\lambda < \lambda_{\max}^{(n)}$. This shows for $\lambda < \lambda_{\max}^{(n)}$, the solution $\mathbf{X}^{(\alpha)}$ of the centroid optimization problem (5.4.12) are distinct.

In the next part, we will show if $\lambda > \lambda_{\min}^{(n)}$, the solutions of the original clustering problem (5.4.11) and the centroid optimization problem (5.4.12) coincide with each other. From the sub-differential conditions, the solution $\mathbf{b}_1^*, \dots, \mathbf{b}_p^*$ of (5.4.11) satisfies the equation

$$\begin{aligned}
&-\delta_i \frac{\hat{\boldsymbol{\pi}}_i}{\mathbf{b}_i^*} + \lambda \sum_{j \neq i} w_{i,j} \mathbf{z}_{i,j} - \phi_i \mathbf{1} = \mathbf{0} \\
\Rightarrow &-\delta_i \hat{\boldsymbol{\pi}}_i + \lambda \sum_{j \neq i} w_{i,j} \mathbf{z}_{i,j} * \mathbf{b}_i^* - \phi_i \mathbf{b}_i^* = \mathbf{0}
\end{aligned} \tag{5.4.14}$$

for $\mathbf{z}_{i,j} \in \partial h(\mathbf{b}_i^* - \mathbf{b}_j^*)$. Set,

$$\mathbf{z}_{i,j} = \begin{cases} \mathbf{z}^{(\alpha, \beta)} & \text{if } i \in \mathcal{C}_\alpha, j \in \mathcal{C}_\beta, \alpha \neq \beta \\ \frac{1}{\delta^{(\alpha)} w_{i,j}} \left[\frac{\delta_i \delta_j (\hat{\boldsymbol{\pi}}_i - \hat{\boldsymbol{\pi}}_j)}{\lambda \mathbf{X}^{(\alpha)}} - \sum_{\beta \neq \alpha} (\delta_j w_i^{(\beta)} - \delta_i w_j^{(\beta)}) \mathbf{z}^{(\alpha, \beta)} \right] & \text{if } i, j \in \mathcal{C}_\alpha. \end{cases}$$

Then, substituting $\mathbf{X}^{(\alpha)}$ in (5.4.14) for \mathbf{b}_i^* when $i \in \mathcal{C}_\alpha$, we get

$$\begin{aligned}
& -\delta_i \hat{\pi}_i + \lambda \sum_{\beta \neq \alpha} w_i^{(\beta)} \mathbf{z}^{(\alpha, \beta)} * \mathbf{X}^{(\alpha)} + \sum_{j \in \mathcal{C}_\alpha} \frac{1}{\delta^{(\alpha)}} [\delta_i \delta_j (\hat{\pi}_i - \hat{\pi}_j)] \\
& - \frac{\lambda}{\delta^{(\alpha)}} \sum_{j \in \mathcal{C}_\alpha} \sum_{\beta \neq \alpha} (\delta_j w_i^{(\beta)} - \delta_i w_j^{(\beta)}) \mathbf{z}^{(\alpha, \beta)} * \mathbf{X}^{(\alpha)} - \phi_i \mathbf{X}^{(\alpha)} \\
& = -\delta_i \hat{\pi}_i + \lambda \sum_{\beta \neq \alpha} w_i^{(\beta)} \mathbf{z}^{(\alpha, \beta)} * \mathbf{X}^{(\alpha)} + \delta_i \hat{\pi}_i - \delta_i \hat{\pi}^{(\alpha)} - \lambda \sum_{\beta \neq \alpha} w_i^{(\beta)} \mathbf{z}^{(\alpha, \beta)} * \mathbf{X}^{(\alpha)} \\
& + \frac{\lambda \delta_i}{\delta^{(\alpha)}} \sum_{\beta \neq \alpha} w^{(\alpha, \beta)} \mathbf{z}^{(\alpha, \beta)} * \mathbf{X}^{(\alpha)} - \phi_i \mathbf{X}^{(\alpha)} \\
& = \frac{\delta_i}{\delta^{(\alpha)}} \left(\sum_{\beta \neq \alpha} w^{(\alpha, \beta)} \mathbf{z}^{(\alpha, \beta)} * \mathbf{X}^{(\alpha)} - \delta^{(\alpha)} \hat{\pi}^{(\alpha)} \right) - \phi_i \mathbf{X}^{(\alpha)} = \frac{\phi^{(\alpha)} \delta_i}{\delta^{(\alpha)}} \mathbf{X}^{(\alpha)} - \phi_i \mathbf{X}^{(\alpha)} = 0
\end{aligned}$$

where $\phi_i = \frac{\phi^{(\alpha)} \delta_i}{\delta^{(\alpha)}}$. Note that $\mathbf{z}_{i,j}$ has the property that $\mathbf{z}_{i,j} = -\mathbf{z}_{j,i}$. It only remains to show that $\mathbf{z}_{i,j} \in \partial h(\mathbf{b}_i^* - \mathbf{b}_j^*)$. For $i \in \mathcal{C}_\alpha$ and $j \in \mathcal{C}_\beta$ it is trivial from the definition. So, we have to show $\mathbf{z}_{i,j} \in \partial h(0)$ when $i, j \in \mathcal{C}_\alpha$. From (5.4.13) and the subsequent bound of $|\phi^{(\alpha) + \delta^{(\alpha)}}|$, we get

$$\frac{1}{X^{(\alpha)}} = \frac{\lambda \sum_{\beta \neq \alpha} w^{(\alpha, \beta)} z_a^{(\alpha, \beta)} - \phi^{(\alpha)}}{\delta^{(\alpha)} \hat{\pi}_a^{(\alpha)}} \leq \frac{2\lambda \sum_{\beta \neq \alpha} w^{(\alpha, \beta)} + \delta^{(\alpha)}}{\delta^{(\alpha)} \hat{\pi}_a^{(\alpha)}}$$

In that case, if $\lambda > \lambda_{\min}^{(n)}$,

$$\begin{aligned}
\|\mathbf{z}_{i,j}\|_2 & \leq \frac{1}{\delta^{(\alpha)} w_{i,j}} \left[\left(\frac{2\lambda \sum_{\beta \neq \alpha} w^{(\alpha, \beta)} + \delta^{(\alpha)}}{\lambda \delta^{(\alpha)}} \right) \left(\delta_i \delta_j \left\| \frac{\hat{\pi}_i - \hat{\pi}_j}{\hat{\pi}^{(\alpha)}} \right\|_2 \right) + \sum_{\beta \neq \alpha} \left| \delta_j w_i^{(\beta)} - \delta_i w_j^{(\beta)} \right| \right] \\
& = \frac{1}{\delta^{(\alpha)} w_{i,j}} \left[\frac{s_{i,j}^{(\alpha)}}{\lambda} + \frac{2 \sum_{\beta \neq \alpha} w^{(\alpha, \beta)}}{\delta^{(\alpha)}} s_{i,j}^{(\alpha)} + \eta_{i,j}^{(\alpha)} \right] \leq 1.
\end{aligned}$$

This completes the proof for perfect recovery condition for Kullback-Leibler divergence.

Case 2: Here we will prove the result for $0 < \mu < 1$. The major steps will be similar as the case when $\mu = 0$, with some notational differences. Note that, the centroid optimization problem

will require us to solve the following equation

$$\delta^{(\alpha)}(\mathbf{X}^{(\alpha)} - \hat{\boldsymbol{\pi}}^{(\alpha)}) + \lambda \sum_{\beta \neq \alpha} w^{(\alpha, \beta)} \mathbf{z}^{(\alpha, \beta)} * (\mathbf{X}^{(\alpha)})^{1-\mu} - \phi^{(\alpha)} (\mathbf{X}^{(\alpha)})^{1-\mu} = \mathbf{0}; \quad (5.4.15)$$

which after taking the sums of the elements of the vectors on the both side turns out to be

$$\begin{aligned} & \lambda \sum_{\beta \neq \alpha} w^{(\alpha, \beta)} \sum_{a=1}^d z_a^{(\alpha, \beta)} (X_a^{(\alpha)})^{1-\mu} - \phi^{(\alpha)} \sum_{a=1}^d (X_a^{(\alpha)})^{1-\mu} = 0 \\ \implies \phi^{(\alpha)} &= \frac{\lambda \sum_{\beta \neq \alpha} w^{(\alpha, \beta)} \sum_{a=1}^d z_a^{(\alpha, \beta)} (X_a^{(\alpha)})^{1-\mu}}{\sum_{a=1}^d (X_a^{(\alpha)})^{1-\mu}} \\ \implies \phi^{(\alpha)} &\leq \lambda \sum_{\beta \neq \alpha} w^{(\alpha, \beta)} \|\mathbf{z}^{(\alpha, \beta)}\|_2 \left\| \frac{(\mathbf{X}^{(\alpha)})^{1-\mu}}{\mathbf{1}^T (\mathbf{X}^{(\alpha)})^{1-\mu}} \right\|_2 \leq \lambda \sum_{\beta \neq \alpha} w^{(\alpha, \beta)}. \end{aligned}$$

From (5.4.15), we can derive similar relation as follows as we did in case 1:

$$\|\mathbf{X}^{(\alpha)} - \mathbf{X}^{(\beta)}\|_2 \geq \|\hat{\boldsymbol{\pi}}^{(\alpha)} - \hat{\boldsymbol{\pi}}^{(\beta)}\|_2 - 2\lambda \left(\frac{\sum_{\alpha \neq \ell} w^{(\alpha, \ell)}}{\delta^{(\alpha)}} + \frac{\sum_{\beta \neq \ell} w^{(\beta, \ell)}}{\delta^{(\beta)}} \right) > 0$$

when $\lambda < \lambda_{\max}^{(n)}$. Also from (5.4.15),

$$\frac{1}{(X_a^{(\alpha)})^{1-\mu}} = \frac{1}{\delta^{(\alpha)} \hat{\pi}_a^{(\alpha)}} \left(\delta^{(\alpha)} (X_a^{(\alpha)})^\mu + \lambda \sum_{\beta \neq \alpha} w^{(\alpha, \beta)} z_a^{(\alpha, \beta)} - \phi^{(\alpha)} \right) \leq \frac{1}{\delta^{(\alpha)} \hat{\pi}_a^{(\alpha)}} \left(\delta^{(\alpha)} + 2\lambda \sum_{\beta \neq \alpha} w^{(\alpha, \beta)} \right).$$

By setting

$$\mathbf{z}_{i,j} = \begin{cases} \mathbf{z}^{(\alpha, \beta)} & \text{if } i \in \mathcal{C}_\alpha, j \in \mathcal{C}_\beta, \alpha \neq \beta \\ \frac{1}{\delta^{(\alpha)} w_{i,j}} \left[\frac{\delta_i \delta_j (\hat{\pi}_i - \hat{\pi}_j)}{\lambda (\mathbf{X}^{(\alpha)})^{1-\mu}} - \sum_{\beta \neq \alpha} (\delta_j w_i^{(\beta)} - \delta_i w_j^{(\beta)}) \mathbf{z}^{(\alpha, \beta)} \right] & \text{if } i, j \in \mathcal{C}_\alpha. \end{cases}$$

it is exactly similar to show that for $\lambda > \lambda_{\min}^{(n)}$ $\|\mathbf{z}_{i,j}\|_2 \leq 1$ if $i, j \in \mathcal{C}_\alpha$ using the above bound for $\frac{1}{(X_a^{(\alpha)})^{1-\mu}}$. Thus we prove the cluster recovery conditions for general dpd loss.

The next theorem will provide the cluster recovery condition for $\mu = 1$. The proof will follow the similar techniques as theorem 5.4.1, hence omitted for brevity.

Theorem 5.4.2 Consider solving the equation (5.2.3), with $\rho(\hat{\boldsymbol{\pi}}_j, \mathbf{b}_j) = \frac{1}{2} \|\hat{\boldsymbol{\pi}}_j - \mathbf{b}_j\|_2^2$. Define

$$\lambda'_{\min} = \max_{1 \leq \alpha \leq k_0} \max_{i, j \in \mathcal{C}_\alpha} \left\{ \frac{\delta_i \delta_j \|\hat{\boldsymbol{\pi}}_i - \hat{\boldsymbol{\pi}}_j\|_2}{\delta^{(\alpha)} w_{i,j} - \eta_{i,j}^{(\alpha)}} \right\};$$

$$\lambda'_{\max} = \min_{1 \leq \alpha < \beta \leq k_0} \left\{ \frac{\|\hat{\boldsymbol{\pi}}^{(\alpha)} - \hat{\boldsymbol{\pi}}^{(\beta)}\|_2}{\frac{1}{\delta^{(\alpha)}} \sum_{l \neq \alpha} w^{(\alpha, l)} + \frac{1}{\delta^{(\beta)}} \sum_{l \neq \beta} w^{(\beta, l)}} \right\}.$$

Suppose, the following conditions hold:

(A1) $w_{i,j} = w_{j,i}$ and $w_{i,j} > 0$ for any $i, j \in \mathcal{C}_\ell, \ell = 1, 2, \dots, k_0$.

(A2)' $\delta^{(\alpha)} w_{i,j} > \eta_{i,j}^{(\alpha)}, \forall i, j \in \mathcal{C}_\alpha$ and $\forall \alpha = 1, 2, \dots, k_0$.

Then, for any $\lambda \in (\lambda'_{\min}, \lambda'_{\max})$, the clusters are recovered perfectly.

Similar cluster recovery conditions were developed by Sun et al. (2021), with more simplified conditions. The extra complication arise from the fact that for general dpd loss with $\mu \neq 1$, the computation become cumbersome, and we don't always have a closed form update of the related ADMM or AMA algorithm. An added layer of complexity comes from the incorporation of the constraints $\mathbf{b}_j^T \mathbf{1} = 1$, for $j = 1, \dots, p$. Note that, the expressions of $\lambda_{\min}^{(n)}$ and $\lambda_{\max}^{(n)}$ involves the linear functions of the random variables $\hat{\boldsymbol{\pi}}_j$. Hence, the phenomenon $\lambda_{\min}^{(n)} < \lambda_{\max}^{(n)}$ is a random event in this case. Using the perfect recovery conditions of Sun et al. (2021), Majumder et al. (2022) have derived the probability of $\lambda_{\min}^{(n)}$ being less than $\lambda_{\max}^{(n)}$, which converges to 1 as $n \rightarrow \infty$. That says, when we look at the solutions of this regularized method over a range of λ , the true model is present in the solution path. If not, then there is no chance of recovering the true model under any selection criterion. Majumder et al. (2022) have used the BIC criteria to select the optimum tuning parameter λ , along with the model selection consistency of the BIC in this context. We will use this BIC criterion for model selection purpose in the simulation study.

Another important point of discussion can be how these conditions are relevant in the cluster recovery problem, and whether there is any intuition behind the algebraic terms $\lambda_{\min}^{(n)}$ and $\lambda_{\max}^{(n)}$. If we observe carefully, $\lambda_{\min}^{(n)}$ measures the maximum difference of the scaled transition probability vectors in a particular cluster, i.e. a measure of intra-cluster separation. On the other hand, $\lambda_{\max}^{(n)}$

measures the inter-cluster separation of the estimated cluster centroids. In the long run, we expect the estimated transition probabilities to converge to the true cluster mean, hence $\hat{\pi}_i$ and $\hat{\pi}_j$ are very close to each other. This results in a very small value of $\lambda_{\min}^{(n)}$. On the other hand, for different clusters \mathcal{C}_α and \mathcal{C}_β , $\hat{\pi}^{(\alpha)}$ and $\hat{\pi}^{(\beta)}$ are far from each other, making $\lambda_{\max}^{(n)}$ far away from 0, hence greater than $\lambda_{\min}^{(n)}$ with higher probability in the long run.

If the weights are chosen properly, we expect that $w_{i,j}$ will be higher for $i, j \in \mathcal{C}_\alpha$, while the weights should be less when σ_i and σ_j belong to two different clusters. In practice, a common choice of the weight is $w_{i,j} = \exp[-g(\|\hat{\pi}_i - \hat{\pi}_j\|)] \mathcal{I}(\hat{\pi}_i \text{ is one of the } k\text{-nearest neighbor of } \hat{\pi}_j)$ or vice versa, where g is a monotone function of the norm $\|\cdot\|$. Chi and Lange (2015), Sun et al. (2021) and Majumder et al. (2022) have suggested $g(\mathbf{x}) = \|\mathbf{x}\|_2^2$, $\|\mathbf{x}\|_1$ or $\|\mathbf{x}\|_\infty$. In such cases, the terms $\eta_{i,j}^{(\alpha)}$ and $s_{i,j}^{(\alpha)}$ are close to 0 when $\hat{\pi}_i$ and $\hat{\pi}_j$ are close to their true cluster mean. Hence, the condition (A2) is satisfied, since $w_{i,j}$ will be away from 0. Alongside, (A1) is also satisfied for properly chosen nearest neighbour.

Compared to the traditional convex clustering, we have improvised the smooth part of the objective functions by incorporating the weights δ_j with the element-wise loss function $\rho(\hat{\pi}_j, \mathbf{b}_j)$. A reasonable choice of δ_j is N_{σ_j}/N , the sample proportion of σ_j , the j^{th} history in Σ^m . There are two advantages of using this weight. Firstly, if the frequency of a history is comparatively less in the sequence, the amount of penalization will be small too. Secondly, the estimates of the true cluster centers $\hat{\pi}^{(\alpha)}$ will be the weighted means of $\hat{\pi}_i, i \in \mathcal{C}_\alpha$. So even for less sample size n , these estimates will be closer to the true cluster center as compared to $\frac{1}{p_\alpha} \sum_{i \in \mathcal{C}_\alpha} \hat{\pi}_i$, possibly helping us separating the cluster centers more accurately.

5.5 Sparse Relaxed Regularized Convex Clustering (SR2C2) Algorithm

So far, we have considered solving the original objective function (5.2.4) using constrained optimization techniques ADMM or AMA. In this section, we propose a new algorithm for regularized SMM fit for squared error loss. A unified framework for ‘‘Sparse Relaxed Regularized Regression’’ (SR3) was proposed by Zheng et al. (2018) for solving penalized regression problems; including, but not limited to LASSO, SCAD, compressed sensing and matrix completion. The usual regularized

linear regression problem is given by

$$\min_{\beta} \frac{1}{2} \|\mathbf{y} - \mathbf{X}\beta\|_2^2 + \lambda R(\mathbf{C}\beta) \quad (5.5.16)$$

where $\mathbf{y} \in \mathbb{R}^n$ is the observation vector, \mathbf{X} is a $n \times p$ design matrix, β is the p -dimensional parameter, \mathbf{C} is some $q \times p$ matrix, R is some penalty function and λ is a regularization parameter. In many scenarios, for example in LASSO, SCAD or compressed sensing problems, R is not differentiable. Hence some kind of relaxation in the objective function is needed for solving such problems. Zheng et al. (2018) have proposed to solve the following formulation

$$\min_{\mathbf{w}, \beta} \frac{1}{2} \|\mathbf{y} - \mathbf{X}\beta\|_2^2 + \lambda R(\mathbf{w}) + \frac{\nu}{2} \|\mathbf{w} - \mathbf{C}\beta\|_2^2 \quad (5.5.17)$$

for some tuning parameter $\nu > 0$, which controls the amount of penalization for \mathbf{w} being different from $\mathbf{C}\beta$. Note that, we can first minimize the objective function in (5.5.17) w.r.t. β , keeping \mathbf{w} , which is nothing but solving a quadratic equation. We denote this solution as $\beta(\mathbf{w})$, and

$$\beta(\mathbf{w}) = \mathbf{H}_\nu^{-1} (\mathbf{X}^T \mathbf{y} + \nu \mathbf{C}^T \mathbf{w}); \quad \mathbf{H}_\nu = \mathbf{X}^T \mathbf{X} + \nu \mathbf{C}^T \mathbf{C}.$$

Substituting this value in the equation (5.5.17), we need to solve the following problem

$$\min_{\mathbf{w}} \frac{1}{2} \|\mathbf{F}_\nu \mathbf{w} - \mathbf{g}_\nu\|_2^2 + \lambda R(\mathbf{w}); \quad (5.5.18)$$

where

$$\mathbf{F}_\nu = \begin{bmatrix} \nu \mathbf{X} \mathbf{H}_\nu^{-1} \mathbf{C}^T \\ \sqrt{\nu} (\mathbf{I} - \nu \mathbf{C} \mathbf{H}_\nu^{-1} \mathbf{C}^T) \end{bmatrix}; \quad \mathbf{G}_\nu = \begin{bmatrix} \mathbf{I} - \mathbf{X} \mathbf{H}_\nu^{-1} \mathbf{X}^T \\ \sqrt{\nu} \mathbf{C} \mathbf{H}_\nu^{-1} \mathbf{X}^T \end{bmatrix}; \quad \mathbf{g}_\nu = \mathbf{G}_\nu \mathbf{y}.$$

It is now easier to solve the equation (5.5.18) using standard techniques. Zheng et al. (2018) have suggested to use the proximal gradient method for solving the problem, and named the algorithm as SR3, which is outlined in the following algorithm (10).

