MISSING DATA ANALYSIS, COMBINATORIAL MODEL SELECTION AND STRUCTURE LEARNING

CHUN FUNG KWOK
ORCID: 0000-0002-0716-3879

SUBMITTED IN TOTAL FULFILMENT OF THE REQUIREMENTS OF THE DEGREE OF DOCTOR OF PHILOSOPHY

JUNE 2019

SCHOOL OF MATHEMATICS AND STATISTICS
THE UNIVERSITY OF MELBOURNE
Abstract

This thesis examines three problems in statistics: the missing data problem in the context of extracting trends from time series data, the combinatorial model selection problem in regression analysis, and the structure learning problem in graphical modelling / system identification.

The goal of the first problem is to study how uncertainty in the missing data affects trend extraction. This work derives an analytical bound to characterise the error of the estimated trend in terms of the error of the imputation. It works for any imputation method and various trend-extraction methods, including a large subclass of linear filters and the Seasonal-Trend decomposition based on Loess (STL).

The second problem is to tackle the combinatorial complexity which arises from the best-subset selection in regression analysis. Given \( p \) variables, a model can be formed by taking a subset of the variables, and the total number of models is \( 2^p \). This work shows that if a hierarchical structure can be established on the model space, then the proposed algorithm, Gibbs Stochastic Search (GSS), can recover the true model with probability one in the limit and high probability with finite samples. The core idea is that when a hierarchical structure exists, every evaluation of a wrong model would give information about the correct model. By aggregating these information, one may recover the correct model without exhausting the model space. As an extension, parallelisation of the algorithm is also considered.

The third problem is about inferring from data the systemic relationship between a set of variables. This work proposes a flexible class of multivariate distributions in a form of a directed acyclic graphical model, which uses a graph and models each node conditioning on the rest using a Generalised Linear Model (GLM), and it shows that while the number of possible graphs is \( \Omega \left( 2^{\binom{p}{2}} \right) \), a hierarchical structure exists and the GSS algorithm applies. Hence, a systemic relationship may be recovered from the data. Other applications like imputing missing data and simulating data with complex covariance structure are also investigated.
Declaration

This is to certify that:

1. the thesis comprises only my original work towards the PhD except where indicated in the Preface,

2. due acknowledgement has been made in the text to all other material used,

3. the thesis is fewer than 100,000 words in length, exclusive of tables, maps, bibliographies and appendices.

Chun Fung Kwok
30 June 2019
Preface

This thesis is based solely on the research conducted during my Ph.D candidature at the School of Mathematics and Statistics, the University of Melbourne, under the supervision of Dr. Guoqi Qian \(^1\) and Dr. Yuriy Kuleshov \(^2\)\(^3\)\(^4\).

This research has been supported by the Melbourne International Fee Remission Scholarship (MIFRS) and the Melbourne International Research Scholarship (MIRS).

Acknowledgements

First, I would like to express my sincere gratitude to late Prof. Mark Joshi, my honour supervisor and mentor, for his teaching and guidance. Through the many conversations we had about education, mathematics and research, he has inspired and motivated me to think deeply and take a broad perspective toward matters.

Next, I would like to thank my Ph.D supervisor Guoqi Qian for, first and foremost, having faith in me and giving me the freedom to explore many areas - both inside and outside statistics - which has made the Ph.D study the highlight of my academic journey so far. I am also grateful for his sharing of insights and the unlimited support he provided. It has been a great pleasure to work with him.

I would also like to thank my Ph.D co-supervisor Yuriy Kuleshov for his patience, encouragement and knowledge. He always encouraged me to aim high and do the best I can, and his sharing of project management techniques in particular helped me enormously in managing my research and becoming a better researcher.

Apart from my supervisors, I would like to thank Alex Zarebski, Felix Leung, Jasper Wijnands and Kate Saunders for the many stimulating discussions we had and the contrasting viewpoints they provided which freed me from my own bias. A special mention goes to Ada Yan, Clare Miller, Hilary Hunt, Tiffany Leung, Puxue Qiao, Yao-ban Chan, Zhendong Huang for keeping me accompanied and entertained throughout my Ph.D study.

Finally, I would like to extend my thanks to Aihua Xia, Ben Hess, Howard Bondell, James McCaw, Jan De Gier, Kirsten Hoak, Kostya Borovkov, Ling Shi and Richard Huggins from the School of Mathematics and Statistics for their

---

\(^1\)School of Mathematics and Statistics, The University of Melbourne, Parkville, Australia
\(^2\)National Climate Centre, Australian Bureau of Meteorology, Docklands, Australia
\(^3\)School of Science, Royal Melbourne Institute of Technology (RMIT) University, Melbourne, Australia
\(^4\)Faculty of Science, Engineering and Technology, Swinburne University of Technology, Melbourne, Australia
support, and to the University of Melbourne for the scholarship, without which this study may not be possible.
To the person who once told me “Do. Or do not. There is no try.”
Contents

Abstract ................................................................. ii
Preface and Acknowledgements ........................................ iv
List of Tables ........................................................... x
List of Figures ............................................................. xi

1 Introduction ......................................................... 1
   1.1 Problems outline and Motivation ........................... 1
   1.2 The Problems, Approaches and Contributions ........... 2
      1.2.1 Time series missing data analysis ..................... 2
      1.2.2 Combinatorial Model Selection ....................... 5
      1.2.3 Structure Learning .................................... 7
   1.3 Preliminaries ................................................ 9
      1.3.1 Missing data analysis ................................ 9
      1.3.2 Trend extraction ...................................... 11
      1.3.3 Exponential-family based modelling ................... 13
      1.3.4 Monte Carlo Optimisation ............................. 15
      1.3.5 Structure learning and Graphical modelling ........ 19

2 Error analysis: estimating trend with missing data .......... 22
   2.1 Introduction .................................................. 22
   2.2 Main results .................................................. 24
      2.2.1 The mathematical formulation ....................... 24
      2.2.2 Bounds evaluation .................................. 26
   2.3 Estimating imputation error ................................ 31
      2.3.1 Theoretical approach ................................ 31
      2.3.2 Data-driven approach ................................. 34
   2.4 Simulation studies .......................................... 34
   2.5 Application .................................................. 37
   2.6 Conclusion ................................................... 38
   2.7 Appendix ..................................................... 39
# List of Tables

1.1  Comparison of multivariate models ........................................ 21

2.1  Examples of trend extraction methods ................................. 26

2.2  List of symbols .................................................................. 30

3.1  Examples of encoding of models ........................................... 65

3.2  Model selection results for the independent-covariates case .... 70

3.3  Model selection results for the dependent-covariates case ...... 71

3.4  Model selection results for the mixed-simulated case ............. 72

4.1  SGLM numerical results ..................................................... 99
# List of Figures

1.1 Comparison of search algorithms ........................................... 19

2.1 Schematic representation of STL ........................................... 29

2.2 Verification and comparison of upper bounds of trend error under setting 1 ................................................................. 35

2.3 Verification and comparison of upper bounds of trend error under setting 2 ................................................................. 36

2.4 An illustration of the Novolazaravskaja time series before and after imputation ................................................................. 38

2.5 Antarctica temperature change measured at the Novolazaravskaja station ................................................................. 39

2.6 An illustration of the proof strategy ........................................... 43

3.1 Data simulation procedure ..................................................... 67

3.2 A roadmap for proving Theorems 10 and 11 ................................. 74

4.1 An example of a Directed Acyclic Graph ................................... 88

4.2 An example of correct models and incorrect models ......................... 94

4.3 Distribution of proportion of recovered edges ............................... 100

4.4 Simulated data from the Lorenz system .................................... 102

4.5 Comparison of the correlation matrices of the original data and the simulated data ................................................................. 105
Chapter 1

Introduction

1.1 Problems outline and Motivation

This work draws motivation from both applied and theoretical aspects of statistics and it is concerned with three problems. The first is about extracting trend from time-series data when some data are missing, motivated by joint-work with the Bureau of Meteorology, Australia. The research interest is on the reliability of the estimated trend when there is significant amount of missing data, and the desired research outcome is to be able to answer questions akin to “if we have X amount of missing data, and extract a trend using method Y, how much uncertainty is there in the trend?”; this subsequently provides a basis for making policy recommendation.

The second is the problem of combinatorial model selection. In regression analysis, given a set of $p$ predictors, the method of best-subset selection evaluates all the possible candidate models (formed by taking subsets of the predictors) and selects the one with the best goodness-of-fit score. However, as the number of subsets in a set of $p$ elements is $2^p$, it is often computationally infeasible to use in practice, e.g. in genetic application, $p$ is often in the order of 10000, and the number of candidate models is about $2^{10000} \approx 10^{3000}$ (in comparison, the number of atoms in the universe is estimated to be around $10^{80}$). Model selection lies at the heart of real-world modelling problems, and improving the selection leads to finding models with better explanatory and/or predictive power. It is also of theoretical interest to find new principles to tackle combinatorial complexity, as this technical difficulty is common to many (NP-)hard problems.

The third is the problem of structure learning. Given some data, structure learning is about inferring a systemic relationship among variables. Traditionally, scientists study a phenomenon by proposing / refining hypotheses and validating
them with data iteratively until a satisfactory result is obtained; data are used to quantify the phenomenon and reject incorrect theories. However, when the system of interest has complex dynamics and interactions, it can be difficult to formulate a hypothesis. As the field of modern statistics advances, and more computing facilities and data become available, an alternative has emerged. Structural learning techniques use data to generate hypotheses, which are then examined by scientists. In this regime, the roles of data and scientists are reversed, and data are turned to knowledge via the means of structure learning. This helps us uncover the mechanism behind complex phenomena and develop a better understanding of the world.

While the three problems may at first sight seem unrelated to each other, they actually share a common theme - which is to characterise how much one can recover from data the underlying dynamics. For the time series problem, when some data are missing, perfect recovery is generally not possible, and this work addresses to what extent the trend can be recovered by providing a bound for the error. For the problems of model selection and structure learning, this work provides statistical guarantees to the selection of models and graphs. The result suggests recovery is always possible in the limit and it occurs with high probability in the finite-sample case. A detailed account of the problems, the approaches to solve them, and the contributions made by this work are presented in the next section.

1.2 The Problems, Approaches and Contributions

1.2.1 Time series missing data analysis

Trend extraction is a common task in time series analysis. Its goal is to recover the underlying signal from noisy data, which may then be used to attribute phenomena to their potential causes or to generate forecast about the future. For example, in climate modelling, it is of interest to recover the centurial change of the sea-level temperature from daily measurements, find out if the change is associated with human activities, and whether the human race is perhaps heading for disaster. Another example is the modelling of influenza outbreak, one may want to estimate the actual number of infection in the population from the reported cases. This information is then used to inform the decision about the allocation of medical resources.

In practice, data often contain missing parts, and it is important to know if the trend extracted under such situation is still valid. Intuitively, if most data are
missing, then the trend is likely to be unreliable, and if little data are missing, then it is the opposite. Making precise the connection between the missing-ness of the data and the quality of the extracted trend is the central question of this part of the work.

To perform trend extraction when there are missing data, one first handles the missing data, then performs the trend extractions. In the inferential context, the de facto standard for handling missing data is the likelihood-based framework established by Rubin (1976, 1987) and Little and Rubin (1987) with refinements over the years, see for instance Rubin (1996) for clarifications of theoretical results, Horton and Lipsitz (2001); Graham et al. (2007) for practical concerns and Horton and Kleinman (2007), Honaker and King (2010) for implementations. This work deals with the inferential problem, but as a side note, in the predictive context, there are also alternatives like matrix factorisation/completion (Candès and Recht, 2009) with notable applications like collaborative filtering (Candès and Tao, 2010) and compressed sensing (Candes and Plan, 2010).