The tuning parameter η in this algorithm determines the step size of proximal-gradient method, with higher values being preferred for computational efficiency. Hence, we get the convergence in

Algorithm 10 SR3 for Penalized Regression

Initialize: $\mathbf{w}^{(0)}$, $t = 0$, $\eta \leq 1/\nu$.
1: **while** Not Converged **do**
2: $t = t + 1$
3: $\mathbf{w}^{(t)} = \text{prox}_{\eta\lambda R}(\mathbf{w}^{(t-1)} - \eta \mathbf{F}_\nu^T (\mathbf{F}_\nu \mathbf{w}^{(t-1)} - \mathbf{g}_\nu))$
4: **end while**

fewest steps if $\eta = 1/\nu$. After some basic algebra, it turns out that

$$\mathbf{w}^{(t-1)} - \frac{1}{\nu} \mathbf{F}_\nu^T (\mathbf{F}_\nu \mathbf{w}^{(t-1)} - \mathbf{g}_\nu) = \mathbf{H}_\nu^{-1} (\mathbf{X}^T \mathbf{y} + \nu \mathbf{C}^T \mathbf{w}^{(t-1)}) = \beta(\mathbf{w}^{(t-1)})$$

which is much easier to compute, bypassing computation of \mathbf{F}_ν and \mathbf{G}_ν . Zheng et al. (2018) have demonstrated that SR3 algorithm is expected to exhibit some robustness and computational efficiency. The major reason behind this is that splitting the objective functions into multiple parts releases the pressure of updating the parameter β in one step, especially for complicated penalty functions. Also, as we discussed, it is possible to partially minimize (5.5.17) w.r.t. β first and then w.r.t. \mathbf{w} , which is equivalent to solving (5.5.18) with nearly spherical level sets. On the contrary, the original problem (5.5.16) requires us to solve over elliptical level sets $\|\mathbf{y} - \mathbf{X}\beta\|_2^2$. Hence, it would be worthy extending this SR3 approach to the original convex-clustering problem for squared error loss. We name this new approach as ‘‘Sparse Relaxed Regularized Convex Clustering’’ (SR2C2). In this algorithm, we need to solve the following optimization problem

$$\min_{\mathbf{b}, \mathbf{v}} \frac{1}{2} \sum_{j=1}^p \|\hat{\pi}_j - \mathbf{b}_j\|_2^2 + \lambda \sum_{l \in \mathcal{E}} w_l \|\mathbf{v}_l\|_2 + \frac{\nu}{2} \sum_{l \in \mathcal{E}} \|\mathbf{v}_l - \mathbf{b}_{l_1} + \mathbf{b}_{l_2}\|_2^2. \quad (5.5.19)$$

Clearly, this is not the augmented Lagrangian problem, since we don’t force $\mathbf{v}_l = \mathbf{b}_{l_1} - \mathbf{b}_{l_2}$. Rather, the last squared term is a bit relaxed penalization so that \mathbf{v}_l and $\mathbf{b}_{l_1} - \mathbf{b}_{l_2}$ are not far from each other. Just like SR3, we first minimize the objective function w.r.t. \mathbf{b} , keeping \mathbf{v} fixed, and then substitute that solution $\mathbf{b}(\mathbf{v})$ (say) in the equation to get the update of \mathbf{v} . We only care about the updates of \mathbf{v}_l , and don’t use the solution of \mathbf{b}_j to construct the clusters. The reason behind this is that \mathbf{b}_j are some smooth function of the solution \mathbf{v}_l -s, and only the variables \mathbf{v}_l can be shrunk to 0. Hence, the clusters can be identified only from the solutions of \mathbf{v}_l , where $\mathbf{v}_l = 0$ implies $\hat{\pi}_{l_1}$ and $\hat{\pi}_{l_2}$ belong

to the same cluster. The final algorithm is outlined in the following algorithm (11).

Algorithm 11 SR2C2 for Squared Error Loss

Initialize $\mathbf{v}_l^{(0)} = \hat{\pi}_{l_1} - \hat{\pi}_{l_2}$

1: **for** $t = 1, 2, 3, \dots$ **do**

2: **for** $j = 1, 2, 3, \dots, p$ **do**

3: $\Delta_j^{(t)} = \left(\sum_{l_1=j} \mathbf{v}_l^{(t-1)} - \sum_{l_2=j} \mathbf{v}_l^{(t-1)} \right)$

4: **end for**

5: **for all** l **do**

6: $\mathbf{v}_l^{(t)} = \text{prox}_{\sigma_l \|\cdot\|_2} \left(\frac{1}{1+p\gamma} (\hat{\pi}_{l_1} - \hat{\pi}_{l_2}) + \frac{\gamma}{1+p\gamma} (\Delta_{l_1}^{(t)} - \Delta_{l_2}^{(t)}) \right)$

7: **end for**

8: **end for**

5.6 Simulation

In this section we will demonstrate how all these algorithms perform in different set-ups. First, we will compare the time complexity of the algorithms. Subsequently, we compare the clustering accuracy of them in terms of some quantitative measure.

5.6.1 Computational Complexity Comparison

We first compare the amount of time needed for significant convergence of the algorithms. In the first simulation, we generate SMM of order $m = 2$, with number of states $|\Sigma| = 4$. We divide this 16 possible pairs into four groups of four elements. For each group, the true transition probabilities are generated from a Dirichlet distribution with parameters $(e^{Z_1}, \dots, e^{Z_4})$ where $Z_i \sim \text{Unif}(0, 1)$. The

true transition probability vectors for the 4 groups are as follows.

$$\mathbf{R} = \begin{matrix} & \begin{matrix} \text{State 1} & \text{State 2} & \text{State 3} & \text{State 4} \end{matrix} \\ \begin{matrix} \left[\begin{array}{cccc} 0.197 & 0.454 & 0.013 & 0.336 \\ 0.504 & 0.147 & 0.225 & 0.124 \\ 0.071 & 0.403 & 0.329 & 0.197 \\ 0.023 & 0.271 & 0.017 & 0.689 \end{array} \right] & \begin{matrix} \text{Group 1} \\ \text{Group 2} \\ \text{Group 3} \\ \text{Group 4} \end{matrix} \end{matrix}$$

Next we generate an SMM of $n = 1000$ using the true probabilities. We have specified the weights $w_{i,j} = \exp[-10\|\hat{\pi}_i - \hat{\pi}_j\|_\infty]$. $\mathcal{S}(\hat{\pi}_i$ is one of the k -nearest neighbor of $\hat{\pi}_j$ or vice versa). This is a similar set-up used in the simulation set-up of Majumder et al. (2022), we have changed the true parameters only in this paper. The values of δ_j are fixed as N_{σ_j}/N , the proportion of j^{th} m -tuple in the sequence. We replicate the scenario for 100 times and report the total time needed to execute our methods for 10 different λ values for all the algorithms. The results are presented in the following figures.

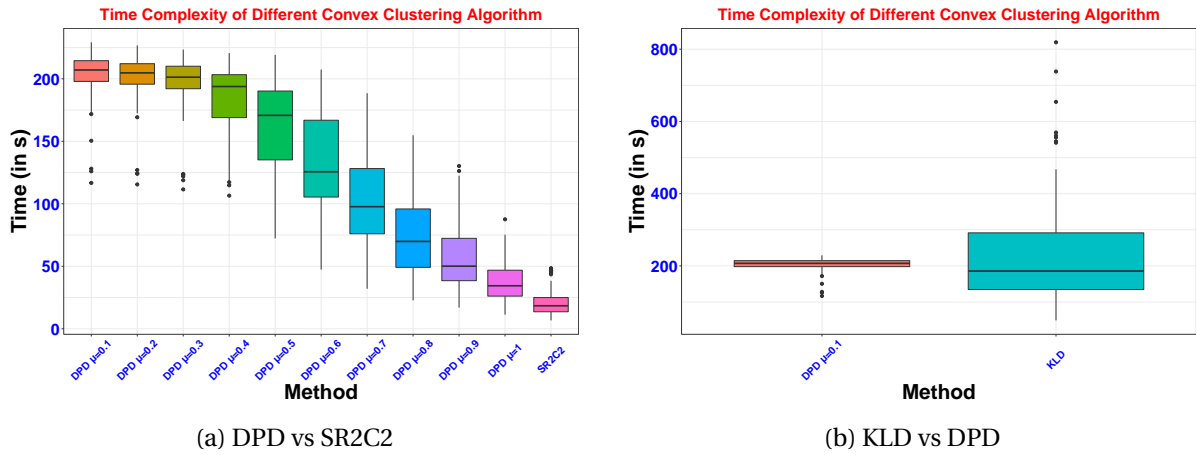


Figure 5.1: Box plot for total time elapsed for 100 replications, using different algorithms

From the figure (5.1a), we can see newly proposed SR2C2 algorithm converges faster than any other algorithm, including the usual squared error loss ($\mu = 1$). Also, as the value of μ decreases

towards 0, amount of time elapsed for convergence also increases. This implies, for DPD, higher the modulus of strong convexity of the loss function is, time complexity is lower. For SR2C2, we need to update lesser number of variables, however that is similar type of updates as compared to the squared error loss. The intrinsic feature of the SR2C2 algorithm makes it more easier for one to converge to the solution of the optimization problem.

For the ADMM, we have tracked the both primal and dual residuals for stopping criteria, while for AMA, we have tracked the duality gap. An illustration have been presented in the figure (5.2) and the figure (5.3). For KLD, we have some sort of approximation while performing the ADMM. This is the reason behind the fact that the primal and the dual residuals are not decreasing monotonically, however they have an overall decreasing pattern. On the other hand, AMA for solving the clustering problem for DPD loss considers exact solution in each step, and the duality gap reduces monotonically in each step.

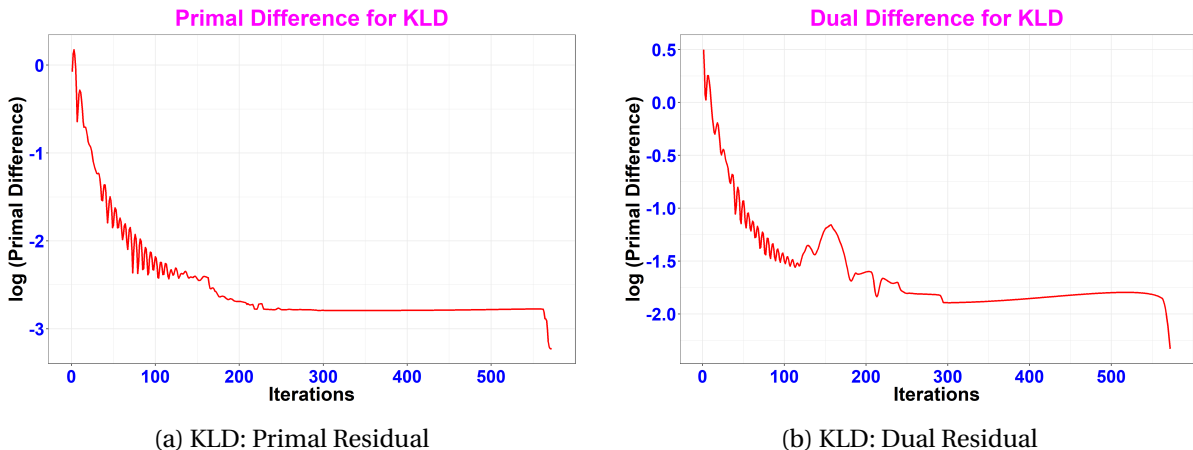


Figure 5.2: Primal and Dual residuals for Kullback-Leibler Divergence, in \log_{10} scale

From the figure (5.1b), we observe that the SMMGECCO algorithm for Kullback-Leibler divergence (DPD with $\mu = 0$) is slightly better in terms of average time elapsed. However, the box plot suggests that the variability of time complexity in KLD is really high. For some iterations, the time elapsed is more than 600 seconds, where the average is just below 200 seconds. This is a concerning issue of using the KLD measure. We have to use ADMM in this case since the loss is not strongly

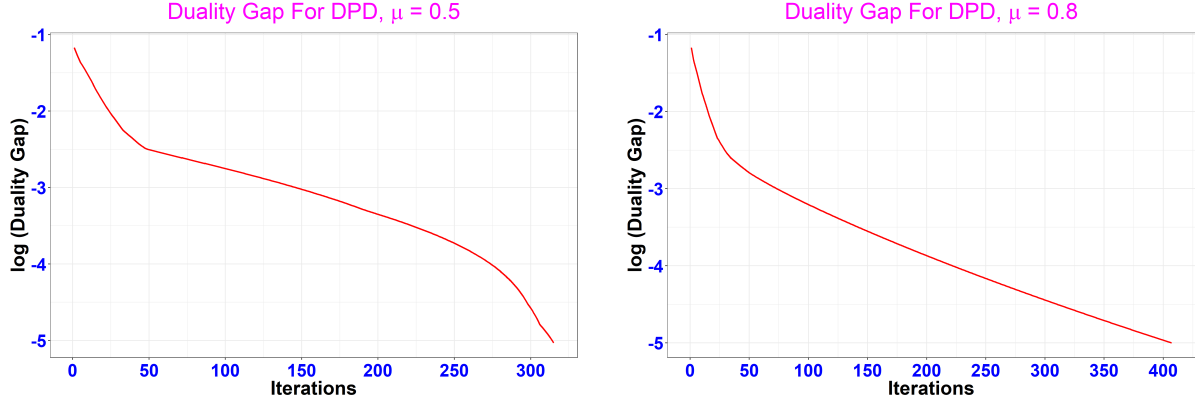


Figure 5.3: Duality gap for DPD, in \log_{10} scale

convex, and for fact ADMM is relatively slow. However, use of KLD could be justified if the clustering accuracy is good, which we will discuss in the next subsection.

5.6.2 Clustering Performance

Simulation Set Up 1

In this simulation study as well, we generate Markov chains of $n = 1000$ and $m = 2$ using the same parameters as we did in the experiment of time complexity comparison. In this experiment, we have been particularly cautious about the selection of the weights $w_{i,j}$. We have discussed that $w_{i,j}$ takes the form of $\exp[-\phi g(\|\hat{\pi}_i - \hat{\pi}_j\|)]$, $\mathcal{I}(\hat{\pi}_i$ is one of the k -nearest neighbor of $\hat{\pi}_j$ or vice versa) for some norm $\|\cdot\|$ for some constant $\phi > 0$. Traditionally, there were no data-dependent measure of ϕ , until Chi and Steinerberger (2019) have come up with a strategy to address this issue. They have proposed $w_{i,j} = \exp[-\frac{\|\hat{\pi}_i - \hat{\pi}_j\|_2^2}{t_i t_j}]$, where t_i is the median ℓ_2 distance of 5 nearest neighbours of $\hat{\pi}_i$. Following that approach, we modify the weights a little bit by taking $w_{i,j} = \exp[-\frac{1}{\sqrt{t_i t_j}} \|\hat{\pi}_i - \hat{\pi}_j\|_\infty]$, $\mathcal{I}(\hat{\pi}_i$ is one of the k -nearest neighbor of $\hat{\pi}_j$ or vice versa), where t_i is median ℓ_∞ distance of k -nearest neighbors of $\hat{\pi}_i$. In our experiment we fix $k = 3$ and the nearest neighbour is determined w.r.t. the ℓ_∞ distance. As we have discussed before, we take $\delta_j = N_{\sigma_j}/N$.

We compare the model selection performance for 12 different loss functions ρ : Kullback-Leibler Divergence, DPD with $\mu = 0.1, 0.2, \dots, 1$ and the newly proposed SR2C2 method. For each replication

of the same experiment, we perform our clustering algorithms for a range of λ - values, and select the model which minimizes the BIC criterion. We also assess the model selection by EBIC criterion and compare it with the BIC one. In either case, we report the Adjusted Rand Index (ARI) where ARI value near 1 is desirable. We repeat the experiment 100 times and compute the mean ARI empirically. We also report the probability of perfect cluster recovery ($ARI = 1$), computed empirically by the proportion of replications with $ARI = 1$. The results are presented in the following tables.

Table 5.1: Model selection performance for all algorithms.

Methods	BIC		EBIC		
	ARI	Prop. of Perfect Recovery	ARI	Prop. of Perfect Recovery	
KLD	0.9028	0.54	0.9028	0.54	
DPD	$\mu = 0.1$	0.9182	0.6	0.9182	0.6
	$\mu = 0.2$	0.9187	0.61	0.9187	0.61
	$\mu = 0.3$	0.9170	0.61	0.9170	0.61
	$\mu = 0.4$	0.9179	0.61	0.9179	0.61
	$\mu = 0.5$	0.9175	0.61	0.9175	0.61
	$\mu = 0.6$	0.9172	0.61	0.9172	0.61
	$\mu = 0.7$	0.9152	0.6	0.9152	0.6
	$\mu = 0.8$	0.9154	0.61	0.9154	0.61
	$\mu = 0.9$	0.9174	0.6	0.9174	0.6
	$\mu = 1$	0.9155	0.6	0.9157	0.6
$\mu = 1, \delta_j = 1$	0.9181	0.61	0.9181	0.61	
SR2C2	0.9147	0.59	0.9147	0.59	

From the results, it is clear that there are practically no difference between the loss functions in terms of perfect recovery. However in this small margin of differentiation, KLD performs relatively poor in compared to others, especially in the probability of recovery of the true cluster. The most important takeaway from this experiment is SR2C2 performs almost similar as DPD loss with different values of μ . SR2C2 takes much less time compared to the others, especially to the squared

error loss. Hence, our new SR2C2 method can be used as an alternative to the existing methods.

Simulation Set Up 2

In the previous example, we observe that the transition probability matrix is well separated which results in good clustering result for all the models. However, it might be worthwhile to explore how the performances vary when the rows of \mathbf{R} is not well separated. To demonstrate that, we keep our original simulation framework, with a new \mathbf{R} matrix as follows:

$$\mathbf{R} = \begin{array}{cccc} & \text{State 1} & \text{State 2} & \text{State 3} & \text{State 4} \\ \left[\begin{array}{cccc} 0.114 & 0.139 & 0.218 & 0.529 \\ 0.211 & 0.295 & 0.096 & 0.398 \\ 0.273 & 0.314 & 0.123 & 0.290 \\ 0.144 & 0.419 & 0.115 & 0.322 \end{array} \right] & \text{Group 1} & \text{Group 2} & \text{Group 3} & \text{Group 4} \end{array}$$

We only perform our clustering for 100 replications for four cases: DPD with $\mu = 0.5$ and $\delta_j = N_{\sigma_j}/N$, DPD with $\mu = 1$ and $\delta_j = N_{\sigma_j}/N$, DPD with $\mu = 1$ and $\delta_j = 1$ (the usual convex clustering) and SR2C2. We find the mean ARI for these 4 approaches to be 0.27, 0.289, 0.266 and 0.257. From this simulation we can conclude that adjusting the weights according to the frequency of the histories give us better result in such problems as compared to the case when $\delta_j = 1$. The performance of SR2C2 is relatively poor, but if we want to sacrifice the model selection accuracy to some extent for computational benefit, it will be still beneficial.

5.7 Discussion

Overall, this chapter opens up a new direction beyond fitting sparse Markov models. The constrained optimization techniques can be used in similar optimization problem where we need to restrict the parameters in a linear hyperplane. We also suggest a new faster method SR2C2 which can be used to speed up the clustering problem. Incorporation of the weights δ_j in the loss function is another major modification of the existing techniques, which pays dividend in identifying the clusters

especially when the clusters are not well separated and the sample size is not high. Theoretical result about true cluster recovery is also a significant contribution to the convex clustering theory, as this deals with a class of loss functions for which the general computation is not easy. In terms of practical aspects, this methods will have wide applicability in the analysis of sequence data, for example in the virus classification problem discussed in chapter 4. It might have wide applicability in any general clustering problem as well.

CHAPTER

6

PREDICTION IN SPARSE MARKOV MODELS

6.1 Introduction

In many statistical problems, prediction for sequential data is of utmost interest. Use of higher order Markov models could be advantageous for prediction since in such models in two ways. First, we characterise a sequence in terms of transition probabilities, which are indeed future probabilities of occurrence of a state given the past states. Secondly, the order of the chain m allows us to use the past information upto suitable lag. In literature, there are many prediction algorithms which exploit this Markov structure. Begleiter et al. (2004) have developed some prediction methods in the VLMC set-up by extending well-established prediction algorithms such as Context Tree Weighting (CTW) or Probabilistic Suffix Tree (PST). Rabiner (1989) first introduced hidden Markov models (HMM) which are capable of modelling and predicting complex sequential data. In the SMM set-

up, Jääskinen et al. (2014) and Xiong et al. (2016) have used their Bayesian approach for fitting SMM for prediction purpose. However, it still needs to be investigated how we predict the h -step future observation X_{n+h} , given the observed sequence X_1, \dots, X_n . It is also worthwhile to construct a $100(1 - \alpha)\%$ simultaneous prediction set for the intermediate possible values $(X_{n+1}, \dots, X_{n+h})$. In this chapter, we develop a bootstrap based method for predicting the h -step ahead observation X_{n+h} . We also develop a score based method for constructing the simultaneous prediction set in the SMM set-up, where the score associated with each h -tuple determines whether to keep that tuple in the prediction set .

The rest of the chapter is organized as follows. In Section 6.2, we develop a score based method for constructing the prediction sets in Markov models in general, with an extension to the sparse Markov model set-up. For predicting the h -step ahead observation in the future, we propose a bootstrap based prediction method in Section 6.3. Extensive simulation studies have been conducted to establish the utility of such method. A real data analysis involving anomaly detection in a specific gene area for *Helicobacter Pylori* bacteria is carried out in Section 6.4, which makes use of the anomaly score method. We conclude this paper by summarizing our findings in Section 6.5.

6.2 Score Based Simultaneous Prediction Set in SMM

Here we focus on constructing prediction sets for the future events in the SMM set-up. Let $\{X_1, \dots, X_n\}$ be an SMM of order m . Our goal is to construct a prediction set for the next h observations, namely $(X_{n+1}, \dots, X_{n+h})$. Note that, if the state space Σ has cardinality d , then there are d^h many possible h -tuples for the future h observations. Computing the probability of each of these h -tuples is computationally inefficient for large values of h . Thus, we have to come up with a suitable algorithm so that we are able to construct prediction sets with smaller number of elements, having $100(1 - \alpha)\%$ coverage for some level α . Obviously, this task is computationally very expensive when we have to deal with m -th order Markov models where there are d^m many transition probability vectors. In such scenarios, use of SMM could be beneficial in reducing the total number of parameters in an efficient way. However, even for the most simple case of order one

Markov models, it is not very easy to construct the prediction set for any given transition matrix P without extra assumptions.

6.2.1 Prediction in Simple Order One Markov Models

To start with, we first consider the formation of prediction sets for order one Markov models. The following theorem gives us an idea how the transition matrix P be structured, so that we are able to construct optimal prediction sets.

Theorem 6.2.1 *Suppose $\{X_1, \dots, X_n\}$ is an order one Markov chain over the state space Σ with $|\Sigma| = d$. Suppose P is the transition matrix with the following properties.*

- (i) *The chain is aperiodic and irreducible.*
- (ii) *Σ can be partitioned into g disjoint groups $\Sigma_1, \Sigma_2, \dots, \Sigma_g$ with cardinalities d_1, d_2, \dots, d_g , and there exist small positive real numbers $\delta \leq \epsilon \ll 0.5$ such that for any $i \in \Sigma_k$ for some k ,*

$$1 - \epsilon \leq \sum_{j \in \Sigma_k} p_{ij} \leq 1 - \delta. \quad (6.2.1)$$

- (iii) *Denote, $d_{max} = \max\{d_1, \dots, d_g\}$, $\beta = \left(\frac{1}{1 - \delta + \epsilon}\right)(\epsilon + \sqrt{2\epsilon(1 - \delta)\log(1 - \delta + \epsilon)})$, and $\eta_* = \frac{(1 - 1/g - \beta)}{\sqrt{(1/g)(1 - 1/g)}}$. Assume,*
- $$\log\left(\frac{g d_{max}}{d}\right) < \frac{1}{2\eta_*^2}. \quad (6.2.2)$$

In that case, given $X_n = i$, we can construct a $100(1 - \alpha)\%$ prediction set $C(i; \alpha, h)$ for $(X_{n+1}, \dots, X_{n+h})$ such that $\frac{|C(i; \alpha, h)|}{d^h} \rightarrow 0$ as $h \rightarrow \infty$.

Proof: In this set up, in-group transitions have more probability. So ideally the sequence of random variables $(X_{n+1}, \dots, X_{n+h})$ will have too many group switching with lower probabilities. Here group switching means transition from X_{n+r} to X_{n+r+1} occurs from one of the groups $\Sigma_1, \dots, \Sigma_g$ to any other group. To get an efficient prediction set, which covers more probability with less number of h -tuples, we need to take those h -tuples for which the number of group switches are small.

Without loss of generality, assume $X_n = i_0 \in \Sigma_1$. Now, we will construct the prediction sets in different scenarios as follows.

Case 1: Let $\epsilon = \delta$, with $d_1 = d_2 = \dots = d_g$. So, starting from any state, probability of staying in the same group in the next step is exactly $1 - \epsilon$, and probability of changing the group is ϵ . Now, number of h -tuple with exactly k switches can be described as the following set:

$$S(i_0, k) = \left\{ (i_1, \dots, i_h) : \sum_{j=1}^h \mathcal{I}(i_{j-1} \text{ and } i_j \text{ do not belong to same group}) = k \right\}.$$

So,

$$P\left((X_{n+1}, \dots, X_{n+h}) \in S(i_0, 0) \mid X_n = i_0\right) = \sum_{(i_1, \dots, i_h) \in \Sigma_1^h} p_{i_0 i_1} p_{i_1 i_2} \dots p_{i_{h-1} i_h} = (1 - \epsilon)^h.$$

Now, the major challenge is to compute the probability of $S(i_0, k)$ for $k \geq 1$ in a proper manner. If $X_n = i_0 \in \Sigma_1$, and $k = 1$ then the h -tuple $(i_1, \dots, i_h) \in S(i_0, 1)$ if for some $j \in \{1, 2, \dots, h\}$, $i_0, \dots, i_{j-1} \in \Sigma_1$ and $i_j, i_{j+1}, \dots, i_h \in \Sigma_u$ for some $u \neq 1$. Thus,

$$\begin{aligned} P\left((X_{n+1}, \dots, X_{n+h}) \in S(i_0, 1) \mid X_n = i_0\right) &= \sum_{j=1}^h \sum_{u=2}^g \sum_{\substack{(i_1, \dots, i_{j-1}) \in \Sigma_1^{j-1} \\ (i_j, \dots, i_h) \in \Sigma_u^{h-j+1}}} p_{i_0 i_1} p_{i_1 i_2} \dots p_{i_{h-1} i_h} \\ &= \sum_{j=1}^h \sum_{u=2}^g \sum_{\substack{(i_1, \dots, i_{j-1}) \in \Sigma_1^{j-1} \\ i_j \in \Sigma_u}} p_{i_0 i_1} p_{i_1 i_2} \dots p_{i_{j-1} i_j} (1 - \epsilon)^{h-j} \\ &= \sum_{j=1}^h (1 - \epsilon)^{h-j} \sum_{(i_1, \dots, i_{j-1}) \in \Sigma_1^{j-1}} p_{i_0 i_1} p_{i_1 i_2} \dots p_{i_{j-2} i_{j-1}} \sum_{i_j \in \Sigma'_1} p_{i_{j-1} i_j} \\ &= \sum_{j=1}^h \epsilon (1 - \epsilon)^{h-j} \sum_{(i_1, \dots, i_{j-1}) \in \Sigma_1^{j-1}} p_{i_0 i_1} p_{i_1 i_2} \dots p_{i_{j-2} i_{j-1}} \\ &= \sum_{j=1}^h \epsilon (1 - \epsilon)^{h-j} (1 - \epsilon)^{j-1} = h \epsilon (1 - \epsilon)^{h-1}. \end{aligned}$$

By similar argument, we can say that for any k number of group switches, $P\left((X_{n+1}, \dots, X_{n+h}) \in S(i_0, k) \mid X_n = i_0\right) = \binom{h}{k} \epsilon^k (1 - \epsilon)^{h-k}$. Clearly, this probability is same as $P(Z = k)$, where $Z \sim \text{Binom}(h, \epsilon)$. In other words, if Z is the number of group switches in the sequence of events

X_{n+1}, \dots, X_{n+h} , then $Z \sim Binom(h, \epsilon)$, given $X_n = i_0$.

The obvious question is how we can construct $C(i_0; \alpha, h)$ from $S(i, k)$, more specifically, how many group switches we are going to allow. We start from $k = 0$, and stop as soon as $F(k; h, \epsilon) = \sum_{j=0}^k \binom{h}{j} \epsilon^j (1-\epsilon)^{h-j} \geq 1-\alpha$. Hence k is the $(1-\alpha)^{th}$ quantile of $Binomial(h, \epsilon)$ distribution, and $F(k; h, \epsilon)$ is the c.d.f of that distribution. However, we don't have a closed form equation of $F(k; h, \epsilon)$, so it may be challenging to find the optimal k . For sufficiently large h , we are able to express the c.d.f by Normal approximation. We can write $F(k; h, \epsilon) = P\left(\sum_{i=1}^h Y_i \leq k\right)$, where $Y_i \stackrel{iid}{\sim} Binom(h, \epsilon)$. Hence,

$$\begin{aligned} P\left(\sum_{i=1}^h Y_i \leq k\right) &= P\left(\frac{\sum_{i=1}^h Y_i - h\epsilon}{\sqrt{h\epsilon(1-\epsilon)}} \leq \frac{k - h\epsilon}{\sqrt{h\epsilon(1-\epsilon)}}\right) = 1 - \alpha \\ \implies k &= \left[h\epsilon + \sqrt{h\epsilon(1-\epsilon)} z_\alpha \right], \end{aligned}$$

where z_α is the $(1-\alpha)^{th}$ quantile of $\mathcal{N}(0, 1)$.

Now, let us calculate the number of h -tuples are there in $C(i_0; \alpha, h)$, i.e compute the cardinality of the sets $S(i, j)$ for $j = 0, 1, \dots, k$. Note that, we have assumed all the group sizes are same, and equal to d_1 . Clearly, $|S(i_0, 0)| = d_1^h$. The way we have constructed $S(i_0, j)$ for $j \geq 1$, we can conclude $|S(i_0, j)| = \binom{h}{j} d_1^h (g-1)^j$. Hence,

$$\begin{aligned} |C(i_0; \alpha, h)| &= \sum_{j=0}^k |S(i_0, j)| = d_1^h \sum_{j=0}^k \binom{h}{j} (g-1)^j \\ \implies |C(i_0; \alpha, h)| &= (g d_1)^h \sum_{j=0}^k \binom{h}{j} (1/g)^{h-j} (1-1/g)^j = d^h F(k; h, 1-1/g) \\ \implies \frac{|C(i_0; \alpha, h)|}{d^h} &= F(k; h, 1-1/g) \approx \Phi\left(\frac{k - h(1-1/g)}{\sqrt{h(1/g)(1-1/g)}}\right) \\ \implies \frac{|C(i_0; \alpha, h)|}{d^h} &\approx \Phi\left(\frac{h\epsilon + \sqrt{h\epsilon(1-\epsilon)} z_\alpha - h(1-1/g)}{\sqrt{h(1/g)(1-1/g)}}\right) \\ &= \Phi\left(\frac{\sqrt{h}(\epsilon - 1 + 1/g)}{\sqrt{(1/g)(1-1/g)}} + z_\alpha \sqrt{\frac{\epsilon(1-\epsilon)}{(1/g)(1-1/g)}}\right) \\ &= \Phi\left(-\sqrt{h}\eta + c\right), \end{aligned}$$

where $\eta = \frac{(1-1/g-\epsilon)}{\sqrt{(1/g)(1-1/g)}} > 0$ and $c = z_\alpha \sqrt{\frac{\epsilon(1-\epsilon)}{(1/g)(1-1/g)}}$. We know that for $x \rightarrow \infty$, $\Phi(-x) \sim \frac{\phi(x)}{x}$.

Hence,

$$\begin{aligned}\Phi(-\sqrt{h}\eta + c) &\approx \frac{\phi(\sqrt{h}\eta - c)}{(\sqrt{h}\eta - c)} \leq c_0 \exp\left\{-\frac{1}{2}(\sqrt{h}\eta - c)^2\right\} \\ &\leq c_0 \exp\left\{-\frac{h}{2\eta^2} + \sqrt{h}\eta c - \frac{c^2}{2}\right\} \rightarrow 0 \quad \text{as } h \rightarrow \infty.\end{aligned}$$

Case 2: Here, consider $\epsilon = \delta$, but d_1, \dots, d_g are different. Let $d_{max} = \max\{d_1, \dots, d_g\}$. Then construct the prediction set for large h as before. But the counts will be different, and we will try to bound $|S(i_0, j)|$. We notice that, $|S(i_0, j)| \leq d_{max}^h \binom{h}{j} (g-1)^j$, and hence

$$\begin{aligned}\frac{|C(i_0; \alpha, h)|}{d^h} &\leq \left(\frac{g d_{max}}{d}\right)^h F(k; h, 1-1/g) \\ &\leq c_0 \exp\left\{h\left(\log\left(\frac{g d_{max}}{d}\right) - \frac{1}{2\eta^2}\right) + \sqrt{h}\eta c - \frac{c^2}{2}\right\}.\end{aligned}\tag{6.2.3}$$

So, if we assume $\log\left(\frac{g d_{max}}{d}\right) - \frac{1}{2\eta^2} < 0$, the proportion will go to 0 exponentially. We can think $\log\left(\frac{g d_{max}}{d}\right)$ as a penalty term for the departure from the symmetry.

Case 3: Now assume $\delta < \epsilon$ and the groups are of unequal size as well. We take the exact same approach as we did before. Since, we don't have exact equality, we cannot claim $\gamma(i_0; j, h) = P\left((X_{n+1}, \dots, X_{n+h}) \in S(i_0, j) \mid X_n = i_0\right) = \binom{h}{j} \epsilon^j (1-\epsilon)^{h-j}$. Instead, we can say

$$\gamma(i_0; j, h) \leq \binom{h}{j} \epsilon^j (1-\delta)^{h-j}.\tag{6.2.4}$$

Now try to bound the tail probability $\sum_{j=k}^h \gamma(i_0; j, h)$. Define $\epsilon_* = \frac{\epsilon}{1-\delta+\epsilon}$. Let us use the previous $k = h\epsilon + \sqrt{h\epsilon(1-\epsilon)}z_\alpha$. Then after some algebraic manipulation and for some $c(h) \rightarrow 0$ as $h \rightarrow \infty$ we get,

$$\begin{aligned}\sum_{j=k}^h \gamma(i_0; j, h) &\leq c(h)(1-\delta+\epsilon)^h \exp\left\{-\frac{h(\epsilon-\epsilon_*)^2}{2\epsilon_*(1-\epsilon_*)} - \frac{\sqrt{h}(\epsilon-\epsilon_*)\sqrt{\epsilon(1-\epsilon)}}{\epsilon_*(1-\epsilon_*)} z_\alpha - \frac{\epsilon(1-\epsilon)}{2\epsilon_*(1-\epsilon_*)} z_\alpha^2\right\} \\ &= c(h) \exp\left\{h\left(\log((1-\delta+\epsilon)) - \frac{(\epsilon-\epsilon_*)^2}{2\epsilon_*(1-\epsilon_*)}\right) - \frac{\sqrt{h}(\epsilon-\epsilon_*)\sqrt{\epsilon(1-\epsilon)}}{\epsilon_*(1-\epsilon_*)} z_\alpha - \frac{\epsilon(1-\epsilon)}{2\epsilon_*(1-\epsilon_*)} z_\alpha^2\right\}.\end{aligned}$$

We want this tail probability $\leq \alpha$. However, $\log((1-\delta+\epsilon)) - \frac{(\epsilon-\epsilon_*)^2}{2\epsilon_*(1-\epsilon_*)} > 0$ for small values of δ and ϵ with $\epsilon > \delta$, making the RHS going to ∞ . This leads to a trivial bound. Hence, our previous choice of k won't work, and probably the choice $k' = h\beta + \sqrt{h\epsilon(1-\epsilon)}z_\alpha$ for some $\beta > \epsilon$ will work. The rationale behind this choice is obvious, as the coefficient of β in the expression of k plays the key role determining the exponential bound. Doing some similar calculation, we get

$$\sum_{j=k'}^h \gamma(i_0; j, h) \leq c(h) \exp \left\{ h \left(\log((1-\delta+\epsilon)) - \frac{(\beta-\epsilon_*)^2}{2\epsilon_*(1-\epsilon_*)} \right) - \frac{\sqrt{h}(\beta-\epsilon_*)\sqrt{\epsilon(1-\epsilon)}}{\epsilon_*(1-\epsilon_*)} z_\alpha - \frac{\epsilon(1-\epsilon)}{2\epsilon_*(1-\epsilon_*)} z_\alpha^2 \right\}.$$

The coefficient of \sqrt{h} in the exponent is strictly negative, and the third term is a constant. Hence, if we equate the coefficient of h in the exponent to be zero, the tail probability will go to 0 on the rate $\exp(-r\sqrt{h})$ for some $r > 0$. This will give the choice of β as

$$\beta = \epsilon_* + \sqrt{2\epsilon_*(1-\epsilon_*)\log(1-\delta+\epsilon)} = \left(\frac{1}{1-\delta+\epsilon} \right) \left(\epsilon + \sqrt{2\epsilon(1-\delta)\log(1-\delta+\epsilon)} \right).$$

Note that, for $\delta = \epsilon$, we get $\beta = \epsilon$. Hence, this expression will generalize the previous cases. Subsequently, the choice of k will turn out to be $k' = h\beta + \sqrt{h\epsilon(1-\epsilon)}z_\alpha$. We can improve this expression a little bit by taking $k' = h\beta + \sqrt{h\epsilon_*(1-\epsilon_*)}z_\alpha$. While counting the size of $C(i_0; \alpha, h)$, we use similar argument as in previous cases, and get

$$\begin{aligned} \frac{|C(i_0; \alpha, h)|}{d^h} &\leq \left(\frac{g d_{max}}{d} \right)^h F(k'; h, 1-1/g) \\ &\approx \left(\frac{g d_{max}}{d} \right)^h \Phi \left(\frac{\sqrt{h}(\beta-1+1/g)}{\sqrt{(1/g)(1-1/g)}} + z_\alpha \sqrt{\frac{\epsilon_*(1-\epsilon_*)}{(1/g)(1-1/g)}} \right) \\ &\leq c_0 \left(\frac{g d_{max}}{d} \right)^h \exp \left\{ -\frac{1}{2} (\sqrt{h}\eta_* - c_*)^2 \right\} \\ &\leq c_0 \exp \left\{ h \left(\log \left(\frac{g d_{max}}{d} \right) - \frac{1}{2\eta_*^2} \right) + \sqrt{h}\eta_* c_* - \frac{c_*^2}{2} \right\}. \end{aligned}$$

where $\eta_* = \frac{(1-1/g-\beta)}{\sqrt{(1/g)(1-1/g)}}$ and $c_* = z_\alpha \sqrt{\frac{\epsilon_*(1-\epsilon_*)}{(1/g)(1-1/g)}}$. By our assumption, $\log \left(\frac{g d_{max}}{d} \right) - \frac{1}{2\eta_*^2} < 0$. Hence, $\frac{|C(i_0; \alpha, h)|}{d^h} \rightarrow 0$ exponentially fast as $h \rightarrow \infty$. [PROVED]

In the above setup, we have a particular structure of the transition matrix so that one group

interacts with other with very low probability. However, this structure is very restrictive in general. We may not be able to get such partition of Σ after estimating the transition matrix. Also we cannot extend this procedure for prediction in higher order Markov models including SMM and VLMC, since we cannot control the behaviour of group transitions arbitrarily in that complex scenario. To resolve this issue, we will now discuss a slightly different procedure for constructing the prediction sets. First consider order one Markov chains. Instead of looking for the partition of Σ with higher in-group transition probability, we look at each row of P , and predict the next transition with higher probability. In other words, if we look at the i^{th} row of the transition matrix P , we choose the higher values in that row in decreasing order so that the sum of these probabilities is more than $1 - \epsilon$ for some small ϵ . Denote the r^{th} highest element in the i^{th} row as $p_{i,(r)}$. So, for each $i \in \Sigma$, we construct the prediction set $C(i; \epsilon, 1)$ for X_{n+1} , given $X_n = i$ with coverage probability at least $1 - \epsilon$ as follows:

$$C(i; \epsilon, 1) = \{j_1, \dots, j_r \in \Sigma : p_{i,j_s} = p_{i,(s)}, \sum_{s=1}^r p_{i,j_s} \geq 1 - \epsilon\}.$$

Clearly this mechanism is more general than the previous one, and it will eventually helpful for constructing prediction sets for larger class of transition matrices. However, we still need to construct the $100(1 - \alpha)\%$ prediction set for $(X_{n+1}, \dots, X_{n+h})$. The procedure will be similar as before, with little modification as follows. Let us denote $r_i = |C(i; \epsilon, 1)|$, and $d_1 = \max\{r_1, \dots, r_d\}$. For simplification of our counting, we can assume all the r_i are same and equal to d_1 , as the coverage probability of the set $C(i; \epsilon, 1)$ will even increase if we throw some more elements into that set. In other words, we will take the top d_1 many elements from each row of P such that $d_1 < d$ and

$$1 - \epsilon \leq \sum_{s=1}^{d_1} p_{i,(s)} \leq 1 - \delta$$

for some small $\epsilon > 0$ and $\delta \leq \epsilon$. Just like before, define

$$S(i_0, j) = \{(i_1, \dots, i_h) \in \Sigma^h : \sum_{s=1}^h \mathcal{I}(i_s \notin C(i_{s-1}; \epsilon, 1)) = j\}.$$

After similar algebraic manipulation, we get

$$\gamma(i_0; j, h) = P\left((X_{n+1}, \dots, X_{n+h}) \in S(i_0, j) | X_n = i_0\right) \leq \binom{h}{j} \epsilon^j (1 - \delta)^{h-j},$$

and thus we can construct the h step prediction set as

$$C(i_0; \alpha, h) = \bigcup_{j=0}^k S(i_0, j)$$

where $k = h\beta + \sqrt{h\epsilon_*(1 - \epsilon_*)}z_\alpha$. Now, note that $|S(i_0, j)| = \binom{h}{j} d_1^{h-j} d_2^j$, where $d_2 = d - d_1$, and hence

$$\frac{|C(i_0; \alpha, h)|}{d^h} = \sum_{j=0}^k \binom{h}{j} \left(\frac{d_1}{d}\right)^{h-j} \left(\frac{d_2}{d}\right)^j \approx \Phi\left(\frac{k - h\frac{d_2}{d}}{\sqrt{\frac{hd_1d_2}{d^2}}}\right) = \Phi\left(\frac{\sqrt{h}(\beta - \frac{d_2}{d})}{\sqrt{\frac{d_1d_2}{d^2}}} + z_\alpha \sqrt{\frac{d^2\epsilon_*(1 - \epsilon_*)}{d_1d_2}}\right).$$

The above proportion goes to 0 exponentially fast as $h \rightarrow \infty$ if $\beta < \frac{d_2}{d}$, and this is the only condition we need to establish the result. Recall that $\beta = \left(\frac{1}{1 - \delta + \epsilon}\right)\left(\epsilon + \sqrt{2\epsilon(1 - \delta)\log(1 - \delta + \epsilon)}\right)$. In practice, for small ϵ , β is at most equal to $(\sqrt{2} + 1)\epsilon$. If we take $d_1 = d - 1$, then we can safely take $\epsilon = 1/d$, since the only one not in $C(i; \epsilon, 1)$ is the smallest probability in the i^{th} row. Subsequently, the condition turns out to be $\min_j p_{ij} < (1/(\sqrt{2} + 1)d)$ for each $i \in \Sigma$. This we can expect to hold, otherwise the transition probabilities will not be very different for a given row, resulting in the Markov model to be closer to independence. The following theorem summarizes all these things discussed above.