Within the likelihood-based framework, generally one makes some assumptions about the missing data-points and attempts to recover the points via imputation. The imputation could be explicit in the sense that the missing point is filled with an actual value or implicit via the Expectation-Maximisation (EM) algorithm (Dempster et al., 1977; Ghahramani and Jordan, 1994; Lauritzen, 1995). Depending on the assumptions made about the missing point, the resulting implementation may be distribution-based or regression-based (Lee and Carlin, 2010), parametric or non-parametric (Schafer, 1999).

Once the missing data are handled, the trend extraction may proceed. The major trend extraction methods may be roughly divided into two categories, the smoothing-based category and the decomposition category. Smoothing-based methods identifies a trend as a gradual (smooth) change of a process and recovers it by applying smoothing operators to the data. Decomposition methods break the data into different components based on some mathematical properties (e.g. frequency, variations) and identify one of the components as the trend.

Research along the smoothing-based path has yielded fruitful results. Some well known methods are Henderson filters (Henderson, 1916), the classical decomposition (dating back to 1920s (Hyndman and Athanasopoulos, 2018)), Seasonal-Trend decomposition based on Loess (STL) (Cleveland et al., 1990), Hodrick-Prescott filters (Hodrick and Prescott, 1997) and X12-ARIMA (Findley et al., 1998), which was updated to X-13-ARIMA in 2013. Most of these methods are referred to as linear filters since they give a set of weights that are applied to the data as a smoothing / averaging operator to produce the underlying trend,
and they differ mainly in the class of functions used for fitting and the smoothness criterion. For non-linear filters, various methods have been proposed, see for example, the optimal order statistic filter (Bovik et al., 1983), the stack filters (Gabbouj et al., 1992) and the median filters (Wen and Zeng, 1999).

Compared to the smoothing-based approach, the decomposition approach is relatively new. Two methods, originating from the field of signal processing, have made their ways to the field of time series analysis. They are the Singular Spectrum Analysis (SSA) and the Empirical Mode Decomposition (EMD). SSA is based on the idea of factorisation: it performs the Singular Value Decomposition to the covariance matrix of trajectory matrices to extract a trend from the data Hassani (2007). SSA has been actively developed (Schoellhamer, 2001; Moskvina and Zhigljavsky, 2003; Kondrashov and Ghil, 2006; Hassani et al., 2011; Mohammad and Nishida, 2011; Shen et al., 2015) and applied to various kinds of data, e.g. climate data (Ghil and Vautard, 1991; Ghil, 2002); financial data (Hassani and Thomakos, 2010; Ghodsi and Yarmohammadi, 2014) and geophysical data (Kondrashov and Ghil, 2006).

On the other hand, EMD is based on the idea of orthogonal projection: the method decomposes signals into finite, nearly-orthogonal components that admit Hilbert transforms (Huang et al., 1998). While the method works in the time domain, it can also be interpreted as a special case of Wavelet methods which work in the frequency domain (Flandrin et al., 2004). As EMD adapts well to nonlinear and non-stationary data, it has been widely studied and applied, as seen in Echeverría et al. (2001); Battista et al. (2007) and Zhang et al. (2008). For a comprehensive review of other trend extraction methods, readers are referred to Alexandrov et al. (2012).

This work considers the explicit imputation methods for handling the missing data and the linear filters / STL for extracting the trend. It assumes that the underlying trend is smooth without any additional distributional assumption. This is motivated by that in practice, policy maker often wants to know (among many things) what can be said about the problem under minimal amount of assumption. The main contribution is to provide (hard) analytic bounds on the error of the estimated trend in terms of error of the imputation, which work for any explicit imputation methods, a large class of linear filters and the STL algorithm. The bounds can then be used to answer (i) Does the difference (in the missing data) make any difference (in the trend estimates)? (ii) How much uncertainty is there in the estimated trend when there are missing data? (iii) To what extent are imputation methods able to recover missing data in the context of trend extraction?
1.2.2 Combinatorial Model Selection

Model selection is the process of selecting a particular model out of a range of models for describing the data. Combinatorial Model Selection refers to when the number of models is combinatorially large. The key example examined in this part of the work is the best-subset selection in the context of linear regression. To avoid potential confusion due to the conventions used in other areas, it is clarified that linear regression in this work refers to regression with linear estimation method, i.e. the regression function is linear in the parameters, and there is no restriction in using non-linear functions/features of predictors.

Given a set of $p$ features, a model can be formed by taking a subset of them, and this gives rise to $2^p$ models. There are many real-world instances where $p$ is large enough that it is not feasible to apply best-subset selection. Traditionally, one may consider alternative selection methods like

- stepwise selection, e.g. forward selection and backward selection, which greedily adds (in the forward case, or removes in the backward case) features one at a time until there is no statistically significant predictor to include in the test-based evaluation, or until there is no improvement in the cross-validation evaluation,

- shrinkage methods, e.g. Ridge (Hoerl and Kennard, 1970), Lasso (Tibshirani, 1996), SCAD (Fan and Li, 2001), LARS (Efron et al., 2004), Elastic-net (Zou and Hastie, 2005) and the Dantzig selector (Candes et al., 2007), which impose penalty on model complexity to introduce bias towards simpler models, and

- information criterion, e.g. AIC (Akaike, 1973), BIC (Schwarz et al., 1978), MDL (Rissanen, 1978), DIC (Spiegelhalter et al., 2002) and FIC (Claeskens and Hjort, 2003), which applies finite sample correction to the Kullback-Leibler divergence calculation.

Comparing to the other methods, the best-subset selection is unintelligent in the sense that all it does is to evaluate all possibilities and pick the best one out of it. But this assumption-agnostic characteristic is exactly what makes it a good (meta-)selection method. In fact, we will see later once we establish the combinatorial principle for the best-subset selection, it can be adapted to another combinatorial problem (structural learning) easily.