Theorem 6.2.2 *Suppose $\{X_0, \dots, X_n\}$ is an order one Markov chain over the state space Σ with $|\Sigma| = d$. Suppose P is the transition matrix with the following properties.*

- (i) *The chain is aperiodic and irreducible.*
- (ii) *For each row in the transition matrix P , we take the highest d_1 many entries $p_{i,(1)} \geq p_{i,(2)} \geq \dots \geq p_{i,(d_1)}$ and the corresponding elements j_1, \dots, j_{d_1} in Σ such that*

$$1 - \epsilon \leq \sum_{r=1}^{d_1} p_{i,j_r} \leq 1 - \delta \tag{6.2.5}$$

for some small positive real numbers $\delta \leq \epsilon \ll 0.5$.

(iii) Denote, $\beta = \left(\frac{1}{1-\delta+\epsilon}\right)\left(\epsilon + \sqrt{2\epsilon(1-\delta)\log(1-\delta+\epsilon)}\right)$, and $d_2 = d - d_1$. Assume,

$$\beta < d_2/d. \quad (6.2.6)$$

In that case, given $X_n = i$, we can construct a $100(1-\alpha)\%$ prediction set $C(i; \alpha, h)$ for $(X_{n+1}, \dots, X_{n+h})$ such that

$$\frac{|C(i; \alpha, h)|}{d^h} \leq C_0 \exp\left[-\frac{hd^2(\beta - \frac{d_2}{d})^2}{2d_1d_2}\right] \rightarrow 0$$

for some constant C_0 as $h \rightarrow \infty$.

The major consequence of this theorem is that we are able to extend this result in higher order Markov chain. In that case, we will consider the first d_1 many elements in Σ with higher probabilities for each of the conditional distribution given a history $\mathbf{x}_0 = (j_1, \dots, j_m)$ of length m with coverage at least $1-\epsilon$. When we construct the prediction sets, we define the above one-step prediction set as $C(\mathbf{x}; \epsilon, 1)$ for any $\mathbf{x} \in \Sigma^m$, and

$$S(\mathbf{x}_0, j) = \left\{ (i_1, \dots, i_h) \in \Sigma^h : \sum_{j=1}^h \mathcal{I}(i_j \notin C(\mathbf{x}_{j-1}; \epsilon, 1)) = j, \mathbf{x}_j \text{ to be the } m\text{-tuple} \right. \\ \left. \text{consisting of the last } m \text{ elements of } (\mathbf{x}_{j-1}, i_j) \right\}.$$

The SMM or VLMC will not reduce the computational complexity much, but help us to store the $d^m \times d^m$ transition matrix in a compact form. For SMM we don't have to compute $C(\mathbf{x}; \epsilon, 1)$ for each $\mathbf{x} \in \Sigma^m$, rather one representative from each group will do the job. Once we reach a state during constructing the prediction set, we look at the group in which the last m -tuple falls in, and take the corresponding $C(\mathbf{x}; \epsilon, 1)$ into our computing.

The Theorem 6.2.2 tells us that the coverage of the prediction set $C(i; \alpha, h)$ is at least $1-\alpha$. If $\epsilon-\delta$ is larger, the coverage increases, although the proportion of elements in the set still goes to 0 exponentially. However, the cardinality of the prediction set increases exponentially with h . The obvious question is can we get a sharper prediction set? For an irreducible and aperiodic chain, the cardinality of the prediction set must be exponentially increasing with h . It cannot be made geometric or sub-exponential. Suppose the transition matrix P has all the elements $p_{ij} > 0$ so that

$\zeta = \max p_{ij} < 1$. Then,

$$P(X_h = i_h, X_{h-1} = i_{h-1}, \dots, X_1 = i_1 | X_0 = i_0) = \prod_{j=1}^h p_{i_{j-1}, i_j} \leq \zeta^h.$$

Hence, for constructing a $100(1 - \alpha)\%$ prediction set, we need at least $\frac{1-\alpha}{\zeta^h}$ many elements in the prediction set, which is exponentially increasing with h . Even if $p_{ij} = 1$ for some $j \neq i$, the cardinality still remains exponential.

6.2.2 Anomaly Score Computation in SMM

To obtain the best possible prediction set in terms of number of elements, we may want to reduce our computation in an efficient way. Intuitively, it is obvious that in all d^h many possible h -tuples, there are groups of sequences which constitutes similar transitions among the sets. To illustrate this in detail, we start with each row of P , and sort them so that $p_{i,r_1} \geq p_{i,r_2} \geq \dots \geq p_{i,r_d}$ and denote $p_{i,r_j} = p_{i,(j)}$. For a sequence of alphabets (i_0, i_1, \dots, i_h) , compute the number

$$n_j = \sum_{k=1}^h \mathcal{I}(p_{i_{k-1}, i_k} = p_{i_{k-1}, (j)}),$$

i.e. the number of transitions which are j^{th} most likely. Clearly $n_1 + \dots + n_d = h$. Define $f(i_h, \dots, i_1 | i_0) = (n_1, \dots, n_d)$. If the ordered transition probabilities in each row are identical, i.e. $p_{i,(j)} = p_{i',(j)} := p_{(j)}$ for all $i, i' = 1, 2, \dots, d$ and $j = 1, \dots, d$, then

$$P(X_h = i_h, X_{h-1} = i_{h-1}, \dots, X_1 = i_1 | X_0 = i_0) = \prod_{j=1}^d p_{(j)}^{n_j}.$$

This condition is very restrictive, we will not get such structure of P in almost all case. However, if the variation of the ordered rows is not much, the probability of occurrence of a particular h tuple with $f(i_h, \dots, i_1 | i_0) = (n_1, \dots, n_d)$ can be approximated by the following quantity

$$s(n_1, \dots, n_d) = \prod_{j=1}^d \bar{p}_{(j)}^{n_j}$$

where $\bar{p}_{(j)} = \frac{1}{d} \sum_{i=1}^d p_{i,(j)}$. Clearly, we include only those h tuples in our prediction set for which $s(n_1, \dots, n_d) > k_0$ for some constant k_0 , and

$$\sum_{(n_1, \dots, n_d): s(n_1, \dots, n_d) > k_0} \frac{h!}{n_1! \dots n_d!} s(n_1, \dots, n_d) \geq 1 - \alpha.$$

In other words, we include those h tuples for which

$$Z = \sum_{j=1}^d n_j \log(\bar{p}_{(j)}) > k_1$$

for some constant k_1 . This threshold can be determined in the following way. Note that $\frac{h!}{n_1! \dots n_d!} s(n_1, \dots, n_d)$ is the occurrence probability of an observation from Multinomial distribution with parameters $(h, \bar{\mathbf{p}})$, i.e. $(n_1, \dots, n_d) \sim \text{Multinomial}(h, \bar{\mathbf{p}})$. Using the CLT for multinomial distribution, the $100(1 - \alpha)\%$ prediction set consists of all those h tuples for which

$$Z > h \sum_{j=1}^d \bar{p}_{(j)} \log(\bar{p}_{(j)}) - z_\alpha \sqrt{h \mathbf{a}^T V \mathbf{a}},$$

where $\mathbf{a}^T = (\log(\bar{p}_{(1)}), \dots, \log(\bar{p}_{(d)}))$, and $V = \text{diag}(\bar{\mathbf{p}}) - \bar{\mathbf{p}}\bar{\mathbf{p}}^T$. Let $C(i_0, \alpha)$ be the cardinality of this set. Then using similar arguments as before,

$$\frac{C(i_0, \alpha)}{d^h} \approx \Phi\left(-\sqrt{h}\beta + z_\alpha \sqrt{\frac{h \mathbf{a}^T V \mathbf{a}}{h \mathbf{a}^T V_1 \mathbf{a}}}\right)$$

where $\beta = \sum_{j=1}^d \bar{p}_{(j)} \log(\bar{p}_{(j)}) - \frac{1}{d} \sum_{j=1}^d \log(\bar{p}_{(j)}) > 0$, and $V_1 = \frac{1}{d} \mathbf{I} - \frac{1}{d^2} \mathbf{1}\mathbf{1}^T$. Thus the ratio goes to 0 exponentially as h increases.

The benefit of this approach is that we only need to compute $\binom{h+d-1}{d-1} \sim h^{d-1}$ many probabilities for complete enumeration. For small d , this will give us dividend in terms of accuracy. If d is large, we can distribute all these d probabilities in fixed number of blocks in each row, and compute occurrences of transitions in a particular block. Same algorithm holds for any sparse Markov models as well, where we only need the ordered transition probabilities for each partition. Note that, we are not going to report the h -tuples explicitly, rather characterize the h -tuples in terms of (n_1, \dots, n_d)

and compute the later. The next algorithm will demonstrate explicitly how our method works in constructing the prediction sets.

Algorithm 12 Score Based Prediction Set in SMM

Input: Observed sequence $\{X_1, \dots, X_n\}$; Order m ; State Space Σ ; $|\Sigma| = d$; Future step h ;
 Partition $\{\mathcal{C}_1, \dots, \mathcal{C}_g\}$ of Σ^m ; \mathbf{R}_i is the transition probability vector of \mathcal{C}_i ; prediction set level α ;
 Set $A = \{(n_1, \dots, n_d) : \sum_{j=1}^d n_j = h, n_j \in \mathbb{N} \cup \{0\}\}$.

- 1: **for** $i = 1, 2, 3, \dots, g$ **do**
- 2: Sort elements of \mathbf{R}_i : $R_{i,(1)} \geq \dots \geq R_{i,(d)}$
- 3: **end for**
- 4: **for** $j = 1, 2, \dots, d$ **do**
- 5: $\bar{R}_{(j)} = \frac{1}{d^m} \sum_{i=1}^g |\mathcal{C}_i| R_{i,(j)}$
- 6: **end for**
- 7: $V = \text{diag}(\bar{R}) - \bar{R}\bar{R}^T$;
- 8: $thres = \sum_{j=1}^d \bar{R}_{(j)} \log(\bar{R}_{(j)}) - z_\alpha \sqrt{\bar{R}^T V \bar{R} / h}$
- 9: $Pred_{Set} = \{\emptyset\}$
- 10: **for** $(n_1, \dots, n_d) \in A$ **do**;
- 11: $B_{n_1, \dots, n_d} = \{(i_1, \dots, i_h) : n_j = \sum_{t=1}^h \mathcal{I}(i_t \text{ is the } j^{th} \text{ likely state given the past histories upto } i_{t-1})\}$;
- 12: $score = \sum_{j=1}^d n_j \log(\bar{R}_{(j)})$
- 13: **if** $score > thres$ **then**
- 14: $Pred_{Set} = Pred_{Set} \cup B_{n_1, \dots, n_d}$;
- 15: **end if**
- 16: **end for**

Output: $Pred_{Set}$.

Most importantly, the score associated with each h -tuple will help us determine whether that particular h -tuple belongs to the $100(1 - \alpha)\%$ prediction set or not. This feature will help us in many scenarios where we have to find anomalies in a fairly long sequences. Hence from practical perspective, this score based method will be immensely important.

6.3 Point Prediction by Bootstrap

In this section, we will discuss a bootstrap based method for h -step future prediction, given the observed sequence. In words, for an SMM of order m , we want to estimate the h -step transition probability $P(X_{n+h} | X_n, \dots, X_{n-m+1})$. Usually, this probability distribution can be computed by sum-

ming out the probabilities of all possible intermediate values $X_{n+1}, \dots, X_{n+h-1}$, which would require computation of a mammoth d^h many probabilities, making our objective infeasible.

To address this issue computationally, we use the bootstrap method to approximate the transition probability $P(X_{n+h}|X_n \dots, X_{n-m+1})$. Given the sequence X_1, \dots, X_n , we first fit an SMM of order m with suitable method, and get the estimated transition probabilities for all possible m -tuples. Next we use the last part of the chain (X_{n-m+1}, \dots, X_n) as the initial values and iterations of estimated SMM/VLMC transition probabilities will be used to generate bootstrap replicates of the X_{n+1}, \dots, X_{n+h} . In this step, we need a large number of bootstrap samples, say B which will give us B independent copies of X_{n+1}, \dots, X_{n+h} . We denote these samples by

$$\Sigma_{n,h}^j \equiv \{X_{n+1}^{*j}, \dots, X_{n+h}^{*j}\}, \quad \text{for } j = 1, 2, \dots, B.$$

The empirical frequencies for different states in $\Sigma_{n,h}^j$ will be used to predict X_{n+h} , and possibly find the maximal probability prediction set for X_{n+h} as well. Bootstrap samples will be really useful in the sense that it will bypass the computation of the probabilities of all possible intermediate values, and are likely to correspond to the paths with higher occurrence probability. Hence, We demonstrate the method using simulation study.

6.3.1 Simulation Set-up

In our simulation, we set the sample space $\Sigma = \{1, 2, \dots, 10\}$, and order of the chain $m = 6$. We divide these 10^6 many possible histories in 100 groups, each containing 10000 many 6-tuples. For the i^{th} group, the transition probabilities are generated from $Dirichlet(e^{Z_{i,1}}, \dots, e^{Z_{i,d}})$, where $Z_{i,1}, \dots, Z_{i,d}$ are generated from i.i.d. $U(0, 1)$ distribution. We generate a chain $\{X_1, \dots, X_n\}$ of length $n = 100000$ following this SMM parameters.

Given the observed sequence, we predict X_{n+h} for h ranging from 1 to 10. In other words we estimate the h -step transition probabilities using the B many bootstrap samples. For generating the bootstrap samples, we use the true transition probabilities for generating future observations as well. This is because, even if we generate the future samples from the fitted SMM, it would give

us the prediction corresponding to that fitted SMM. From the predicted observations, we can only justify whether there is a huge deviation in terms of the h -step estimated transition probabilities from sample to sample. The estimated transition probability and the predictor is computed using the following formula

$$\hat{P}(X_{n+h} = i | X_n, \dots, X_{n-m+1}) = \frac{1}{B} \sum_{j=1}^B \mathcal{I}(X_{n+h}^{*j} = i)$$

$$\hat{X}_{n+h} = \arg \max_{i \in \Sigma} \hat{P}(X_{n+h} = i | X_n, \dots, X_{n-m+1})$$

Note that, there are intrinsic variability among the bootstrap samples themselves, and hence this estimated probabilities and \hat{X}_{n+h} are random quantities too. Depending on the generated B bootstrap samples, these estimates could vary. To measure this variability, we repeat the process 100 times, which means in every replication we generate different sets of B many bootstrap samples. We report the average value of the transition probabilities obtained from these 100 replications, along with their standard errors. As the number of bootstrap sample B , we take different values, ranging from $B = 10000$ to $B = 500000$, and compare the performances in terms of the standard errors as B changes.

We present the results in the following section.

6.3.2 Results

In table (6.1), we report the summary of the estimates of X_{n+h} for different values of h and B . We first report the state which turned out to be the estimator of X_{n+h} maximum times among those 100 replications, and in the bracket we report that proportion.

It is clear from the table that as B increases, the proportion gets closer to 1, which implies for sufficiently large number of bootstrap samples, the most likely observation X_{n+h} can be predicted accurately. These predictors obviously depends on the accuracy of the estimated transition probabilities as well. If the top two or more transition probabilities for some h are very close to each other, then slight variations will change the predictor. From the tables (6.2)-(6.5), we get an idea how these h -step transition probabilities change as B varies.

Table 6.1: Summary of the Predicted Sets.

Most predicted states for each h (Proportion of replications predicting the most predicted state)										
	h=1	h=2	h=3	h=4	h=5	h=6	h=7	h=8	h=9	h=10
B=10000	1 (1)	1 (0.61)	1 (1)	1 (0.49)	1 (0.53)	1 (0.45)	5 (0.4)	1 (0.4)	1 (0.47)	1 (0.45)
B=25000	1 (1)	1 (0.84)	1 (0.99)	5 (0.52)	1 (0.68)	1 (0.58)	1 (0.56)	1 (0.55)	1 (0.54)	1 (0.54)
B=50000	1 (1)	1 (0.88)	1 (1)	5 (0.57)	1 (0.7)	1 (0.52)	1 (0.64)	1 (0.55)	1 (0.63)	1 (0.59)
B=100000	1 (1)	1 (0.95)	1 (1)	5 (0.62)	1 (0.93)	1 (0.69)	1 (0.8)	1 (0.71)	1 (0.73)	1 (0.67)
B=200000	1 (1)	1 (1)	1 (1)	5 (0.66)	1 (0.92)	1 (0.79)	1 (0.89)	1 (0.85)	1 (0.89)	1 (0.74)
B=500000	1 (1)	1 (1)	1 (1)	5 (0.8)	1 (0.99)	1 (0.98)	1 (0.98)	1 (0.98)	1 (0.96)	1 (0.94)

Table 6.2: Bootstrap Estimates of h -step Transition Probabilities with $B = 10000$.

Bootstrap Estimate of h step Transition Probabilities										
	State 1	State 2	State 3	State 4	State 5	State 6	State 7	State 8	State 9	State 10
h=1	0.198	0.0871	0.019	0.118	0.0399	0.0085	0.1338	0.1494	0.1459	0.1004
h=2	0.1303	0.1139	0.105	0.067	0.0863	0.0408	0.0876	0.1164	0.1272	0.1254
h=3	0.1222	0.1102	0.1064	0.0933	0.1011	0.0848	0.0893	0.0871	0.1049	0.1006
h=4	0.11	0.1015	0.101	0.1017	0.1099	0.0964	0.0833	0.098	0.0948	0.1033
h=5	0.1101	0.1083	0.0964	0.1006	0.1072	0.0889	0.0837	0.0988	0.1021	0.1038
h=6	0.1099	0.1091	0.096	0.1015	0.1087	0.0882	0.0851	0.0968	0.1017	0.1031
h=7	0.1095	0.108	0.0966	0.1019	0.1089	0.0887	0.0847	0.0975	0.1015	0.1027
h=8	0.1097	0.1083	0.0957	0.1017	0.1088	0.0884	0.0853	0.0973	0.1021	0.1027
h=9	0.1096	0.1084	0.096	0.1024	0.1079	0.0881	0.0852	0.0972	0.102	0.1031
h=10	0.1096	0.1084	0.0966	0.1014	0.1087	0.0883	0.0849	0.0969	0.1017	0.1034
Bootstrap Standard Error of Transition Probabilities										
	State 1	State 2	State 3	State 4	State 5	State 6	State 7	State 8	State 9	State 10
h=1	0.0037	0.0031	0.0013	0.0032	0.002	0.0009	0.0036	0.0035	0.0037	0.0032
h=2	0.0033	0.0031	0.0031	0.0026	0.0028	0.002	0.0025	0.0033	0.0037	0.0034
h=3	0.003	0.0033	0.0031	0.0032	0.0027	0.0029	0.0028	0.0028	0.0032	0.0029
h=4	0.003	0.0032	0.0027	0.003	0.003	0.0033	0.0027	0.003	0.0033	0.0029
h=5	0.0032	0.0032	0.0032	0.0032	0.0033	0.0029	0.0029	0.003	0.0029	0.0025
h=6	0.0033	0.003	0.0025	0.0032	0.0031	0.0029	0.0028	0.0031	0.003	0.0029
h=7	0.0033	0.003	0.0033	0.0029	0.0028	0.0032	0.0029	0.003	0.0028	0.0029
h=8	0.0034	0.0029	0.0031	0.0031	0.0033	0.0028	0.0028	0.0032	0.0031	0.0032
h=9	0.0033	0.0028	0.003	0.0026	0.0031	0.0027	0.0028	0.0026	0.0033	0.0028
h=10	0.003	0.0029	0.0028	0.0031	0.0032	0.0029	0.0027	0.0028	0.003	0.0029

Table 6.3: Bootstrap Estimates of h -step Transition Probabilities with $B = 25000$.

Bootstrap Estimate of h step Transition Probabilities										
	State 1	State 2	State 3	State 4	State 5	State 6	State 7	State 8	State 9	State 10
h=1	0.1983	0.087	0.0191	0.1183	0.0399	0.0084	0.1336	0.1496	0.1447	0.1011
h=2	0.1302	0.1141	0.1047	0.0668	0.0865	0.0408	0.087	0.117	0.1276	0.1254
h=3	0.1222	0.1109	0.1063	0.0931	0.1008	0.0845	0.0892	0.0871	0.1052	0.1006
h=4	0.11	0.1021	0.1009	0.1017	0.1104	0.0964	0.0831	0.0977	0.0949	0.1028
h=5	0.1106	0.1085	0.0963	0.1004	0.1077	0.0886	0.0836	0.0985	0.1019	0.104
h=6	0.1101	0.1089	0.0965	0.1014	0.1083	0.0883	0.0851	0.0965	0.1019	0.1029
h=7	0.1099	0.1082	0.0964	0.1018	0.1086	0.0885	0.0848	0.0971	0.1021	0.1025
h=8	0.1096	0.1083	0.0964	0.1017	0.1085	0.0882	0.085	0.0975	0.102	0.1029
h=9	0.1099	0.1084	0.0964	0.102	0.1082	0.0885	0.0847	0.0967	0.102	0.1033
h=10	0.1097	0.1087	0.0965	0.102	0.1084	0.0881	0.0849	0.0965	0.1022	0.103
Bootstrap Standard Error of Transition Probabilities										
	State 1	State 2	State 3	State 4	State 5	State 6	State 7	State 8	State 9	State 10
h=1	0.0025	0.0019	9.00E-04	0.0021	0.0012	6.00E-04	0.0022	0.0021	0.0021	0.0019
h=2	0.0022	0.0019	0.002	0.0017	0.0017	0.0013	0.0017	0.0018	0.0024	0.0017
h=3	0.0019	0.0021	0.002	0.0017	0.0018	0.0019	0.0019	0.0019	0.0019	0.0017
h=4	0.0018	0.0017	0.002	0.002	0.0018	0.0018	0.0015	0.0018	0.002	0.0021
h=5	0.0023	0.002	0.0018	0.0018	0.002	0.002	0.0016	0.0019	0.0018	0.0017
h=6	0.002	0.002	0.0018	0.002	0.0019	0.0018	0.0016	0.0019	0.002	0.0019
h=7	0.0019	0.0017	0.002	0.0016	0.0021	0.0019	0.0016	0.0018	0.0015	0.0019
h=8	0.0019	0.0017	0.0015	0.0018	0.0019	0.0018	0.0019	0.0018	0.0019	0.0019
h=9	0.002	0.002	0.0018	0.0021	0.002	0.0018	0.0016	0.0019	0.0017	0.0019
h=10	0.0021	0.0019	0.002	0.0022	0.0021	0.0018	0.0017	0.002	0.0019	0.0015

Table 6.4: Bootstrap Estimates of h -step Transition Probabilities with $B = 100000$.

Bootstrap Estimate of h step Transition Probabilities										
	State 1	State 2	State 3	State 4	State 5	State 6	State 7	State 8	State 9	State 10
h=1	0.1983	0.087	0.0188	0.1181	0.0401	0.0084	0.1338	0.1497	0.1449	0.1008
h=2	0.13	0.1141	0.1048	0.067	0.0864	0.0407	0.0874	0.1168	0.1273	0.1254
h=3	0.122	0.1103	0.1064	0.093	0.101	0.0845	0.0895	0.0873	0.105	0.1008
h=4	0.1098	0.102	0.1011	0.102	0.1102	0.0965	0.0833	0.098	0.0947	0.1024
h=5	0.1104	0.1085	0.0963	0.1	0.1074	0.0889	0.0836	0.0986	0.1023	0.104
h=6	0.1099	0.1091	0.0963	0.1016	0.1081	0.0884	0.0852	0.0965	0.102	0.1028
h=7	0.1101	0.1082	0.0965	0.102	0.1085	0.0884	0.0849	0.0969	0.1017	0.1028
h=8	0.1096	0.1084	0.0964	0.102	0.1084	0.0882	0.0848	0.0973	0.102	0.1029
h=9	0.1098	0.1084	0.0962	0.102	0.1083	0.0883	0.0849	0.0968	0.102	0.1033
h=10	0.1097	0.1084	0.0964	0.1021	0.1083	0.0883	0.0848	0.0969	0.1019	0.1032
Bootstrap Standard Error of Transition Probabilities										
	State 1	State 2	State 3	State 4	State 5	State 6	State 7	State 8	State 9	State 10
h=1	0.0013	0.0008	0.0004	0.0011	0.0006	0.0003	0.0011	0.001	0.0011	0.001
h=2	0.001	0.0009	0.0009	0.0007	0.0009	0.0007	0.0008	0.001	0.0011	0.0011
h=3	0.0011	0.0009	0.001	0.001	0.0011	0.0008	0.0009	0.0008	0.001	0.001
h=4	0.001	0.0009	0.001	0.0009	0.0011	0.0009	0.0009	0.0009	0.0009	0.0011
h=5	0.001	0.0009	0.0009	0.001	0.0009	0.0009	0.0009	0.0009	0.0009	0.001
h=6	0.001	0.0009	0.0008	0.0009	0.0009	0.0009	0.0008	0.001	0.0009	0.0008
h=7	0.001	0.001	0.0009	0.0009	0.001	0.0008	0.0009	0.0008	0.0009	0.001
h=8	0.001	0.001	0.001	0.0011	0.001	0.0009	0.0009	0.0011	0.0009	0.0009
h=9	0.001	0.001	0.0009	0.001	0.0009	0.0008	0.0008	0.001	0.0008	0.0009
h=10	0.0011	0.001	0.0008	0.001	0.001	0.001	0.0008	0.0009	0.0009	0.0009

Table 6.5: Bootstrap Estimates of h -step Transition Probabilities with $B = 500000$.

Bootstrap Estimate of h step Transition Probabilities										
	State 1	State 2	State 3	State 4	State 5	State 6	State 7	State 8	State 9	State 10
h=1	0.1983	0.087	0.0189	0.1181	0.04	0.0085	0.1339	0.1495	0.1448	0.1009
h=2	0.1299	0.1141	0.105	0.067	0.0864	0.0406	0.0873	0.1168	0.1274	0.1255
h=3	0.1219	0.1104	0.1067	0.093	0.1009	0.0845	0.0894	0.0873	0.105	0.1008
h=4	0.1098	0.1021	0.1011	0.1018	0.1104	0.0964	0.0832	0.0979	0.0947	0.1026
h=5	0.1103	0.1086	0.0962	0.1002	0.1076	0.0888	0.0836	0.0985	0.1022	0.1039
h=6	0.11	0.1089	0.0962	0.1016	0.1083	0.0885	0.0852	0.0966	0.102	0.1028
h=7	0.11	0.108	0.0964	0.1022	0.1084	0.0885	0.0848	0.097	0.1018	0.1028
h=8	0.1097	0.1084	0.0963	0.1022	0.1083	0.0882	0.0847	0.0971	0.1021	0.1029
h=9	0.1097	0.1084	0.0964	0.1021	0.1083	0.0883	0.0848	0.0969	0.102	0.1031
h=10	0.1096	0.1084	0.0964	0.1021	0.1084	0.0883	0.0848	0.0968	0.102	0.1031
Bootstrap Standard Error of Transition Probabilities										
	State 1	State 2	State 3	State 4	State 5	State 6	State 7	State 8	State 9	State 10
h=1	0.0005	0.0004	0.0002	0.0004	0.0003	0.0001	0.0005	0.0005	0.0004	0.0004
h=2	0.0004	0.0005	0.0004	0.0004	0.0004	0.0003	0.0004	0.0005	0.0005	0.0005
h=3	0.0005	0.0004	0.0004	0.0004	0.0004	0.0004	0.0004	0.0004	0.0004	0.0004
h=4	0.0004	0.0005	0.0004	0.0004	0.0005	0.0004	0.0004	0.0004	0.0005	0.0004
h=5	0.0005	0.0004	0.0005	0.0004	0.0005	0.0004	0.0004	0.0004	0.0005	0.0004
h=6	0.0004	0.0004	0.0005	0.0004	0.0005	0.0004	0.0004	0.0004	0.0005	0.0004
h=7	0.0004	0.0005	0.0004	0.0005	0.0005	0.0004	0.0004	0.0004	0.0004	0.0004
h=8	0.0004	0.0004	0.0004	0.0004	0.0004	0.0004	0.0004	0.0004	0.0004	0.0004
h=9	0.0005	0.0005	0.0004	0.0004	0.0004	0.0004	0.0004	0.0004	0.0004	0.0004
h=10	0.0005	0.0004	0.0005	0.0004	0.0004	0.0004	0.0003	0.0004	0.0004	0.0004

In these tables, we first report the mean of the transition probabilities for all 100 replications, giving us and their standard error for different values of B . We see that for $h = 1$, transition probabilities are almost same even if B is 10000. The reason behind is that for $h = 1$, we are actually predicting X_{n+1} . This is equivalent to find the one-step transition probabilities corresponding to the SMM, and verifying the consistency of the estimated transition probabilities. The major observation from this simulation is that for some specific values of h , there is a very little difference between the highest element the second highest in the estimated transition probability vector. For example, for $h = 4$ and $B = 500,000$, the top two probabilities correspond to state 5 and state 1 respectively, with probabilities 0.1102 and 0.1098, which are same if we round upto two decimals. Hence, even with $B = 500,000$, there are few occasions when the predictor of X_{n+4} turns out to be the state 1. However, when the probabilities are separated a little bit, the bootstrap estimates of the probabilities give us the most likely predictor. The standard errors of these estimates are very low as well, and gets lower with increasing B . This says, there is no random fluctuation in the bootstrap estimates, and they are most likely to produce consistent estimators.

Another observation from the study is that with increasing h , the transition probability vectors tend to get similar. If we look into the last three rows of each table, i.e. rows corresponding to $h = 8, 9$ and 10, we see that there are very little to none differences between the transition probability vectors. This phenomenon can be justified with the stationarity argument. If h increases, the transition probabilities $P(X_{n+h}|X_n, \dots, X_{n-m+1})$ converges to the stationary probability distribution for the chain. In this example, we can see for $h \geq 8$, the probabilities are close to the stationary probability. This means, for large enough h , we don't need to perform the bootstrap sampling for predicting X_{n+h} , instead we estimate the stationary probabilities for each state from X_1, \dots, X_n by their empirical frequencies, and get the most likely state in long run.

6.4 Real Data Application: Anomaly Detection in Helicobacter Pylori Bacteria

Helicobacter Pylori is a gram-negative, microaerophilic and spiral-shaped bacteria which can be found in the stomach. The helical shape of the bacteria is developed in order to penetrate the mucoid lining in the stomach and causes infection. It is believed that nearly 50% of the humans have been exposed with this bacteria, however only 10–12% exhibit clinical symptoms including gastritis, upset stomach, carcinoma or peptic ulcer.

There are many different strains of this bacteria, among which the “Type I” strain is associated with more serious infection. There are many further strains of this type, “**Strain 26695**” is one of them. The full genome sequence of this strain has been developed by Tomb et al. (1997), and the data is collected from the NCBI database. It has a circular genome of 1,667,867 base pairs and 1590 coding sequences. Studies have confirmed that more infectious HP bacteria differ from the less infectious by a region of genes, called the CAG pathogenity island, or CAG region. There are about 26 genes in this region which mutate internally, resulting in different levels of infections. Liu et al. (1999) identified there is a sequence anomaly in the CAG7 protein in this gene. When CAG7 regions of different strains are compared, there are significant unusual patterns can be seen in the strain **26695**. Liu et al. (1999) have conducted rigorous statistical analysis which underscore many sequential features of this protein, and identified the potential anomalies. In this section, we will use the idea of SMM in detecting the anomaly of this CAG7 gene in this **26695** strain. The structure of this CAG region is mostly homogeneous, unless there are some known anomalies. Therefore, we use the remaining portion of the CAG region except CAG7 protein as our training sequence, and the CAG7 protein as the test sequence.

6.4.1 Method

We first fit an SMM of order $m = 3$ from the training RNA sequence, having length of about 28000. For modelling purpose, we have used the convex clustering methodology adapted by Majumder et al. (2022) for fitting sparse Markov models, which partitions the set of all possible histories in

Σ^m , based on the empirical transition probability vectors which are close to each other. In this example, $\Sigma = \{A, G, T, C\}$, hence $|\Sigma| = 4$. Define the total number of m -tuples as $p = |\Sigma|^m$. If $\hat{\pi}_i$ is the empirical transition probability of the i^{th} m -tuple, then Majumder et al. (2022) proposed to minimize a penalized criterion function, derived from Chi and Lange (2015), as follows:

$$\min_{\mathbf{b}_1, \dots, \mathbf{b}_p} \frac{1}{2} \sum_{j=1}^p \|\hat{\pi}_j - \mathbf{b}_j\|_2^2 + \lambda \sum_{1 \leq i < j \leq p} w_{i,j} \|\mathbf{b}_i - \mathbf{b}_j\|_2 \quad (6.4.7)$$

where $\lambda > 0$ is a penalty parameter, $w_{i,j}$ are suitable non-negative weights and $\mathbf{b}_j = (b_{j,1}, \dots, b_{j,d})^T \in \Pi_d$ for $j = 1, \dots, p$, where Π_d is the d -dimensional simplex $\Pi_d = \{(u_1, \dots, u_d) \in [0, 1]^d : u_1 + \dots + u_d = 1\}$. Denote the optimum solution of (6.4.7) by \mathbf{b}_i^* , $i = 1, 2, \dots, p$. The penalization will ensure that depending on the value of the tuning parameter λ , $\mathbf{b}_i^* = \mathbf{b}_j^*$ if $\hat{\pi}_i$ and $\hat{\pi}_j$ are close to each other. Hence, the resulting solution will give us a partition of Σ^m . Denote the number of groups by k_λ and the estimated partitions are $\hat{\mathcal{C}}_1, \dots, \hat{\mathcal{C}}_{k_\lambda}$. For the i^{th} group, the elements of the common transition probability vector \mathbf{R}_i are computed by the pooled transitions from the elements of that group to the four states. The tuning parameter λ is selected using the BIC criterion. The weights are selected so that $w_{i,j} = \exp\left[-\frac{1}{\sqrt{t_i t_j}} \|\hat{\pi}_i - \hat{\pi}_j\|_\infty\right] \mathcal{I}(\hat{\pi}_i \text{ is one of the 5-nearest neighbor of } \hat{\pi}_j \text{ or vice versa})$, where t_i is median d_∞ distance of 5-nearest neighbors of $\hat{\pi}_i$.

Once we have the fitted model, we apply the algorithm (12) to compute the score of the test sequence, and determine whether this score falls below the threshold. We discuss our findings next.

6.4.2 Result

The fitted SMM has 13 groups, the group sizes are 13, 10, 9(twice), 7, 4(twice) and 1(five singleton groups). The length of the CAG7 gene in terms of base pairs is 5784. Prior studies by Liu et al. (1999) have confirmed that the anomaly appears in the later part of the genome sequence, determined based on the sequence patten. Hence, instead of taking the whole CAG7 gene as our test sequence, we use the last 25% of the sequence for detecting anomaly. We apply the algorithm (12) to determine the score of the test sequence. The estimated values of the ordered values $\bar{R}_{(1)}, \dots, \bar{R}_{(4)}$ are (0.398, 0.272, 0.201, 0.129). The last 3 base pairs just preceding the test sequence, which we call as

the past history is $past = \{T, G, C\}$. Hence, we compute the anomaly score of the test sequence x by $s(x|past)$ using the score formula in the algorithm (12).

It turns out that the anomaly score of the test sequence is -1.338 , while the threshold for 95% prediction set is -1.325 . This shows, the anomaly score value falls below the required threshold at 95% confidence level. It can be also computed that the estimated probability of the m -tuples whose anomaly score falls below the observed score is 0.001, which is a very low probability event. Hence, our study confirms the sequence anomaly in the CAG7 gene of the strain **26695** of *Helicobacter Pylori*.

This study has some further implications. Although we have checked a portion of the sequence for anomaly detection, which is known to have potential anomalies, we can certainly use this technique in other RNA sequences whose anomalies are not known. For example, a particular strain of a disease can exhibit different severity levels compared to the other standard strains. In that case, one reference genome sequence can be modelled using SMM, and the different parts of the test sequence can be scanned to find anomalous regions of certain lengths. Compared to the traditional methods of sequential matching or computing pattern statistics, this probabilistic method will be more useful in terms of quantifying the likelihood of a particular sequence to occur or not.

6.5 Conclusion

Overall, this paper gives some new direction of developing a prediction algorithm which will be an useful tool constructing prediction sets in sparse Markov model. The bootstrap method is quite impressive, where we can directly simulate the future observations from the fitted SMM and get the estimates of the h -step transition probabilities. On the other hand, the score based method is proven to be useful in constructing simultaneous prediction sets in a compact manner. The anomaly detection problem justifies its practical applicability in sequential data. This broadens up the scope for further research in different scientific problems, not limited to DNA or RNA sequence analysis, data compression, text prediction and other practical aspects.

REFERENCES

- Banert, S., Bot, R. I., and Csetnek, E. R. (2016). Fixing and extending some recent results on the admm algorithm. *arXiv preprint arXiv:1612.05057*.
- Basharin, G. P., Langville, A. N., and Naumov, V. A. (2004). The life and work of AA Markov. *Linear algebra and its applications*, 386:3–26.
- Basu, A., Harris, I. R., Hjort, N. L., and Jones, M. (1998). Robust and efficient estimation by minimising a density power divergence. *Biometrika*, 85(3):549–559.
- Begleiter, R., El-Yaniv, R., and Yona, G. (2004). On prediction using variable order Markov models. *Journal of Artificial Intelligence Research*, 22:385–421.
- Belloni, A. and Oliveira, R. I. (2017). Approximate group context tree. *The Annals of Statistics*, 45(1):355–385.
- Bennett, I., Martin, D. E., and Lahiri, S. N. (2022). Fitting sparse Markov models through a collapsed gibbs sampler. *Submitted for publication*.
- Besag, J. (1974). Spatial interaction and the statistical analysis of lattice systems. *Journal of the Royal Statistical Society: Series B (Methodological)*, 36(2):192–225.
- Besag, J. (1978). Some methods of statistical analysis for spatial data. *Bull. Int. Statist. Inst.*, 47(2):77–92.
- Billingsley, P. (1961). Statistical methods in Markov chains. *The Annals of Mathematical Statistics*, pages 12–40.
- Boyd, S., Parikh, N., Chu, E., Peleato, B., Eckstein, J., et al. (2011). Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends® in Machine learning*, 3(1):1–122.
- Bradley, R. C. (1986). Basic properties of strong mixing conditions. In *Dependence in probability and statistics*, pages 165–192. Springer.
- Braun, J. V. and Muller, H.-G. (1998). Statistical methods for dna sequence segmentation. *Statistical Science*, pages 142–162.
- Bühlmann, P. (2000). Model selection for variable length Markov chains and tuning the context algorithm. *Annals of the Institute of Statistical Mathematics*, 52(2):287–315.
- Bühlmann, P., Wyner, A. J., et al. (1999). Variable length Markov chains. *The Annals of Statistics*, 27(2):480–513.
- Cénac, P., Chauvin, B., Paccaut, F., and Pouyanne, N. (2018). Characterization of stationary probability measures for variable length Markov chains. *arXiv preprint arXiv:1807.01075*.
- Chan, K. S. and Geyer, C. J. (1994). Discussion: Markov chains for exploring posterior distributions. *Ann. Statist.*, 22(4):1747–1758.

- Chi, E. C. and Lange, K. (2015). Splitting methods for convex clustering. *Journal of Computational and Graphical Statistics*, 24(4):994–1013.
- Chi, E. C. and Steinerberger, S. (2019). Recovering trees with convex clustering. *SIAM Journal on Mathematics of Data Science*, 1(3):383–407.
- Cogburn, R. (1960). *Asymptotic properties of stationary sequences*. University of California Press.
- De Bruijn, N. G. (1981). *Asymptotic methods in analysis*, volume 4. Courier Corporation.
- Deng, W. and Yin, W. (2016). On the global and linear convergence of the generalized alternating direction method of multipliers. *Journal of Scientific Computing*, 66(3):889–916.
- Denker, M., Goldie, C., and Morrow, G. (1986). Uniform integrability and the central limit theorem for strongly mixing processes. In *Dependence in probability and statistics*, pages 269–289. Springer.
- Doebelin, W. (1937). Sur les propriétés asymptotiques de mouvement régis par certains types de chaînes simples. *Bulletin mathématique de la Société roumaine des sciences*, 39(1):57–115.
- Doob, J. L. (1953). *Stochastic processes*, volume 101. New York Wiley.
- Douc, R., Fort, G., Moulines, E., Soulier, P., et al. (2004). Practical drift conditions for subgeometric rates of convergence. *The Annals of Applied Probability*, 14(3):1353–1377.
- Doukhan, P., Massart, P., and Rio, E. (1994). The functional central limit theorem for strongly mixing processes. In *Annales de l’IHP Probabilités et statistiques*, volume 30, pages 63–82.
- Durrett, R. (2008). *Probability models for DNA sequence evolution*. Springer Science & Business Media.
- Eckhard, L. (1996). Central limit theorems for sums of α -mixing random variables. *Stochastics: An International Journal of Probability and Stochastic Processes*, 59(3-4):241–258.
- Fort, G. and Moulines, E. (2000). V-subgeometric ergodicity for a hastings–metropolis algorithm. *Statistics & probability letters*, 49(4):401–410.
- Galves, A., Galves, C., Garcia, J. E., Garcia, N. L., and Leonardi, F. (2012). Context tree selection and linguistic rhythm retrieval from written texts. *The Annals of Applied Statistics*, 6(1):186–209.
- Garcia, J. E., González-López, V. A., de Holanda, R. S. B., and Geraldo, C. U.-B. (2011). Minimal Markov models. In *Fourth Workshop on Information Theoretic Methods in Science and Engineering*, page 25.
- Hanson, D. L. and Wright, F. T. (1971). A bound on tail probabilities for quadratic forms in independent random variables. *The Annals of Mathematical Statistics*, 42(3):1079–1083.
- Harris, T. (1955). Recurrent Markov processes. 2. In *Annals of Mathematical Statistics*, volume 26, pages 152–153.
- Hobert, J. P. and Geyer, C. J. (1998). Geometric ergodicity of gibbs and block gibbs samplers for a hierarchical random effects model. *Journal of Multivariate Analysis*, 67(2):414–430.