To tackle combinatorial complexity (within a reasonable time-frame), some compromises need to be made, and there are mainly two directions to go. The first is to relax optimality to sub-optimality. For example, consider running a
local optimum finder with multiple restart. Under suitable conditions, the global optimum can be found with guarantee as the number of restart goes to infinity. But in the case of finite number of restart, one only has the guarantee of local-optimality.

The second is to relax certainty to sub-certainty. Here is a simple example to illustrate the idea. Consider that there are 100 people applying for 1 job position. To ensure that the company hires from the best ten candidates, the company must interview at least 91 people, because in the worst case the best 10 candidates may only come in the last 10 interviews. However, if one does not require 100% certainty, then one may consider the strategy of randomly interviewing 30 people and hiring the best person out of them. This strategy gives about 98% chance of hiring from the best 10 candidates. In other words, one may trade 2% of certainty to cut off more than two-third of the work. Returning to tackling combinatorial complexity, relaxing certainty usually involves using an probabilistic algorithm that finds the optimal solution with probability 1 in the limit, but only with high probability (i.e. no guarantee) in finite time.

As a side note, there has been many interesting developments along this line of research, ranging from fundamental algorithms like quicksort and verification of matrix multiplication (with improvement of runtime from $O(n^2)$ to $O(n \log n)$ and $O(n^3)$ to $O(n^2)$ respectively) to some NP-hard problems (with probabilistic P algorithms). For more detail, readers are referred to Mitzenmacher and Upfal (2005) and Vidyasagar and Blondel (2001).

This work follows the relaxing-certainty-to-sub-certainty direction. It recasts an optimisation problem into a density estimation / sampling problem, and proposes a Markov-Chain-Monte-Carlo (MCMC) algorithm to solve the problem. The algorithm, referred to as Stochastic Search using Gibbs sampler, finds the global optimum by first searching the space of configurations randomly following the Gibbs (a.k.a. Boltzmann) distribution, and then inferring from the search trajectory about the location of the global optimum.

The main contribution of this part of the work is to provide a hierarchical principle to tackle combinatorial complexity and an algorithm that employs the principle. It is shown that if a hierarchical structure can be arranged out of the model space, then the algorithm would find the global optimum with probability one in the limit (and with high probability in finite time given sufficiently large data and number of MCMC iterations). In other words, the algorithm is able to select the best model out of the $2^p$ possibilities. The core idea is that when a hierarchical structure exists, every evaluation of a wrong model would say something about the true model, and by aggregating these information along the entire
search trajectory, one may be able to recover the true model. The proof of the selec-
tion consistency also gives hints of how far this idea may be generalised. Loosely
speaking, the idea applies whenever a model space has a hierarchical structure and
a score function that respects the ordering of the levels, i.e. monotonicity in the
structure must be maintained in the scores.

1.2.3 Structure Learning

Seeking mechanisms behind phenomenons has always been a major way how scien-
tists develop understanding of the world. Currently, there are still many important
applications yet to be solved; to name a few, we still know relatively little about
how our brains function, how cells coordinate and communicate, and what genes
and diseases are linked. Structure learning is a tool for discovering systemic re-
lationship between variables. The structure is encoded using a graphical model,
which is a probabilistic model specified with a graph. A graph consists of nodes
and edges, where nodes represent variables and an edge between two nodes rep-
resents the dependence between two variables. When the edges are directed, the
model is called directed graphical model or Bayesian Network (BN); when the
edges are undirected, the model is called undirected graphical model or Markov
Random Field (MRF).

There are three major ways to learn a structure from data. The first is to use
domain knowledge to determine the structure. For a particular problem domain,
sometimes there is a natural graphical model well suited to the problem, e.g. the
Ising model for image analysis (Geman and Geman, 1984), the Hidden Markov
Model (HMM) for speech recognition (Rabiner, 1989) and the Latent Dirichlet
Allocation (LDA) for document modelling and text classification (Blei et al., 2003).
This approach often leads to reliably good performance and interpretable results.

The second way is to use constraint-based algorithms. The seminal algorithms
are the Inductive Causation (IC) algorithm (Pearl and Verma, 1992), the SGS
algorithm and the PC algorithm (Spirtes and Glymour, 1991). These algorithms
apply conditional independence tests iteratively to pairs of nodes to recover the
graph structure. Some common conditional independence tests used in these algo-
rithms include the mutual information, Monte Carlo permutation test, Pearson’s
Chi-squared test, fast mutual information and Fisher’s Z test (Scutari et al., 2010).
There has been active development along this path and many variations have been
proposed; some are to improve efficiency (Silverstein et al., 2000; Tsamardinos
et al., 2003; Le et al., 2016), some are to reduce the false discovery (due to the large
number of conditional independence test performed) (Li and Wang, 2009; Strobl
et al., 2016), and others to handle latent variables, e.g. FCI in Spirtes et al. (2000, p.144-145) and RFCI in Colombo et al. (2012) and order-dependence (Colombo and Maathuis, 2014). For an extensive review and comparison of constraint-based algorithms, readers are referred to Yu et al. (2016); Singh et al. (2018).

The third way is to use score-based algorithms. Score-based algorithms search through the space of structures, evaluate each structure with some score function, and then select the one with the best score. The search strategy usually follows those used in combinatorial optimisation problem, see for instance ant colony optimisation used in De Campos et al. (2002), simulated annealing in Janžura and Nielsen (2006), tabu search in Scutari et al. (2010), evolutionary algorithms in LarrañAga et al. (2013) and particle swarm optimization in Gheisari and Meybodi (2016). On the other hand, there are also many choices for the scoring function, some of the well-established ones are the mutual information (Chow and Liu, 1968), the conditional entropy (Herskovits et al., 1990), the Minimum Description Length (MDL) (Suzuki, 1993) and the Bayesian Dirichlet metric (Heckerman et al., 1995).