- Hocking, T. D., Joulin, A., Bach, F., and Vert, J.-P. (2011). Clusterpath an algorithm for clustering using convex fusion penalties. In *28th international conference on machine learning*, page 1.
- Ibragimov, I. (1975). Independent and stationary sequences of random variables. *Wolters, Noordhoff Pub.*
- Ibragimov, I. A. (1962). Some limit theorems for stationary processes. *Theory of Probability & Its Applications*, 7(4):349–382.
- Jääskinen, V., Xiong, J., Corander, J., and Koski, T. (2014). Sparse Markov chains for sequence data. *Scandinavian Journal of Statistics*, 41(3):639–655.
- Jarner, S. F., Roberts, G. O., et al. (2002). Polynomial convergence rates of Markov chains. *The Annals of Applied Probability*, 12(1):224–247.
- Jones, G. L. et al. (2004). On the Markov chain central limit theorem. *Probability surveys*, 1(299–320):5–1.
- Karlin, S. and Taylor, H. M. (1975). *A first course in stochastic processes*. Elsevier.
- Kendall, D. G. (1959). Unitary dilations of Markov transition operators, and the corresponding integral representations for transition-probability matrices. *Probability and statistics*, pages 139–161.
- Kimura, M. (1981). Estimation of evolutionary distances between homologous nucleotide sequences. *Proceedings of the National Academy of Sciences*, 78(1):454–458.
- Kontoyiannis, I., Mertzanis, L., Panotopoulou, A., Papageorgiou, I., and Skoularidou, M. (2020). Bayesian context trees: Modelling and exact inference for discrete time series. *arXiv preprint arXiv:2007.14900*.
- Lai, C. K. (1967). *Markov chains with stationary transition probabilities*. Springer-Verlag.
- Lindsten, F., Ohlsson, H., and Ljung, L. (2011). Just relax and come clustering! a convexification of k-means clustering. *Linköping University, Department of Electrical Engineering, Automatic Control*.
- Liu, G., McDaniel, T. K., Falkow, S., and Karlin, S. (1999). Sequence anomalies in the cag7 gene of the helicobacter pylori pathogenicity island. *Proceedings of the National Academy of Sciences*, 96(12):7011–7016.
- Majumder, T., Lahiri, S., and Martin, D. (2022). Fitting sparse Markov models to categorical time series using regularization. *arXiv preprint arXiv:2202.05485*.
- Markov, A. (1906). Extension of law of big numbers on variables, depending from each other. *Izvestiya Fiziko-Matematicheskogo Obschestva pri Kazanskom Universitete*, 2:135–156.
- Mengersen, K. L., Tweedie, R. L., et al. (1996). Rates of convergence of the hastings and metropolis algorithms. *The annals of Statistics*, 24(1):101–121.
- Meyn, S. P. and Tweedie, R. L. (2012). *Markov chains and stochastic stability*. Springer Science & Business Media.

- Nummelin, E. (2004). *General irreducible Markov chains and non-negative operators*, volume 83. Cambridge University Press.
- Nummelin, E. and Tuominen, P. (1982). Geometric ergodicity of Harris recurrent Markov chains with applications to renewal theory. *Stochastic Processes and Their Applications*, 12(2):187–202.
- Nummelin, E. and Tweedie, R. L. (1978). Geometric ergodicity and r -positivity for general Markov chains. *The Annals of Probability*, pages 404–420.
- Orey, S. (1971). *Lecture notes on limit theorems for Markov chain transition probabilities*. Van Nostrand Reinhold.
- Orey, S. et al. (1959). Recurrent Markov chains. *Pacific Journal of Mathematics*, 9(3):805–827.
- Panahi, A., Dubhashi, D., Johansson, F. D., and Bhattacharyya, C. (2017). Clustering by sum of norms: Stochastic incremental algorithm, convergence and cluster recovery. In *International conference on machine learning*, pages 2769–2777. PMLR.
- Papageorgiou, I. and Kontoyiannis, I. (2022). Posterior representations for Bayesian context trees: Sampling, estimation and convergence. *arXiv preprint arXiv:2202.02239*.
- Pelckmans, K., De Brabanter, J., Suykens, J. A., and De Moor, B. (2005). Convex clustering shrinkage. In *PASCAL Workshop on Statistics and Optimization of Clustering Workshop*.
- Rabiner, L. R. (1989). A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286.
- Raftery, A. E. (1985). A model for high-order Markov chains. *Journal of the Royal Statistical Society: Series B (Methodological)*, 47(3):528–539.
- Rissanen, J. (1983). A universal prior for integers and estimation by minimum description length. *The Annals of Statistics*, 11(2):416–431.
- Roberts, G., Rosenthal, J., et al. (1997). Geometric ergodicity and hybrid Markov chains. *Electronic Communications in Probability*, 2:13–25.
- Roberts, G. O. and Polson, N. G. (1994). On the geometric convergence of the Gibbs sampler. *Journal of the Royal Statistical Society: Series B (Methodological)*, 56(2):377–384.
- Roberts, G. O. and Tweedie, R. L. (1996). Geometric convergence and central limit theorems for multidimensional Hastings and Metropolis algorithms. *Biometrika*, 83(1):95–110.
- Roos, T. and Yu, B. (2009a). Estimating sparse models from multivariate discrete data via transformed lasso. In *2009 Information Theory and Applications Workshop*, pages 290–294. IEEE.
- Roos, T. and Yu, B. (2009b). Sparse Markov source estimation via transformed lasso. In *2009 IEEE Information Theory Workshop on Networking and Information Theory*, pages 241–245. IEEE.
- Rosenthal, J. S. (1995). Minorization conditions and convergence rates for Markov chain Monte Carlo. *Journal of the American Statistical Association*, 90(430):558–566.
- Ross, S. M. (1996). *Stochastic processes*, volume 2. Wiley.

- Rue, H. and Held, L. (2005). *Gaussian Markov random fields: theory and applications*. CRC press.
- Shefi, R. and Teboulle, M. (2014). Rate of convergence analysis of decomposition methods based on the proximal method of multipliers for convex minimization. *SIAM Journal on Optimization*, 24(1):269–297.
- Sun, D., Toh, K.-C., and Yuan, Y. (2021). Convex clustering: Model, theoretical guarantee and efficient algorithm. *J. Mach. Learn. Res.*, 22:9–1.
- Tierney, L. (1994). Markov chains for exploring posterior distributions. *the Annals of Statistics*, pages 1701–1728.
- Tomb, J.-F., White, O., Kerlavage, A. R., Clayton, R. A., Sutton, G. G., Fleischmann, R. D., Ketchum, K. A., Klenk, H. P., Gill, S., Dougherty, B. A., et al. (1997). The complete genome sequence of the gastric pathogen helicobacter pylori. *Nature*, 388(6642):539–547.
- Tseng, P. (1991). Applications of a splitting algorithm to decomposition in convex programming and variational inequalities. *SIAM Journal on Control and Optimization*, 29(1):119–138.
- Tuominen, P. and Tweedie, R. L. (1994). Subgeometric rates of convergence of f-ergodic Markov chains. *Advances in Applied Probability*, 26(3):775–798.
- Tweedie, R. L. (1974a). R-theory for Markov chains on a general state space i: solidarity properties and r-recurrent chains. *The Annals of Probability*, pages 840–864.
- Tweedie, R. L. (1974b). R-theory for Markov chains on a general state space ii: r-subinvariant measures for r-transient chains. *The Annals of Probability*, pages 865–878.
- Tweedie, R. L. (1975). Sufficient conditions for ergodicity and recurrence of Markov chains on a general state space. *Stochastic Processes and their Applications*, 3(4):385–403.
- Vere-Jones, D. (1962). Geometric ergodicity in denumerable Markov chains. *The Quarterly Journal of Mathematics*, 13(1):7–28.
- Wang, M. and Allen, G. I. (2021). Integrative generalized convex clustering optimization and feature selection for mixed multi-view data. *Journal of Machine Learning Research*, 22:1–73.
- Watson, G. S. (1996). Spectral decomposition of the covariance matrix of a multinomial. *Journal of the Royal Statistical Society: Series B (Methodological)*, 58(1):289–291.
- Withers, C. S. (1981). Central limit theorems for dependent variables. i. *Zeitschrift für Wahrscheinlichkeitstheorie und verwandte Gebiete*, 57(4):509–534.
- Wu, Z. and McGoogan, J. M. (2020). Characteristics of and important lessons from the coronavirus disease 2019 (covid-19) outbreak in china: summary of a report of 72 314 cases from the chinese center for disease control and prevention. *jama*, 323(13):1239–1242.
- Xiong, J., Jääskinen, V., Corander, J., et al. (2016). Recursive learning for sparse Markov models. *Bayesian analysis*, 11(1):247–263.
- Yuan, M. and Lin, Y. (2006). Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 68(1):49–67.

- Zhang, Y., Li, R., and Tsai, C.-L. (2010). Regularization parameter selections via generalized information criterion. *Journal of the American Statistical Association*, 105(489):312–323.
- Zheng, P., Askham, T., Brunton, S. L., Kutz, J. N., and Aravkin, A. Y. (2018). A unified framework for sparse relaxed regularized regression: Sr3. *IEEE Access*, 7:1404–1423.
- Zhu, C., Xu, H., Leng, C., and Yan, S. (2014). Convex optimization procedure for clustering: Theoretical revisit. *Advances in Neural Information Processing Systems*, 27.

APPENDICES

APPENDIX

A

PROOF OF THE THEOREMS IN CHAPTER

3

A.1 Proof of Theorem 3.2.1

From now on, we will denote ℓ_{n,α_n} as ℓ_n for simplicity. First observe that, $E_{\pi_n}(\ell_n) = (n - m_n + 2)\pi_n(\alpha_n) - 1$, i.e. $E_{\pi_n}(\frac{\ell_n}{n\pi_n(\alpha_n)}) \rightarrow 1$ as $n \rightarrow \infty$. Also, by assumption (iv), $Var_{\pi_n}(\frac{\ell_n}{n\pi_n(\alpha_n)}) \rightarrow 0$. Hence $\frac{\ell_n}{n\pi_n(\alpha_n)} \xrightarrow{P} 1$ as $n \rightarrow \infty$, eventually leading to $\ell_n \xrightarrow{P} \infty$.

For any function $f : \Sigma^{m_n} \rightarrow \mathbb{R}$, define

$$s_j^{(n)}(f) = \sum_{t=\sigma_{\alpha_n}(j)+1}^{\sigma_{\alpha_n}(j+1)} f(Y_t^{(n)}), \quad t = 0, 1, \dots, \ell_n - 1.$$

By strong Markov property, $s_j^{(n)}(g_n)$ are iid random variables with mean $E_{\alpha_n}[\sum_{j=1}^{\tau_{\alpha_n}} g_n(Y_j^{(n)})] = E_{\alpha_n}[s_0^{(n)}(g_n)]$ and variance $Var_{\alpha_n}[s_0^{(n)}(g_n)] = E_{\alpha_n}[(s_0^{(n)}(g_n))^2] - E_{\alpha_n}^2[s_0^{(n)}(g_n)]$. Define $\bar{g}_n(x) = g_n(x) -$

$E_{\pi_n}(g_n(X))$, where $X \sim \pi_n$. Note that $E_{\pi_n}[g_n] = \pi_n(\alpha_n)E_{\alpha_n}[\sum_{j=1}^{\tau_{\alpha_n}} g_n(Y_j^{(n)})]$. Hence

$$\begin{aligned} E_{\alpha_n}[s_j^{(n)}(\bar{g}_n)] &= E_{\alpha_n}\left[\sum_{j=1}^{\tau_{\alpha_n}} (g_n(Y_j^{(n)}) - E_{\pi_n}(g_n))\right] \\ &= E_{\alpha_n}\left[\sum_{j=1}^{\tau_{\alpha_n}} g_n(Y_j^{(n)}) - \tau_{\alpha_n} E_{\pi_n}(g_n)\right] \\ &= E_{\alpha_n}\left[\sum_{j=1}^{\tau_{\alpha_n}} g_n(Y_j^{(n)})\right] - E_{\alpha_n}[\tau_{\alpha_n}]E_{\pi_n}(g_n) \\ &= 0, \quad \text{since } \pi_n(\alpha_n) = 1/E_{\alpha_n}[\tau_{\alpha_n}]. \end{aligned}$$

The normalized and centralized random variables are defined as follows:

$$s_j^{(n)}(\bar{g}_n) = \frac{s_j^{(n)}(\bar{g}_n)}{\sqrt{\text{Var}_{\alpha_n}[s_0^{(n)}(\bar{g}_n)]}} = \frac{s_j^{(n)}(\bar{g}_n)}{\sqrt{E_{\alpha_n}[s_0^{(n)}(\bar{g}_n)]^2}}.$$

Thus, $s_j^{(n)}(\bar{g}_n)$, $j = 0, 1, \dots, \ell_n - 1$ are i.i.d random variables with mean 0 and variance 1. Denote $\nu_n^2 = E_{\alpha_n}[s_0^{(n)}(\bar{g}_n)]^2$. Fix $0 < \epsilon < 1$. Define $\underline{n} = [(1-\epsilon)(n-m_n+2)\pi_n(\alpha_n)]$, $\bar{n} = [(1+\epsilon)(n-m_n+2)\pi_n(\alpha_n)]$, and $n^* = [(n-m_n+2)\pi_n(\alpha_n)]$, where $[\cdot]$ is the greatest integer function.

Note that,

$$\begin{aligned} \sum_{j=0}^{n-m_n+1} \bar{g}_n(Y_j^{(n)}) &= \sum_{t=0}^{\sigma_{\alpha_n}(0)} \bar{g}_n(Y_t^{(n)}) + \sum_{j=0}^{\ell_n-1} s_j^{(n)}(\bar{g}_n) + \sum_{t=\sigma_{\alpha_n}(\ell_n)+1}^{n-m_n+1} \bar{g}_n(Y_t^{(n)}) \\ \Rightarrow \left| \sum_{j=0}^{n-m_n+1} \bar{g}_n(Y_j^{(n)}) - \sum_{j=0}^{\ell_n-1} s_j^{(n)}(\bar{g}_n) \right| &\leq \sum_{t=0}^{\sigma_{\alpha_n}(0)} \left| \bar{g}_n(Y_t^{(n)}) \right| + s_{\ell_n}^{(n)}(|\bar{g}_n|) \\ \Rightarrow \left| \frac{1}{\sqrt{n^*} \nu_n} \sum_{j=0}^{n-m_n+1} \bar{g}_n(Y_j^{(n)}) - \frac{1}{\sqrt{n^*}} \sum_{j=0}^{\ell_n-1} s_j^{(n)}(\bar{g}_n) \right| &\leq \frac{1}{\sqrt{n^*} \nu_n} \sum_{t=0}^{\sigma_{\alpha_n}(0)} \left| \bar{g}_n(Y_t^{(n)}) \right| + \frac{s_{\ell_n}^{(n)}(|\bar{g}_n|)}{\sqrt{n^*} \nu_n}. \end{aligned}$$

Using WLLN over the triangular array $\{s_j^{(n)}(|\bar{g}_n|)/v_n : 0 \leq j \leq \ell_n\}_{n \geq 1}$,

$$\begin{aligned} \frac{1}{\ell_n v_n^2} \sum_{j=0}^{\ell_n-1} (s_j^{(n)}(|\bar{g}_n|))^2 - \frac{E_{a_n}[s_0^{(n)}(|\bar{g}_n|)]^2}{v_n^2} &\xrightarrow{p} 0, \quad \text{and} \\ \frac{1}{(\ell_n + 1)v_n^2} \sum_{j=0}^{\ell_n} (s_j^{(n)}(|\bar{g}_n|))^2 - \frac{E_{a_n}[s_0^{(n)}(|\bar{g}_n|)]^2}{v_n^2} &\xrightarrow{p} 0. \end{aligned}$$

Subtracting above two terms, we get

$$\frac{(s_{\ell_n}^{(n)}(|\bar{g}_n|))^2}{\ell_n v_n^2} \xrightarrow{p} 0 \implies \frac{s_{\ell_n}^{(n)}(|\bar{g}_n|)}{\sqrt{n^*} v_n} \xrightarrow{p} 0.$$

By similar logic, $\frac{1}{\sqrt{n^*} v_n} \sum_{t=0}^{\sigma_{a_n}(0)} |\bar{g}_n(Y_t^{(n)})| \xrightarrow{p} 0$. Combining all, we get

$$\left| \frac{1}{\sqrt{n^*} v_n} \sum_{j=0}^{n-m_n+1} \bar{g}_n(Y_j^{(n)}) - \frac{1}{\sqrt{n^*}} \sum_{j=0}^{\ell_n-1} s_j^{(n)}(\bar{g}_n) \right| \xrightarrow{p} 0. \quad (\text{A.1.1})$$

Since $\frac{\ell_n}{n\pi_n(\alpha_n)} \xrightarrow{p} 1$ as $n \rightarrow \infty$, $\exists n_0 \in \mathbb{N}$ s.t. for $n \geq n_0$

$$P(\underline{n} \leq \ell_n - 1 \leq \bar{n}) \geq 1 - \epsilon.$$

Hence, by Kolmogorov's inequality, for any arbitrary $\beta > 0$,

$$\begin{aligned} P\left(\left| \frac{1}{\sqrt{n^*}} \sum_{j=0}^{\ell_n-1} s_j^{(n)}(\bar{g}_n) - \frac{1}{\sqrt{n^*}} \sum_{j=0}^{n^*} s_j^{(n)}(\bar{g}_n) \right| > \beta\right) &\leq \epsilon + P\left(\left| \max_{n \leq l \leq n^*} \sum_{j=l}^{n^*} s_j^{(n)}(\bar{g}_n) \right| > \beta \sqrt{n^*}\right) \\ &\quad + P\left(\left| \max_{n^* \leq l \leq \bar{n}} \sum_{j=n^*}^l s_j^{(n)}(\bar{g}_n) \right| > \beta \sqrt{n^*}\right) \\ &\leq \epsilon + \frac{2\epsilon\pi_n(\alpha_n)(n - m_n + 2)E_{a_n}[s_0^{(n)}(\bar{g}_n)]^2}{\beta^2 n^*} \\ &\leq \epsilon + \frac{4\epsilon}{\beta^2}. \end{aligned} \quad (\text{A.1.2})$$

From the equation A.1.2, we can conclude

$$\left| \frac{1}{\sqrt{n^*}} \sum_{j=0}^{\ell_n-1} s'_j(\bar{g}_n) - \frac{1}{\sqrt{n^*}} \sum_{j=0}^{n^*} s'_j(\bar{g}_n) \right| \xrightarrow{p} 0. \quad (\text{A.1.3})$$

Combining with equation A.1.1, we get

$$\frac{1}{\sqrt{n^*} v_n} \sum_{j=0}^{n-m_n+1} \bar{g}_n(Y_j^{(n)}) - \frac{1}{\sqrt{n^*}} \sum_{j=0}^{n^*} s'_j(\bar{g}_n) \xrightarrow{p} 0. \quad (\text{A.1.4})$$

Using CLT for row-wise i.i.d random variables with mean 0 and variance 1, we get

$$\frac{1}{\sqrt{n^*}} \sum_{j=0}^{n^*} s'_j(\bar{g}_n) \xrightarrow{d} \mathcal{N}(0, 1). \quad (\text{A.1.5})$$

Hence, from the equation A.1.4, we conclude

$$\begin{aligned} & \frac{1}{\sqrt{n^*} v_n} \sum_{j=0}^{n-m_n+1} \bar{g}_n(Y_j^{(n)}) \xrightarrow{d} \mathcal{N}(0, 1) \\ \Rightarrow & \frac{1}{\sqrt{n \pi_n(\alpha_n)} v_n} \sum_{j=0}^{n-m_n+1} \bar{g}_n(Y_j^{(n)}) \xrightarrow{d} \mathcal{N}(0, 1) \\ \Rightarrow & \frac{1}{\sqrt{n}} \sum_{j=0}^{n-m_n+1} \frac{\bar{g}_n(Y_j^{(n)})}{\sqrt{\frac{E_{\alpha_n}(\tau_{\alpha_n})}{E_{\alpha_n}[s_0^{(n)}(\bar{g}_n)]^2}}} \xrightarrow{d} \mathcal{N}(0, 1). \end{aligned}$$

A.2 Proof of Theorem 3.2.2

Consider the chain $\Phi_{2n} = \{Z_{n,j} : 1 \leq j \leq k_n\}$. By assumption (iv), $k_n \rightarrow \infty$, hence the length of the chain increases to ∞ as n increases. The transition matrix of this chain is $Q_n = P_n^{m_n}$. By assumption (i), every element of Q_n is positive. Denote the smallest element of Q_n to be q_n . For simplicity, we denote the state space $\Sigma^{m_n} = \mathcal{A}_n$. Note that, for any $a = (a_1, \dots, a_{m_n}) \in \mathcal{A}_n, b = (b_1, \dots, b_{m_n}) \in \mathcal{A}_n$,

$$P(Z_{n,2} = b | Z_{n,1} = a) = \prod_{k=1}^{m_n} P_n(\gamma_{k-1}, \gamma_k) \geq (p_n)^{m_n}$$

where $\gamma_0 = a, \gamma_{m_n} = b$ and $\gamma_k = (a_{k+1}, \dots, a_{m_n}, b_1, \dots, b_k)$ for $1 \leq k \leq m_n - 1$. Hence,

$$q_n = \min_{a, b \in \mathcal{A}_n} P(Z_{n,2} = b | Z_{n,1} = a) \geq (p_n)^{m_n}.$$

For the chain Φ_{2n} , the stationary distribution is π_n as well. Now, we have to derive a bound for the strong mixing coefficients for this chain. For $b \in \mathcal{A}_n$ and $s \in \mathbb{N}$, define $V_b^{(s)} = \max_{a \in \mathcal{A}_n} Q_n(a, b)$ and $v_b^{(s)} = \min_{a \in \mathcal{A}_n} Q_n(a, b)$. Note that for any $a, a', b \in \mathcal{A}_n$,

$$\begin{aligned} Q_n^{(s+1)}(a, b) - Q_n^{(s+1)}(a', b) &= \sum_{c \in \mathcal{A}_n} Q_n(a, c) Q_n^{(s)}(c, b) - \sum_{c \in \mathcal{A}_n} Q_n(a', c) Q_n^{(s)}(c, b) \\ &= \sum_{c \in E_{a, a'}} (Q_n(a, c) - Q_n(a', c)) Q_n^{(s)}(c, b) + \sum_{c \notin E_{a, a'}} (Q_n(a, c) - Q_n(a', c)) Q_n^{(s)}(c, b) \\ &\leq \sum_{c \in E_{a, a'}} (Q_n(a, c) - Q_n(a', c)) (V_b^{(s)} - v_b^{(s)}) \leq \delta_n (V_b^{(s)} - v_b^{(s)}) \\ &\leq \delta_n^s (V_b^{(1)} - v_b^{(1)}) \leq \delta_n^s \end{aligned}$$

where $E_{a, a'} = \{c \in \mathcal{A}_n : Q_n(a, c) \geq Q_n(a', c)\}$, and hence $\sum_{c \in E_{a, a'}} (Q_n(a, c) - Q_n(a', c)) = -\sum_{c \notin E_{a, a'}} (Q_n(a, c) - Q_n(a', c))$. This gives us the following variational inequality

$$\sup_{a \in \mathcal{A}_n} \|Q_n^{s+1}(a, \cdot) - \pi_n(\cdot)\| \leq \delta_n^s.$$

Suppose, $\alpha_n(s)$ be the s^{th} order α -mixing coefficient of the Markov chain having the transition matrix Q_n . Then, by definition,

$$\sup_{k: 1 \leq k \leq n-s} \{|\mathcal{P}(A \cap B) - \mathcal{P}(A)\mathcal{P}(B)| : A \in \mathcal{F}_1^k, B \in \mathcal{F}_{k+s}^n\} \leq \alpha_n(s)$$

where $\mathcal{F}_i^j = \sigma(Z_{n,i}, \dots, Z_{n,j})$. Now, we know that for the Harris ergodic Markov chain Φ_{2n}

$$\alpha_n(s+1) \leq \sup_{a \in \mathcal{A}_n} \|Q_n^{(s)}(a, \cdot) - \pi_n(\cdot)\| \leq (\delta_n)^s. \quad (\text{A.2.6})$$

In the section (2), the α -mixing coefficient for the whole triangular array is defined as

$$\alpha(s) = \sup_n \alpha_n(s).$$

So, from the equation (A.2.6), we get

$$\alpha(s+1) \leq \sup_n (\delta_n)^s = (\sup_n \delta_n)^s = \delta^s. \quad (\text{A.2.7})$$

Now consider the chain $\{Y_{n,j} = f_n(Z_{n,j})\}$. Denote the α -mixing coefficient for this chain as $\alpha_{f_n}(s)$, and the α -mixing coefficient for the entire triangular array as $\alpha_f(s)$. Jones et al. (2004), we know that $\alpha_{f_n}(s) \leq \alpha_n(s)$. Hence,

$$\alpha_f(s+1) \leq \sup_n \alpha_{f_n}(s+1) \leq \sup_n \alpha_n(s+1) = \alpha(s+1) \leq \delta^s.$$

Clearly, $\sum_{s=1}^{\infty} (\alpha_f(s))^{1-2/p} < \infty$ for any $p > 2$ as $\delta < 1$. Along with this, all other conditions described in the theorem (3.2.1) are satisfied by the assumptions taken in this theorem. Hence the desired CLT result holds.

A.3 Proof of the Example in Section 3.3

The obvious question is how we have derived the expression of $\pi_n(1^{m-1}0)$. Here, we provide a key steps to obtain this probability. For any history $(i_1, \dots, i_m) \in \{0, 1\}^m$, denote the index by $w(i_1, \dots, i_m) = 1 + \sum_{j=0}^{m-1} 2^j i_{m-j}$. Then the function w will provide an unique index, ranging from 1 to 2^m . For simplicity, we write $\pi_n(j)$ as the stationary probability of the m -tuple $(i_1, \dots, i_m) \in \{0, 1\}^m$ having $w(i_1, \dots, i_m) = j$ to be $\pi_n(j)$. Using the definition of stationary probability from the transition matrix, we find

$$\sum_{j=0}^{2^{m-k}-1} \pi_n(2^k j + 2^{k-1}) = \sum_{j=0}^{2^{m-k}-1} \pi_n(2^k j + 2^k - 1); \quad k = 2, 3, \dots, m. \quad (\text{A.3.8})$$

We also define the following quantities and obtain the relations

$$A_{k-1,1} := \sum_{j=0}^{2^{m-k}-1} \pi_n(2^k j + 2^{k-1} - 1) = c_{k-1} \sum_{j=0}^{2^{m-k}-1} \pi_n(2^k j + 2^{k-1}) =: c_{k-1} A_{k-1,2} \quad (\text{A.3.9})$$

and $c_{k-1} = (1 - p_{k-2})/p_{k-2}$ for $k = 2, 3, \dots, m$ and $c_m = (1 - p_{m-1})/p_m$. Define $A_{m,1} = \pi_n(2^m - 1)$, $A_{m,2} = \pi_n(2^m)$. Now, observe that

$$\begin{aligned} A_{m-2,1} &= c_{m-2} A_{m-2,2} = c_{m-2} (A_{m-1,1} + A_{m,1}) \\ &= c_{m-2} (1 + c_{m-1}) A_{m,1}; \\ A_{m-3,1} &= c_{m-3} (1 + c_{m-2}) (1 + c_{m-1}) A_{m,1}; \\ &\cdot \\ &\cdot \\ A_{1,1} &= c_1 (1 + c_2) \dots (1 + c_{m-1}) A_{m,1} \end{aligned}$$

and

$$\begin{aligned} A_{1,2} &= (1 + c_2) \dots (1 + c_{m-1}) A_{m,1} \\ &= c_{m-2} (1 + c_{m-1}) A_{m,1}; \\ A_{2,2} &= (1 + c_3) \dots (1 + c_{m-1}) A_{m,1}; \\ &\cdot \\ &\cdot \\ A_{m-2,2} &= (1 + c_{m-1}) A_{m,1}; \\ A_{m-1,2} &= A_{m,1}. \end{aligned}$$

Combining all the relations, we find

$$A_{m,1} = \pi_n(2^m - 1) = \frac{1}{\prod_{i=1}^{m-1} a_i + a_m + \sum_{i=2}^{m-1} a_i a_{i+1} \dots a_{m-1}}$$

where $a_i = 1 + c_i$. Replacing the values of c_i , we find that

$$\frac{1}{\pi_n(1^{m-1}0)} = \prod_{j=0}^{m-2} \frac{1}{p_j} + \frac{p_{m-1} + 1 - p_m}{1 - p_m} + \sum_{j=1}^{m-2} \prod_{k=j}^{m-2} \frac{1}{p_j} = q(p, m);$$

APPENDIX

B

PROOF OF THE THEOREMS IN CHAPTER

4

B.1 Proof of Theorem 4.3.3

(a) For notational simplicity, we write $b_{i,a}^*(\lambda)$ as $b_{i,a}^*$. Let

$$R(\mathbf{B}, \mathbf{W}) = \frac{1}{2} \sum_{j=1}^p \|\hat{\pi}_j - \mathbf{b}_j\|_2^2 + \lambda \sum_{1 \leq i < j \leq p} w_{i,j} \|\mathbf{b}_i, \mathbf{b}_j\|_2^2.$$

Suppose $b_{i,a}^* < 0$ for some of the (i, a) pairs, $i = 1, 2, \dots, p$ and $a = 1, 2, \dots, d$. Let, $b_{i,a}^{**} = b_{i,a}^* \mathcal{I}(b_{i,a}^* > 0)$.

Since $\hat{\pi}_{i,a} \geq 0$, we get $|\hat{\pi}_{i,a} - b_{i,a}^{**}| \leq |\hat{\pi}_{i,a} - b_{i,a}^*|$. Also, $|b_{i_1,a}^* - b_{i_2,a}^*| \geq |b_{i_1,a}^{**} - b_{i_2,a}^{**}|$, since the negative elements are shrunk to 0. Hence for any $i = 1, 2, \dots, p$,

$$\|\hat{\pi}_i - \mathbf{b}_i^*\|_2^2 \geq \|\hat{\pi}_i - \mathbf{b}_i^{**}\|_2^2; \quad \|\mathbf{b}_{i_1}^* - \mathbf{b}_{i_2}^*\|_2 \geq \|\mathbf{b}_{i_1}^{**} - \mathbf{b}_{i_2}^{**}\|_2.$$

Since $\mathbf{b}_{i_1}^* \neq \mathbf{b}_{i_2}^*$ for at least one i , $R(\mathbf{B}^{**}, \mathbf{W}) < R(\mathbf{B}^*, \mathbf{W})$, contradicting that \mathbf{B}^* is the optimum solution. Hence $b_{i,a}^* \geq 0, \forall i = 1, \dots, p; a = 1, \dots, d$.

(b) If we initialize $\Gamma^{(0)} = \mathbf{0}$, we get $\mathbf{b}_i^{(1)} = \hat{\pi}_i$, which satisfies $\sum_{a=1}^d b_{i,a}^{(1)} = 1$. Subsequently, $\gamma_i^{(1)} = \mathcal{P}_{C_i}(\gamma_i^{(0)} - \nu \mathbf{g}_i^{(1)}) = (\gamma_i^{(0)} - \nu \mathbf{g}_i^{(1)}) \min \left\{ 1, \frac{\lambda w_i}{\|\gamma_i^{(0)} - \nu \mathbf{g}_i^{(1)}\|_2} \right\}$, and thus $\gamma_i^{(1)T} \mathbf{1} = 0$. Using a similar argument, for any iteration t , $\sum_{a=1}^d b_{i,a}^{(t)} = 1$. Hence the limiting quantity will still have the property that the sum of the elements of b_i is always 1. This completes the proof that \mathbf{b}_i^* is indeed a probability distribution.

B.2 Proof of Theorem 4.3.4

Note that as $n \rightarrow \infty$, $N_{\sigma_j} \rightarrow \infty$. Let q_j be the stationary probability of the state σ_j . Then, $N_{\sigma_j}/(n-m) \xrightarrow{P} q_j$ as $n \rightarrow \infty$; and for $j \in \mathcal{C}_\alpha$, we have

$$\begin{aligned} \sqrt{N_{\sigma_j}}(\hat{\pi}_j - \mathbf{R}_\alpha) &\xrightarrow{d} \mathcal{N}(\mathbf{0}, \Sigma_\alpha) \\ \implies \sqrt{(n-m)}(\hat{\pi}_j - \mathbf{R}_\alpha) &\xrightarrow{d} \mathcal{N}(\mathbf{0}, q_j \Sigma_\alpha) \end{aligned}$$

where $\Sigma_\alpha = \text{diag}(\mathbf{R}_\alpha) - \mathbf{R}_\alpha \mathbf{R}_\alpha^{(T)}$.

The proof mainly relies on calculating the probability of $\hat{\pi}_j$ and \mathbf{R}_α , $j \in \mathcal{C}_\alpha$ being close to each other. Suppose $\|\hat{\pi}_j - \mathbf{R}_\alpha\|_2 < \epsilon/2$ for some $\epsilon > 0$, and $\forall j \in \mathcal{C}_\alpha, \alpha = 1, 2, \dots, k_0$. In that case,

$$\begin{aligned} \lambda_{\min}^{(n)} &< \frac{\epsilon/2}{\min_{1 \leq \alpha \leq k_0} \min_{i, j \in \mathcal{C}_\alpha} (p_\alpha w_{i,j} - \mu_{i,j}^{(\alpha)})} \\ \lambda_{\max}^{(n)} &> \min_{1 \leq \alpha \leq k_0} \left\{ \frac{\|\mathbf{R}_\alpha - \mathbf{R}_\beta\|_2 - \epsilon}{\frac{1}{p_\alpha} \sum_{l \neq \alpha} w^{(\alpha, l)} + \frac{1}{p_\beta} \sum_{l \neq \beta} w^{(\beta, l)}} \right\}. \end{aligned}$$

Thus, for ϵ sufficiently small, $\lambda_{\min}^{(n)} < \lambda_{\max}^{(n)}$. We will later find a bound on how small ϵ needs to be to achieve this.

We compute a lower bound on the following probability:

$$P\left(\|\hat{\pi}_j - \mathbf{R}_\alpha\|_2 < \frac{\epsilon}{2}; \forall j \in \mathcal{C}_\alpha, \forall \alpha = 1, \dots, k_0\right).$$

Note that the variance-covariance matrix Σ_α of the limiting distribution is not full rank, as we have a linear constraint on the elements of $\boldsymbol{\pi}_j$. Define $Z_j = (\hat{\pi}_{j,1} - R_{j,1}, \dots, \hat{\pi}_{j,d-1} - R_{j,d-1})^T$, and let $\Sigma_{\alpha,-d}$ be the upper $(d-1) \times (d-1)$ block of Σ_α . Now,

$$\begin{aligned} \|\hat{\boldsymbol{\pi}}_j - \mathbf{R}_\alpha\|_2^2 &= \sum_{l=1}^{d-1} (\hat{\pi}_{j,l} - R_{j,l})^2 + \left(\sum_{l=1}^{d-1} (\hat{\pi}_{j,l} - R_{j,l}) \right)^2 \\ &= Z_j^T Z_j + (\mathbf{1}^T Z_j)^2 = Z_j^T (\mathbf{I} + \mathbf{1}\mathbf{1}^T) Z_j. \end{aligned}$$

Define $U_j = \sqrt{\frac{n-m}{q_j}} \Sigma_{\alpha,-d}^{-1/2} Z_j$. By asymptotic normality of $\hat{\boldsymbol{\pi}}_j$, $\sqrt{n-m} Z_j \xrightarrow{d} \mathcal{N}(\mathbf{0}, q_j \Sigma_{\alpha,-d})$, hence $U_j \xrightarrow{d} \mathcal{N}(\mathbf{0}, \mathbf{I})$. Thus,

$$\begin{aligned} P\left(\|\hat{\boldsymbol{\pi}}_j - \mathbf{R}_\alpha\|_2 \geq \frac{\epsilon}{2}\right) &= P\left(Z_j^T (\mathbf{I} + \mathbf{1}\mathbf{1}^T) Z_j \geq \frac{\epsilon^2}{4}\right) = P\left(U_j^T \Sigma_{\alpha,-d}^{1/2} (\mathbf{I} + \mathbf{1}\mathbf{1}^T) \Sigma_{\alpha,-d}^{1/2} U_j \geq \frac{(n-m)\epsilon^2}{4q_j}\right) \\ &= P\left(U_j^T \mathbf{M} U_j \geq \frac{(n-m)\epsilon^2}{4q_j}\right); \quad \mathbf{M} = \Sigma_{\alpha,-d}^{1/2} (\mathbf{I} + \mathbf{1}\mathbf{1}^T) \Sigma_{\alpha,-d}^{1/2}. \end{aligned}$$

For a symmetric matrix \mathbf{M}_1 , Hanson and Wright (1971) have determined a lower bound on the tail probability of any quadratic form $U^T \mathbf{M}_1 U$ of a sub-Gaussian random variable U with mean $\mathbf{0}$ and variance-covariance matrix $\sigma^2 \mathbf{I}$ as follows:

$$P\left(U^T \mathbf{M}_1 U \geq t + \sigma^2 t r(\mathbf{M}_1)\right) \leq \exp\left[-\min\left(\frac{a_1 t^2}{\sigma^4 \|\mathbf{M}_1\|_F}, \frac{a_2 t}{\sigma^2 \|\mathbf{M}_1\|_{sp}}\right)\right] \quad (\text{B.2.1})$$

for some constants $a_1, a_2 > 0$. Here $\|\cdot\|_F$ and $\|\cdot\|_{sp}$ are the Frobenius and spectral norms, respectively.

Applying the bound in (B.2.1) to our problem, we obtain, as $n \rightarrow \infty$,

$$\begin{aligned} P\left(\|\hat{\boldsymbol{\pi}}_j - \mathbf{R}_\alpha\|_2 \geq \frac{\epsilon}{2}\right) &= P\left(U_j^T \mathbf{M} U_j \geq \frac{(n-m)\epsilon^2}{4q_j}\right) \\ &\leq \exp\left[-\min\left(\frac{a_1((n-m)\epsilon^2 - 4q_j t r(\mathbf{M}))^2}{16q_j^2 \|\mathbf{M}\|_F}, \frac{a_2((n-m)\epsilon^2 - 4q_j t r(\mathbf{M}))}{4q_j \|\mathbf{M}\|_{sp}}\right)\right]. \end{aligned}$$

As n increases, $(n-m)^2 \gg (n-m)$, and eventually for larger n , $\min\left(\frac{a_1((n-m)\epsilon^2 - 4q_j t r(\mathbf{M}))^2}{16q_j^2 \|\mathbf{M}\|_F}, \frac{a_2((n-m)\epsilon^2 - 4q_j t r(\mathbf{M}))}{4q_j \|\mathbf{M}\|_{sp}}\right)$,

$$\left. \frac{a_2((n-m)\epsilon^2 - 4q_j \operatorname{tr}(\mathbf{M}))}{4q_j \|\mathbf{M}\|_{sp}} \right) = \frac{a_2((n-m)\epsilon^2 - 4q_j \operatorname{tr}(\mathbf{M}))}{4q_j \|\mathbf{M}\|_{sp}}. \text{ Now,}$$

$$\begin{aligned} \operatorname{tr}(\mathbf{M}) &= \operatorname{tr}(\Sigma_{\alpha,-d}^{1/2}(\mathbf{I} + \mathbf{1}\mathbf{1}^T)\Sigma_{\alpha,-d}^{1/2}) = \operatorname{tr}(\Sigma_{\alpha,-d}) + \operatorname{tr}(\mathbf{1}^T \Sigma_{\alpha,-d} \mathbf{1}) \\ &= \sum_{l=1}^{d-1} R_{\alpha,l}(1 - R_{\alpha,l}) + \sum_{l=1}^{d-1} R_{\alpha,l} - \sum_{l_1=1}^{d-1} \sum_{l_2=1}^{d-1} R_{\alpha,l_1} R_{\alpha,l_2} \\ &= \sum_{l=1}^{d-1} R_{\alpha,l}(1 - R_{\alpha,l}) + \left(\sum_{l=1}^{d-1} R_{\alpha,l} \right) \left(1 - \sum_{l=1}^{d-1} R_{\alpha,l} \right) = \sum_{l=1}^d R_{\alpha,l}(1 - R_{\alpha,l}) = s_\alpha(s_\alpha y); \end{aligned}$$

$$\|\mathbf{M}\|_{sp} = \|\Sigma_{\alpha,-d} + \Sigma_{\alpha,-d}^{1/2} \mathbf{1}\mathbf{1}^T \Sigma_{\alpha,-d}^{1/2}\|_{sp} \leq \|\Sigma_{\alpha,-d}\|_{sp} + \mathbf{1}^T \Sigma_{\alpha,-d} \mathbf{1} \leq \max_{l=1,2,\dots,d-1} R_{\alpha,l} + R_{\alpha,d}(1 - R_{\alpha,d}) = v_\alpha$$

as $\|\Sigma_{\alpha,-d}\|_{sp} \leq \max_{l=1,2,\dots,d-1} R_{\alpha,l}$ by the result of Watson (1996). Hence,

$$\begin{aligned} P\left(\|\hat{\pi}_j - \mathbf{R}_\alpha\|_2 \geq \frac{\epsilon}{2}\right) &\leq \exp\left[-\frac{a_2((n-m)\epsilon^2 - 4q_j s_\alpha)}{4q_j v_\alpha}\right] = \exp\left(\frac{s_\alpha}{v_\alpha}\right) \exp\left[-\frac{a_2(n-m)\epsilon^2}{4q_j v_\alpha}\right] \\ \implies P\left(\|\hat{\pi}_j - \mathbf{R}_\alpha\|_2 < \frac{\epsilon}{2}; \forall j \in \mathcal{C}_\alpha, \forall \alpha = 1, \dots, k_0\right) \\ &\geq 1 - \sum_{\alpha=1}^{k_0} \sum_{j \in \mathcal{C}_\alpha} P\left(\|\hat{\pi}_j - \mathbf{R}_\alpha\|_2 \geq \frac{\epsilon}{2}\right) \geq 1 - \sum_{\alpha=1}^{k_0} \exp\left(\frac{s_\alpha}{v_\alpha}\right) \sum_{j \in \mathcal{C}_\alpha} \exp\left[-\frac{a_2(n-m)\epsilon^2}{4q_j v_\alpha}\right]. \end{aligned}$$

Now, setting $C_1^{(\alpha)} = \exp\left(\frac{s_\alpha}{v_\alpha}\right)$, $C_{2,j} = \frac{a_2 \epsilon^2}{4q_j v_\alpha}$, one gets the conclusions of the theorem.

B.3 Proof of Theorem 4.3.5

Recall that $\hat{\pi}_{j,\ell} = N_{\sigma_j,\ell}/N_{\sigma_j}$. Denote the common transition probability for the estimated group

$\hat{\mathcal{C}}_\alpha(\lambda)$ as

$$\hat{R}_{\alpha,\ell}^{(\lambda)} = \frac{\sum_{\sigma_j \in \hat{\mathcal{C}}_\alpha(\lambda)} N_{\sigma_j,\ell}}{\sum_{\sigma_j \in \hat{\mathcal{C}}_\alpha(\lambda)} N_{\sigma_j}} = \frac{N_{\hat{\mathcal{C}}_\alpha(\lambda),\ell}}{N_{\hat{\mathcal{C}}_\alpha(\lambda)}} \quad \forall \alpha = 1, \dots, k_\lambda; \ell = 1, \dots, d.$$

Thus, the log-likelihood is given by

$$\ell_n(\lambda) = \sum_{\alpha=1}^{k_\lambda} \sum_{\ell=1}^d N_{\hat{\mathcal{C}}_\alpha(\lambda),\ell} \log \hat{R}_{\alpha,\ell}^{(\lambda)}.$$

Note that, as λ increases, the number of clusters decreases. Also, by the continuity of the solution of (4.2.3) w.r.t. λ , M_{λ_2} is a submodel of M_{λ_1} for $\lambda_1 < \lambda_2$ as the separate clusters for lower λ values are clumped together to form new clusters with larger size as λ increases. Hence, we can write $M_{\lambda_2} \subseteq M_{\lambda_1}$. Subsequently, $\ell_n(\lambda_1) \geq \ell_n(\lambda_2)$. Let q_j be the stationary probability of the state σ_j , and $Q^{(a)}(\lambda)$ be the stationary probability of the partition $\hat{\mathcal{C}}_a(\lambda)$. Thus, $Q^{(a)}(\lambda) = \sum_{\sigma_j \in \hat{\mathcal{C}}_a(\lambda)} q_j$. We have to show that the true model minimizes the BIC with probability tending to 1 as $n \rightarrow \infty$. We prove this for two cases.