This work takes a viewpoint of structure learning similar to Chow and Liu (1968) in the sense that ultimately it aims to find the best approximation (within a class of distributions) to a target multivariate distribution. A directed (acyclic) graphical model is proposed, where each node given all the other nodes is modelled as a Generalised Linear Model (GLM), and it is referred to as the Structural GLM (SGLM). Note that this is different to the Graphical models via GLM (GGLM) by Yang et al. (2012) and Mixed graphical model (MGLM) by Yang et al. (2014). The key difference to Yang et al. (2012) is that GGLM assumes the conditional distributions for all the nodes are from the same exponential-family distribution; this allows global simplification to give a closed-form multivariate densities. In contrast, our SGLM uses different exponential-family distributions for different nodes, and in general simplification is not possible. Furthermore, while SGLM is distributionally equivalent to the clique-2 MGLM, it is constructed using a directed graph (instead of the undirected by MGLM), and this leads to different generalisation and structure learning algorithm. More detailed discussion will be presented in the latter chapter.

This work uses the score-based approach to learn the graph structure, where the penalised likelihood is used as the score function and the Gibbs Stochastic Search is used as the search strategy. The major challenge here is that given $p$ nodes, the number of directed acyclic graphs grows exponentially, at a rate between $2^{\binom{p}{2}}$ and $3^{\binom{p}{2}}$. So even for a moderate $p = 100$, the search space would have more than $2^{\binom{100}{2}} = 2^{4950} \gg 10^{4000}$ models to evaluate, which is too large.
to search through exhaustively. However, it will be shown that a hierarchical structure can be established out of the model space so that the Stochastic Search using Gibbs Sampler would apply.

In summary, the contribution of this part of the work is to provide a new flexible class of multivariate distribution and a corresponding structural inference algorithm with proven identification consistency. This class of distributions supports mixture of discrete and continuous variables, and it serves well as approximation to general joint probability distribution. In addition to structure discovery, it may also be used to impute missing data in multivariate dataset, or simulate realistic high-dimensional non-Gaussian data with complex covariance structure.

1.3 Preliminaries

In this section, we review methods that this thesis is based on and methods that will serve as benchmarks in the latter chapters. Readers who are well-acquainted with the following topics may wish to skip to the end of each topic and read the comments about how the topic relates to this work.

1.3.1 Missing data analysis

When some data are missing in the dataset, is it valid to perform statistical inference as usual? To answer this rigorously, Rubin (1976); Little and Rubin (1987); Glynn et al. (1986, 1993) developed a likelihood-based framework – which later became the standard in the statistics community – that gives explicit conditions about when missing-data mechanism can be ignored. The key formulation is given below.

The first step is to recognise that successfully observing a variable is itself an observation. Suppose we have \( n \) datapoints, \( Y = \{Y_1, Y_2, ..., Y_n\} \). For \( i = 1, 2, ..., n \), associate each \( Y_i \) with \( M_i \) and write \( M = \{M_1, M_2, ..., M_n\} \), where \( M_i \) is the random variable such that \( M_i = 0 \) if the \( i \)-th data-point is observed, and \( M_i = 1 \) if the data-point is missing, then the likelihood of the data is given by

\[
L(\theta, \phi | Y, M) = f(Y, M | \theta, \phi),
\]

where \( \theta \) and \( \phi \) are the (vector) parameters that relate to \( Y \) and \( M \) respectively. When there are no missing data, i.e. \( M_i = 0 \) for all \( i = 1, 2, ..., n \), then as one would expect, \( f(Y, M | \theta, \phi) \) reduces to the usual case of \( f(Y | \theta, \phi) \).
Assuming \( Y \) and \( M \) have independent parameters, the joint density can be factorised as:

\[
f(Y, M | \theta, \phi) = f(Y | \theta) f(M | Y, \phi). \tag{1.1}
\]

Next, breaking \( Y \) into two components, the observed data \( Y_{\text{obs}} \) and the missing data \( Y_{\text{mis}} \), and integrating out \( Y_{\text{mis}} \), we have

\[
f(Y_{\text{obs}}, M | \theta, \phi) = \int f(Y_{\text{obs}}, Y_{\text{mis}} | \theta) f(M | Y_{\text{obs}}, Y_{\text{mis}}, \phi) \, dY_{\text{mis}}. \tag{1.2}
\]

If the missing-data mechanism does not depend on the missing values \( Y_{\text{mis}} \), i.e. \( f(M | Y_{\text{obs}}, Y_{\text{mis}}, \phi) = f(M | Y_{\text{obs}}, \phi) \), then the expression can be simplified to

\[
f(Y_{\text{obs}}, M | \theta, \phi) = \int f(Y_{\text{obs}}, Y_{\text{mis}} | \theta) \, dY_{\text{mis}} \cdot f(M | Y_{\text{obs}}, \phi)
= f(Y_{\text{obs}} | \theta) \cdot f(M | Y_{\text{obs}}, \phi)
\]

Examining the last equation, we observe that maximising the likelihood \( L(\theta, \phi | Y_{\text{obs}}, M) = f(Y_{\text{obs}}, M | \theta, \phi) \) (on the left) w.r.t. \( \theta \) requires only maximising the term \( f(Y_{\text{obs}} | \theta) \) (on the right). In other words, \( f(M | Y_{\text{obs}}, \phi) \) plays no role in the inference of \( \theta \). In such a case, the missing-data mechanism is called ignorable. We distinguish the following two cases. The first case is

\[
f(M | Y, \phi) = f(M | \phi), \quad \forall Y, \phi,
\]

referred to as the Missing-Completely-At-Random (MCAR) mechanism. And the second case is

\[
f(M | Y, \phi) = f(M | Y_{\text{obs}}, \phi), \quad \forall Y_{\text{mis}}, \phi,
\]

referred to as the Missing-At-Random (MAR) mechanism.

If the missing-data mechanism is not MCAR or MAR, then it is classified as Missing Not At Random (MNAR) in which case equation (1.2) cannot be simplified. To handle MNAR, one has to explicitly model the missing data mechanism \( f(M | Y_{\text{obs}}, Y_{\text{mis}}) \). Starting with \( f(Y, M | \theta, \phi) \) in equation (1.1) again, there are two ways to proceed:

1. \( f(Y, M | \theta, \phi) = f(M | Y, \phi) f(Y | \theta) \)
2. \( f(Y, M | \theta, \phi) = f(Y | M, \theta) f(M | \phi) \)

The first equation can be interpreted as first sample the data \( Y \), then based on their values, probabilistically “select” a subset to be observed. Hence, this model
is referred to as the selection model. The second equation can be interpreted as first sampling a missing pattern, then based on the pattern, sample the data $Y$. This model is called the pattern-mixture model.

If we break $M = (M_1, M_2)$ where $M_1$ corresponds to the selection sampling given a fixed missing pattern and $M_2$ corresponds to the missing pattern sampling, then one can unify and generalise the previous two cases by specifying the joint distribution of $Y$ and $M$ to be of the form

$$f(Y, M_1, M_2 | \theta, \phi) = f(M_1 | M_2, Y, \phi_1) f(Y | M_2, \theta) f(M_2 | \phi_2).$$

It is easy to see that choosing a degenerate distribution for $f(M_2 | \phi_2)$ would give the special case 1 in the above, and choosing a degenerate distribution for $f(M_1 | M_2, Y, \phi_1)$ would give the special case 2 in the above.

**Relation to this work**

The analysis presented in this work uses imputation to handle the missing data, and many existing imputation methods assume the likelihood framework, and in particular the MCAR or MAR assumptions.

**1.3.2 Trend extraction**

In time series analysis, trend is generally represented as a smooth, slow changing features of the time series (Kendall and Ord, 1990, p.27). As the defining feature, the smoothness of the trend naturally leads one to consider polynomial for modelling the trend. This is traditionally done by fitting a polynomial locally at the point of interest using the surrounding points. To illustrate, suppose we are interested in recovering the trend value at $y_k$, and we consider fitting a third order polynomial at time $k$ using the $2m + 1$ points surrounding $k$, i.e. $(y_{k-m}, ..., y_k, ..., y_{k+m})$. Suppose $m = 3$, then the polynomial is given by

$$y_t = a_0 + a_1(t - k) + a_2(t - k)^2 + a_3(t - k)^3, \quad t = k - m, ..., k + m.$$  

(Note that the quantity of interest is $\hat{y}_k = a_0$) Solving the system of linear equations gives

$$a_0 = \frac{1}{21}(-1, 3, 6, 7, 6, 3, -2) \cdot (y_{k-3}, y_{k-2}, y_{k-1}, y_k, y_{k+1}, y_{k+2}, y_{k+3}), \quad (1.3)$$

where $\cdot$ is the dot product (Kendall and Ord, 1990, p.29).
The expression (1.3) says that the fitted value is an weighted average of the surrounding points. This generalises to the concept of linear filter, which is a function \( S : \mathbb{R}^p \times \mathbb{R}^p \to \mathbb{R} \) such that given a data input \( y = (y_1, y_2, ..., y_p) \) and a set of weights \( w = (w_1, w_2, ..., w_p) \), it returns the weighted average \( S_w(y) = \sum_{i=1}^p w_i y_i \). This produces a class of smoothing operators wider than local polynomials; some examples in this class are the Henderson filter \( \text{Henderson, 1916} \), Hodrick-Prescott filter \( \text{Hodrick and Prescott, 1997} \), Kernel smoothers \( \text{(REF)} \) and Loess \( \text{Cleveland, 1979} \).

**Loess**

Locally weighted regression \( \text{(Cleveland, 1979)} \), a.k.a. Loess, is a nonparametric method in regression analysis. It models the dependent variable as a smooth function of the independent variables, and the function is estimated by fitting the data locally with polynomials. Specifically, Loess assumes a data generating process of

\[
y_i = f(x_i) + \epsilon_i, \quad i = 1, 2, ..., N,
\]

where \( x_i, y_i \) are observations of the dependent and independent variables respectively, \( N \) is the total number of observations and \( \epsilon_i \) are independent normal random variables with mean 0 and variance \( \sigma^2 \). As the functional relationship \( f \) is assumed to be smooth, it justifies the use of Taylor’s theorem and provides grounds for approximating \( f \) locally by polynomials.

Loess approximates the functional relationship \( f \) by fitting a polynomial locally at each point \( x \) (in the domain of \( f \)) using points in the neighbourhood of \( x \). The fitting uses weighted least square (WLS) regression. Overall, three quantities are needed in this procedure, the degree of polynomials to be fit, the size of the neighbourhood and the weights for performing the WLS regression. For the degree of polynomials to be fit, first-degree or second-degree polynomials are commonly used and are sufficient as long as the functional relationship is not too erratic. Quadratic fitting is generally preferred over linear fitting near extrema \( \text{(Fan and Gijbels, 1996, p.77)} \). Alternatively, the degree of polynomials can be chosen using M-plots as suggested in Cleveland et al. \( \text{(1988)} \). For the neighbourhood size, as it directly controls the smoothness of the estimates, the choice should be made based on the research context. But in cases where one wants to avoid subjectivity, the neighbourhood size can also be chosen through data driven techniques like cross validation. Lastly, for the weights for the WLS regression, they are specified for each point in the neighbourhood of the point to be fit using the tricube weight.
function (Cleveland, 1979),

\[
W(u) = \begin{cases} 
(1-u^3)^3 & \text{for } 0 \leq u < 1, \\
0 & \text{otherwise.}
\end{cases}
\]

Concretely, suppose we have \( N \) data points, the degree of polynomial to be fit is \( q \) and the size of the neighbourhood is \( m \), and we want to fit a polynomial locally at the point \((x_1, y_1)\). First we identify the \( m \) data points nearest to \((x_1, y_1)\), denoted by \((a_1, b_1), (a_2, b_2), \ldots, (a_m, b_m)\). Next, to each of these points we assign a weight,

\[
v_i(x_1) = W \left( \frac{|a_i - x_1|}{\max_{i=1,2,\ldots,m} |a_i - x_1|} \right), \quad i = 1, \ldots, m
\]

where \( W(\cdot) \) is the tricube weight function. Then we fit a degree-\( q \) polynomial, denoted by \( p_d(x) \), to these points using weighted least square regression. Finally, the fitted value is given by \( p_d(x_1) \).