Case 1: Suppose that $\lambda < \lambda_0$ and $M_{\lambda_0} \subset M_\lambda$. Clearly, $k_{\lambda_0} < k_\lambda$. Since M_{λ_0} is the true underlying model by Theorem (1), $M_{\lambda_0} = \{\mathcal{C}_1, \dots, \mathcal{C}_{k_0}\}$, and

$$Z_n = -2\left(\ell_n(\lambda_0) - \ell_n(\lambda)\right) \xrightarrow{d} \chi_{(k_\lambda - k_0)(d-1)}^2.$$

Hence, as $n \rightarrow \infty$,

$$\begin{aligned} P\left(BIC_n(\lambda_0) \geq BIC_n(\lambda)\right) &= P\left(Z_n > (k_\lambda - k_0)(d-1) \log n\right) \\ &\leq \exp\left[-\frac{(k_\lambda - k_0)(d-1)}{4} \log n\right] \\ &= n^{-\frac{(k_\lambda - k_0)(d-1)}{4}} \rightarrow 0. \end{aligned}$$

Case 2: Now let $\lambda_0 < \lambda$ and $M_\lambda \subset M_{\lambda_0}$. For $\alpha' = 1, \dots, k_\lambda$, w.lo.g. we can write

$$\hat{\mathcal{C}}_{\alpha'}(\lambda) = \bigcup_{\alpha=t_{\alpha'-1}+1}^{\alpha=t_{\alpha'}} \mathcal{C}_\alpha$$

for $0 = t_0 < t_1 < t_2 < \dots < t_{k_\lambda} = k_0$. Now, as $n \rightarrow \infty$,

$$\begin{aligned} \frac{1}{n-m} \ell_n(\lambda_0) &= \frac{1}{n-m} \sum_{\alpha=1}^{k_0} \sum_{\ell=1}^d N_{\mathcal{C}_{\alpha,\ell}} \log \hat{R}_{\alpha,\ell}^{(\lambda_0)} \xrightarrow{p} \sum_{\alpha=1}^{k_0} \sum_{\ell=1}^d \left(\sum_{j \in \mathcal{C}_\alpha} q_{\sigma_j} \right) R_{\alpha,\ell} \log R_{\alpha,\ell} \\ &= \sum_{\alpha=1}^{k_0} \sum_{\ell=1}^d Q^{(a)}(\lambda_0) R_{\alpha,\ell} \log R_{\alpha,\ell} = A_0; \end{aligned}$$

and

$$\begin{aligned}
\frac{1}{n-m} \ell_n(\lambda) &= \frac{1}{n-m} \sum_{\alpha'=1}^{k_\lambda} \sum_{\ell=1}^d N_{\hat{\mathcal{C}}_{\alpha'}(\lambda), \ell} \log \hat{R}_{\alpha', \ell}^{(\lambda)} = \frac{1}{n-m} \sum_{\alpha'=1}^{k_\lambda} \sum_{\ell=1}^d \left(\sum_{j \in \hat{\mathcal{C}}_{\alpha'}(\lambda)} N_{\sigma_j, \ell} \right) \log \left(\frac{\sum_{j \in \hat{\mathcal{C}}_{\alpha'}(\lambda)} N_{\sigma_j, \ell}}{\sum_{j \in \hat{\mathcal{C}}_{\alpha'}(\lambda)} N_{\sigma_j}} \right) \\
&= \frac{1}{n-m} \sum_{\alpha'=1}^{k_\lambda} \sum_{\ell=1}^d \left(\sum_{\alpha=t_{\alpha'-1}+1}^{t_{\alpha'}} N_{\mathcal{C}_{\alpha, \ell}} \right) \log \left(\frac{\sum_{\alpha=t_{\alpha'-1}+1}^{t_{\alpha'}} N_{\mathcal{C}_{\alpha, \ell}}}{\sum_{\alpha=t_{\alpha'-1}+1}^{t_{\alpha'}} N_{\mathcal{C}_{\alpha}}} \right) \\
&\stackrel{p}{\rightarrow} \sum_{\alpha'=1}^{k_\lambda} \sum_{\ell=1}^d \left(\sum_{\alpha=t_{\alpha'-1}+1}^{t_{\alpha'}} Q^{(\alpha)}(\lambda_0) R_{\alpha, \ell} \right) \log \left(\frac{\sum_{\alpha=t_{\alpha'-1}+1}^{t_{\alpha'}} Q^{(\alpha)}(\lambda_0) R_{\alpha, \ell}}{\sum_{\alpha=t_{\alpha'-1}+1}^{t_{\alpha'}} Q^{(\alpha)}(\lambda_0)} \right) = A(\lambda).
\end{aligned}$$

Now, applying the Jensen's inequality by using the strict convexity of $-\log x$,

$$\begin{aligned}
A(\lambda) &= - \sum_{\ell=1}^d \sum_{\alpha'=1}^{k_\lambda} \left(\sum_{\alpha=t_{\alpha'-1}+1}^{t_{\alpha'}} Q^{(\alpha)}(\lambda_0) R_{\alpha, \ell} \right) \log \left(\frac{\sum_{\alpha=t_{\alpha'-1}+1}^{t_{\alpha'}} Q^{(\alpha)}(\lambda_0)}{\sum_{\alpha=t_{\alpha'-1}+1}^{t_{\alpha'}} Q^{(\alpha)}(\lambda_0) R_{\alpha, \ell}} \right) \\
&= - \sum_{\ell=1}^d \sum_{\alpha'=1}^{k_\lambda} \left(\sum_{\alpha=t_{\alpha'-1}+1}^{t_{\alpha'}} Q^{(\alpha)}(\lambda_0) R_{\alpha, \ell} \right) \log \left(\frac{\sum_{\alpha=t_{\alpha'-1}+1}^{t_{\alpha'}} Q^{(\alpha)}(\lambda_0) R_{\alpha, \ell} \cdot (1/R_{\alpha, \ell})}{\sum_{\alpha=t_{\alpha'-1}+1}^{t_{\alpha'}} Q^{(\alpha)}(\lambda_0) R_{\alpha, \ell}} \right) \\
&< - \sum_{\ell=1}^d \sum_{\alpha'=1}^{k_\lambda} \sum_{\alpha=t_{\alpha'-1}+1}^{t_{\alpha'}} Q^{(\alpha)}(\lambda_0) R_{\alpha, \ell} \log(1/R_{\alpha, \ell}) \\
&= \sum_{\ell=1}^d \sum_{\alpha'=1}^{k_\lambda} \sum_{\alpha=t_{\alpha'-1}+1}^{t_{\alpha'}} Q^{(\alpha)}(\lambda_0) R_{\alpha, \ell} \log R_{\alpha, \ell} = A_0.
\end{aligned}$$

Hence, $\frac{1}{n-m} (\ell_n(\lambda_0) - \ell_n(\lambda)) \stackrel{p}{\rightarrow} A_0 - A(\lambda) > 0$, and $P\left(\frac{1}{n-m} (\ell_n(\lambda_0) - \ell_n(\lambda)) \geq \frac{1}{2} (A_0 - A(\lambda))\right) \rightarrow 1$ as $n \rightarrow \infty$. Since $\log n/N \rightarrow 0$ as $n \rightarrow \infty$,

$$\begin{aligned}
P(BIC_n(\lambda_0) \geq BIC_n(\lambda)) &= P(2\ell_n(\lambda_0) \leq 2\ell_n(\lambda) + (k_0 - k_\lambda)(d-1)\log n) \\
&= P(\ell_n(\lambda_0) - \ell_n(\lambda) \leq (k_0 - k_\lambda)(d-1)\log n) \\
&= P\left(\frac{1}{n-m} (\ell_n(\lambda_0) - \ell_n(\lambda)) \leq (k_0 - k_\lambda)(d-1) \frac{\log n}{n-m}\right) \\
&\rightarrow 0.
\end{aligned}$$

B.4 Proof of Theorem 4.3.6

By definition we can easily conclude that the weights $w_{i,j}$ are symmetric, hence the first part of (A1) is satisfied. Now, observe that,

$$\begin{aligned}\|\hat{\pi}_i - \hat{\pi}_j\|_2 &\leq \|\hat{\pi}_i - \mathbf{R}_\alpha\|_2 + \|\hat{\pi}_j - \mathbf{R}_\alpha\|_2 < \epsilon, \text{ for } i, j \in \mathcal{C}_\alpha \\ \|\hat{\pi}_i - \hat{\pi}_j\|_2 &\geq \|\mathbf{R}_\alpha - \mathbf{R}_\beta\|_2 - \|\hat{\pi}_i - \mathbf{R}_\alpha\|_2 - \|\hat{\pi}_j - \mathbf{R}_\beta\|_2 \\ &\geq \delta - \epsilon, \text{ for } i \in \mathcal{C}_\alpha, j \in \mathcal{C}_\beta, \alpha \neq \beta.\end{aligned}$$

Hence, for $\epsilon < \delta/2$, $w_{i,j} > 0$ for $i, j \in \mathcal{C}_\alpha$, and thus (A1) holds.

First, assume that the cluster sizes are different. Note that, for $i \in \mathcal{C}_\alpha$,

$$\sum_{\beta \neq \alpha} w_i^{(\beta)} = \sum_{\beta \neq \alpha} \sum_{i' \in \mathcal{C}_\beta} w_{i,i'} \leq (k' + 1 - p_\alpha) \exp[-\phi(\delta - \epsilon)^2],$$

since at most $k' - (p_\alpha - 1)$ many $w_{i,i'}$ can be non-zero if $i' \notin \mathcal{C}_\alpha$. Thus, for $i, j \in \mathcal{C}_\alpha$

$$\mu_{i,j}^{(\alpha)} \leq \sum_{\beta \neq \alpha} w_i^{(\beta)} + \sum_{\beta \neq \alpha} w_j^{(\beta)} \leq 2(k' + 1 - p_\alpha) \exp[-\phi(\delta - \epsilon)^2];$$

and

$$\begin{aligned}\frac{\mu_{i,j}^{(\alpha)}}{p_\alpha w_{i,j}} &< \frac{2(k' + 1 - p_\alpha) \exp[-\phi(\delta - \epsilon)^2]}{p_\alpha \exp[-\phi \epsilon^2]} < 2\left(\frac{k' + 1}{p_{min}} - 1\right) \exp[-\phi(\delta^2 - 2\delta\epsilon)] \\ &= \exp\left[2\phi\delta\left(\epsilon - \frac{\delta}{2} + \frac{1}{2\phi\delta} \log\left(\frac{2(k' + 1 - p_{min})}{p_{min}}\right)\right)\right] = \exp[2\phi\delta(\epsilon - \epsilon_{max})] \\ &< 1, \text{ for } \epsilon < \epsilon_{max}.\end{aligned}$$

Thus Condition (A2) holds. Now,

$$\begin{aligned}\delta_1 &\geq p_{min} \exp[-\phi \epsilon^2] - 2(k' + 1 - p_\alpha) \exp[-\phi(\delta - \epsilon)^2] \\ &\geq p_{min} \exp[-\phi \epsilon_{max}^2] - 2(k' + 1 - p_\alpha) \exp[-\phi(\delta - \epsilon_{max})^2] = \delta_1^{(min)}.\end{aligned}$$

Also,

$$w^{(\alpha,l)} = \sum_{i \in \mathcal{C}_\alpha} w_i^{(l)} \leq p_\alpha (k' + 1 - p_\alpha) \exp[-\phi(\delta - \epsilon)^2].$$

Hence,

$$\begin{aligned}\delta_2 &\leq \max_{1 \leq \alpha < \beta \leq k_0} (2k' + 2 - p_\alpha - p_\beta) \exp[-\phi(\delta - \epsilon)^2] \\ &\leq 2(k' + 1 - p_{min}) \exp[-\phi(\delta - \epsilon_{max})^2] = \delta_2^{(max)}.\end{aligned}$$

B.4.1 Proof of Corollary 4.3.6.1

Note that (a) follows directly from the definition of $\lambda_{min}^{(n)}$ and $\lambda_{max}^{(n)}$. Now if (a) holds, $\lambda_{min}^{(n)} < \lambda_{max}^{(n)}$ if $\epsilon < \frac{\delta \delta_1^{(min)}}{\delta_1^{(min)} + \delta_2^{(max)}}$, and (b) holds if $\epsilon < \epsilon_{max}$, proving the result.

B.4.2 Proof of Corollary 4.3.6.2

Under the conditions of theorem (4.3.6), $w_{i,j} > 0$ for $i, j \in \mathcal{C}_\alpha$ for some $\alpha = 1, \dots, k_0$. If $p_\alpha = p/k_0$, and $k = p/k_0 - 1$, then $k' = k = p/k_0 - 1$ as well, since all the weights $w_{i,j} = 0$ if the two m -tuples σ_i and σ_j belong to different cluster. Hence, $\delta_2^{(max)} = 0$. Also, in this case $\mu_{i,j}^{(\alpha)} = 0$ if $\epsilon < \delta/2$, and hence $p_\alpha w_{i,j} > \mu_{i,j}^{(\alpha)} = 0$ for $i, j \in \mathcal{C}_\alpha$. Thus,

$$\delta_1 \geq (p/k_0) \exp[-\phi \epsilon^2] \geq (p/k_0) \exp[-\phi(\delta/2)^2]$$

and $\lambda_{min}^{(n)} < \frac{\delta}{2\delta_1}$, $\lambda_{max}^{(n)} = \infty$.

APPENDIX

C

SUPPLEMENTS FOR CHAPTER 5

C.1 Stopping Criterion for ADMM

Consider the following constrained optimization problem

$$\begin{aligned} & \min_{\mathbf{b}, \mathbf{v}} f(\mathbf{b}) + g(\mathbf{v}) \\ & \text{subject to } \mathbf{A}_1 \mathbf{b} + \mathbf{B}_1 \mathbf{v} = \mathbf{c}_1 \\ & \mathbf{A}_2 \mathbf{b} + \mathbf{B}_2 \mathbf{v} = \mathbf{c}_2 \end{aligned}$$

The augmented Lagrangian for this problem is

$$\begin{aligned} \mathcal{L}_\gamma(\mathbf{b}, \mathbf{v}, \gamma, \phi) &= f(\mathbf{b}) + g(\mathbf{v}) + \langle \gamma, \mathbf{c}_1 - \mathbf{A}_1 \mathbf{b} - \mathbf{B}_1 \mathbf{v} \rangle + \langle \phi, \mathbf{c}_2 - \mathbf{A}_2 \mathbf{b} - \mathbf{B}_2 \mathbf{v} \rangle \\ &+ \frac{\nu}{2} \|\mathbf{c}_1 - \mathbf{A}_1 \mathbf{b} - \mathbf{B}_1 \mathbf{v}\|_2^2 + \frac{\nu}{2} \|\mathbf{c}_2 - \mathbf{A}_2 \mathbf{b} - \mathbf{B}_2 \mathbf{v}\|_2^2. \end{aligned}$$

If the optimum solution of the optimization problem is $(\mathbf{b}^*, \mathbf{v}^*, \gamma^*, \phi^*)$, then they satisfy the primal feasibility

$$\mathbf{A}_1 \mathbf{b}^* + \mathbf{B}_1 \mathbf{v}^* = \mathbf{c}_1 \quad (\text{C.1.1})$$

$$\mathbf{A}_2 \mathbf{b}^* + \mathbf{B}_2 \mathbf{v}^* = \mathbf{c}_2;$$

and the dual feasibility

$$\mathbf{0} \in \partial f(\mathbf{b}^*) - \mathbf{A}_1^T \gamma^* - \mathbf{A}_2^T \phi^* \quad (\text{C.1.2})$$

$$\mathbf{0} \in \partial g(\mathbf{v}^*) - \mathbf{B}_1^T \gamma^* - \mathbf{B}_2^T \phi^*.$$

The ADMM updates after $(t+1)^{th}$ iteration satisfy the equation

$$\mathbf{0} \in \partial f(\mathbf{b}^{(t+1)}) - \mathbf{A}_1^T \gamma^{(t+1)} - \mathbf{A}_2^T \phi^{(t+1)} + \nu \mathbf{A}_1^T \mathbf{B}_1 (\mathbf{v}^{(t)} - \mathbf{v}^{(t+1)}) + \nu \mathbf{A}_2^T \mathbf{B}_2 (\mathbf{v}^{(t)} - \mathbf{v}^{(t+1)}) \quad (\text{C.1.3})$$

$$\nu \mathbf{A}_1^T \mathbf{B}_1 (\mathbf{v}^{(t+1)} - \mathbf{v}^{(t)}) + \nu \mathbf{A}_2^T \mathbf{B}_2 (\mathbf{v}^{(t+1)} - \mathbf{v}^{(t)}) \in \partial f(\mathbf{b}^{(t+1)}) - \mathbf{A}_1^T \gamma^{(t+1)} - \mathbf{A}_2^T \phi^{(t+1)}.$$

The LHS of the above term should converge to $\mathbf{0}$ as the RHS will be $\mathbf{0}$ when the iterations reach optimality. We denote the quantity in the LHS as the dual residual. In the context of convex clustering, Chi and Lange (2015) have shown that $\mathbf{A}_1 = [A_{1,1}, \dots, A_{1,|\mathcal{S}|}]$, where $A_{1,l} = (\mathbf{e}_{l_1} - \mathbf{e}_{l_2}) \otimes \mathbf{I}$, $\mathbf{B}_1 = -\mathbf{I}_{d|\mathcal{S}|}$ and $\mathbf{c}_1 = \mathbf{0}$. From the constraints $\mathbf{1}^T \mathbf{b}_j = 1$, we can conclude,

$$\mathbf{A}_2 = \mathbf{I}_p \otimes \mathbf{1}_d, \mathbf{B}_2 = [\mathbf{0}], \mathbf{c}_2 = \mathbf{1}_p.$$

Note that, we have taken $\mathbf{b}^T = (\mathbf{b}_1^T, \dots, \mathbf{b}_p^T)$ and $\mathbf{v}^T = (\mathbf{v}_1^T, \dots, \mathbf{v}_{|\mathcal{S}|}^T)$. Hence the p many dual residual vectors are

$$\mathbf{s}_i^{(t+1)} = -\nu \left[\sum_{l_1=i} (\mathbf{v}_{l_1}^{(t+1)} - \mathbf{v}_{l_1}^{(t)}) - \sum_{l_2=i} (\mathbf{v}_{l_2}^{(t+1)} - \mathbf{v}_{l_2}^{(t)}) \right].$$

The primal residuals are already defined in section 5.3. Denote $\mathbf{A}^T = [\mathbf{A}_1^T, \mathbf{A}_2^T]$, $\mathbf{B}^T = [\mathbf{B}_1^T, \mathbf{B}_2^T]$, $\mathbf{c}^T = (\mathbf{c}_1^T, \mathbf{c}_2^T)$ and $\boldsymbol{\eta}^T = (\gamma^T, \phi^T)$. Boyd et al. (2011) suggests to stop if $\|\mathbf{r}^{(t)}\|_2 \leq \epsilon_{\text{pri}}$ and $\|\mathbf{s}^{(t)}\|_2 \leq \epsilon_{\text{dual}}$, where

$$\epsilon_{\text{pri}} = \sqrt{p} \epsilon_{\text{abs}} + \epsilon_{\text{rel}} \max \left\{ \|\mathbf{A} \mathbf{b}^{(t)}\|_2, \|\mathbf{V}^{(t)}\|_F, \|\mathbf{c}\|_2 \right\};$$

$$\epsilon_{\text{dual}} = \sqrt{d} \epsilon_{\text{abs}} + \epsilon_{\text{rel}} \|\mathbf{A}^T \boldsymbol{\eta}^{(t)}\|_2.$$

After some basic algebra, we find that

$$\begin{aligned}\|\mathbf{A}\mathbf{b}^{(t)}\|_2 &= \sqrt{\sum_{i<j} \|\mathbf{b}_{l_1}^{(t)} - \mathbf{b}_{l_2}^{(t)}\|_2^2 + \sum_{i=1}^p (\mathbf{1}^T \mathbf{b}_i^{(t)})^2} \\ \|\mathbf{A}^T \boldsymbol{\gamma}^{(t)}\|_2^2 &= \sum_{i=1}^p \left\| \sum_{l_1=i} \gamma_{l_1}^{(t)} - \sum_{l_2=i} \gamma_{l_2}^{(t)} \right\|_2^2 + d \sum_{i=1}^p (\phi_j^{(t)})^2\end{aligned}$$

which justifies our selection.

C.2 Dual Objective for AMA for Squared Error Loss

This computation will be similar as described in Chi and Lange (2015). By adjusting the constants δ_j , we find that the dual for the optimization problem is given by

$$\begin{aligned}\mathcal{D}(\boldsymbol{\Gamma}) &= \inf_{\mathbf{B}, \mathbf{V}} \mathcal{L}_0(\mathbf{B}, \mathbf{V}, \boldsymbol{\Gamma}) \\ &= -\frac{1}{2} \sum_{j=1}^p \frac{1}{\delta_j} \left\| \sum_{l_1=j} \gamma_{l_1} - \sum_{l_2=j} \gamma_{l_2} \right\|_2^2 - \sum_{l \in \mathcal{E}} \langle \gamma_l, \hat{\boldsymbol{\pi}}_{l_1} - \hat{\boldsymbol{\pi}}_{l_2} \rangle - \sum_{l \in \mathcal{E}} \mathcal{H}_{C_l}(\gamma_l),\end{aligned}$$

where $\mathcal{H}_A(\mathbf{x}) = 0$ if $\mathbf{x} \in A$, ∞ otherwise. Note that we don't need Φ here, as for DPD with $\mu = 1$, the optimum solution is a valid probability vector.