**Relation to this work**

The analytical result in Chapter 2 applies to a subclass of the linear filters. Also, Loess in particular will be used as a concrete example to demonstrate consistency with the theoretical result.

### 1.3.3 Exponential-family based modelling

**The exponential family of distributions**

Nelder and Wedderburn (1972) proposed to use the exponential family distribution to unify the common distributions so that the same algorithm may be used to perform parameter estimation for the corresponding regression problem. A random variable \( Y \) follows the exponential family of distributions if its probability density / mass function can be written as

\[
f(y; \theta, \phi) = \exp \left( \frac{y\theta - b(\theta)}{a(\phi)} + c(y, \phi) \right)
\]

where \( \theta \) is the canonical parameter, \( \phi \) is the scale parameter, and \( b(\cdot), a(\cdot), c(\cdot) \) are functions. In addition, the mean and variance of \( Y \) is given by

\[
E(Y) = b'(\theta), \quad Var(Y) = b''(\theta)a(\phi)
\]
In addition to providing computational convenience, the exponential family of distribution itself also has theoretical significance. The Pitman-Koopman-Darmois Theorem (Pitman, 1936; Koopman, 1936; Darmois, 1935) states that under IID sampling scheme, the exponential family is the only family of (continuous) probability distribution with fixed support that has a sufficient statistics of which the dimension stays constant as the sample size $n$ increases.

We further clarify that the exponential family of distribution used in this work follows the one as given in Casella and Berger (2002, p.137-138) and Brown (1986), of which the moment-generating function exists by construction.

**Generalised Linear Model (GLM)**

Given a dataset with $n$ data points $(y_i, x_i), i = 1, 2, ..., n, y_i \in \mathbb{R}, x_i \in \mathbb{R}^p$, the Generalised Linear Model assumes that

$$y_i | x_i \sim f(y_i; \theta_i, \phi_i), \quad E(y_i | x_i) = h(x_i \beta), \quad (1.4)$$

where $f$ follows the exponential-family distribution, $\beta \in \mathbb{R}^p$ and $h$ is the inverse-link function.

GLM is undoubtedly a major milestone in statistics as it has spawned many lines of researches, including model selection (Zou and Hastie, 2005; Friedman et al., 2010), likelihood theory (Wedderburn, 1974; McCullagh, 1983; Nelder and Pregibon, 1987; Godambe and Thompson, 1989; Jung, 1996), predictive modelling with ensembles (e.g. GLM with boosting in Tutz and Binder (2007) and bagging in Song et al. (2013)) and estimation with big data (e.g. stochastic gradient methods with GLM in Toulis et al. (2014), random subspaces with GLM in Song et al. (2013)). The line of research that is related to this work is the regression context where GLM has been extended to

- **Generalised Linear Mixed Model (GLMM)** (Gilmour et al., 1985) to take into account of covariance structure (commonly used in clustered data),

- **Generalised Additive Model (GAM)** (Hastie and Tibshirani, 1986) to handle non-linearity (also to Generalised Additive Mixed Model (GAMM) for covariance structure in addition),

- **Multivariate GLM (MGLM)** (Fahrmeir and Tutz, 1994) to handle data with time / space dependence, e.g. times series, panel data and spatial data, and

- **Graphical GLM (GGLM)** (Yang et al., 2012) to handle structure relationship.
Relation to this work

The second part of our work is based on the best-subset selection with the GLM. And the third part of our work continues the GLM development by proposing a graphical model (and a corresponding structure-learning algorithm) that contrasts with and goes beyond GGLM.

1.3.4 Monte Carlo Optimisation

Simulated Annealing

Simulated Annealing is an algorithm inspired by the annealing of solids for combinatorially-hard optimisation problem (Kirkpatrick et al., 1983). In real annealing, a solid is heated beyond its melting point, then gradually cooled down to reach the configuration of minimal energy. In simulated annealing, a sequence of parameters, referred to as the cooling schedule, are used to mimic the annealing process to minimise a cost function.

Denote a cost function of a configuration \( s \) by \( C(s) \). The algorithm begins with step \( i = 0 \) at a randomly initialised state \( s_i \) at a pre-specified temperature \( T_i \). Next, it samples a configuration from the neighborhood of \( s_i \). If the new state has a lower cost, move to the new state and re-iterate; if the new state has a higher cost, then move to the new state only with probability \( \exp(-\Delta_{\text{cost}}/T_i) \), where \( \Delta_{\text{cost}} = C(\text{newState}) - C(\text{currentState}) \), and re-iterate. This process continues until a stopping criterion is met (e.g. the maximum number of iterations is reached, or there is no state transition for certain number of iterations, or temperature \( T_i \) is small enough). In a nutshell, simulated annealing is a random walk being greedy when there is improvement and increasingly risk-averse when there isn’t. The pseudo-code is presented in Algorithm 1.

The condition for convergence of the global optimum were found (independently) by Geman and Geman (1984), Gidas (1985), Mitra et al. (1986), Anily and Federgruen (1987) and Hajek (1988). With minor cosmetic difference, the condition requires the cooling schedule to go to zero no faster than a large constant divided by the logarithm of the step number. Hajek (1988)’s result is the most complete, stating the necessary and sufficient condition for convergence to the global optimum. The condition requires that \( T_k = c / \log(1 + k) \) where \( c \) is greater than or equal to the depth of the deepest local minimum which is not the global minimum.

Putting aside the mathematical detail, the reason why simulated annealing converges to the global minimum is based on two key observations. The first
Algorithm 1: Simulated Annealing algorithm pseudo-code

Input: A cost function and a startState
Output: An endState that minimises the cost function

Set \( i = 0 \) and set-up temperature \( T_i \);
currentState = startState;
currentCost = cost(currentState);

while the stopping criterion is not satisfied do
    newState = perturb(currentState);
    newCost = cost(newState);
    if newCost < currentCost then
        currentState = newState;
        currentCost = newCost;
    else
        Simulate \( U \sim uniform(0, 1) \);
        \( \Delta_{cost} = newCost - currentCost \);
        if \( U < \exp(-\Delta_{cost}/T_i) \) then
            currentState = newState;
            currentCost = newCost;
        end
    end
    \( T_{i+1} = updateTemperature(T_i, i) \);
    \( i = i + 1 \);
end

endState = currentState;
return endState


development of simulated annealing in the 1990s and 2000s. For completeness sake, we briefly survey the main developments. The two main directions are speeding up the search process and adapting it to different settings.
In the former case, different jump mechanisms are investigated to achieve a faster convergence/cooling schedule. Szu and Hartley (1987) considers replacing the Gaussian transition density by a Cauchy transition density, speeding up the cooling schedule from \( T_0 / \log(k) \) to \( T_0 / k \), where \( T_0 \) is some initial temperature. Ingber (1989) considers a way to construct a transition density that achieves an exponential cooling schedule, which is even faster and works better for multi-dimensional parameter space than the Cauchy cooling schedule. Tsallis and Stariolo (1996) generalises simulated annealing to include the Gaussian transition and Cauchy transition as special cases, and uses the more general transition density to improve numerical results. Xavier-de Souza et al. (2010) considers parallel runs of simulated annealing, referred to as the coupled simulated annealing, and uses the information exchange between the runs to achieve a better optimisation efficiency. Ventresca and Tizhoosh (2007) exploits the lack of symmetry in general high-dimensional problem via the use of opposite neighbours to improve the accuracy and convergence rate.

In the latter case, much work has been done for simulated annealing with uncertainty, i.e. when the cost function can only be evaluated with noise, see for instance Gelfand and Mitter (1989); Painton and Diwekar (1995); Gutjahr and Pflug (1996); Prudius and Andradóttir (2005) and Mattila et al. (2013). The other important extension is to handle continuous domains (Vanderbilt and Louie, 1984; Press and Teukolsky, 1991) with convergence analysis (Locatelli, 2000a,b; Yang, 2000) and finite-time guarantees (Lecchini-Visintini et al., 2008). Another notable variation is simulated annealing with constraints as considered in Romeijn and Smith (1994); Wah and Chen (2000) and Hedar and Fukushima (2006).

**Genetic algorithm (GA)**

Genetic algorithm is a member of the evolutionary algorithms, a class of optimisation algorithms based on heuristics inspired by the biological evolution process. The algorithm begins with a small population of solutions, evaluates all of them, selects the best ones according to some criteria, then performs the following two operations to generate new offspring solutions:

- **Crossover** combines features from the good solutions. For example, suppose the two binary strings 111000, 101101 are two good solutions to a problem, then a possible crossover may take the first 3 entries of the first string and combine it with the last 3 entries of the second string to give an offspring of 111101.
(In the model selection context, one may use a binary string of length \( p \) to encode a regression model, where 1 at entry \( i \) refers to the inclusion of the \( i \)-th predictor variable out of the \( p \) variables.)

- **Mutation** randomly perturbs the current solution. For example, suppose the binary string 101101 is a good solution, then a mutation randomly samples a position (from 1 to 6 in this case) and flips the bit at that position (say the sample is 3, then the mutated offspring is 100101).

After crossover and mutation, a new offspring population is created. Finally, the algorithm replaces the original population by the new one and re-iterates the process until some termination criteria is satisfied (e.g. when there is no new offspring or no improvement for certain number of regenerations).

While the convergence of GA has been proved under certain settings (Eiben et al., 1990), GA is usually considered a heuristic algorithm, and much of the development is devoted to adapting it to other situations. GA originally is formulated for problems with discrete inputs - as it draws inspiration from DNA sequences - and it has been extended to handle continuous inputs (Wright, 1991; Eshelman and Schaffer, 1993; Adewuya, 1996). Other developments include adapting GA to optimisation with constraints (Michalewicz and Janikow, 1991; Orvosh and Davis, 1994; Michalewicz, 1995; Carlson and Shonkwiler, 1998), optimisation with multiple objectives (Murata and Ishibuchi, 1995; Deb et al., 2002; Konak et al., 2006), speeding up GA by parallelisation (Mühlenbein et al., 1991; Lin et al., 1994; Adeli and Kumar, 1995; Cantú-Paz, 1998), and improving the evolution mechanism for faster convergence to optimum (Rasheed and Hirsh, 2000; Nia and Alipouri, 2009).

**Relation to this work**

Simulated Annealing uses no information from the past; it only looks at the local landscape and compares the local point and the point that it jumps to. Genetic algorithm uses some structural information from the population of the same generation. It keeps the good genes, discards the bad ones and allows crossover between the good models. Gibbs Stochastic Search, as we will see later, uses information of the entire MCMC sample path to form the model selection. By aligning these methods according to the structural information used, a spectrum can be formed; this is shown in Figure 1.1. In the second part of this work, Gibbs Stochastic Search will be compared against the Simulated Annealing and the Genetic algorithm.
1.3.5 Structure learning and Graphical modelling

In the following, we review the construction of the copula model, the vine copulas model and the graphical GLM. Inference methods are omitted as they are not the focus of this work. Readers looking for details on estimation are referred to Jaworski et al. (2010), Joe and Kurowicka (2011) and Yang et al. (2012).

Copula

Copula is a multivariate distribution function of which all the marginal distributions follow the uniform distribution. The fundamental result underlying copula models is given by the Sklar’s theorem (Sklar, 1959) which states that, for every $d$-dimensional distribution function $F$ with marginal distributions $F_i, i = 1, 2, ..., d$, there exists a copula $C$ such that

$$F(x_1, x_2, ..., x_d) = C(F_1(x_1), F_2(x_2), ..., F_d(x_d)), \quad \forall \mathbf{x} = (x_1, x_2, ..., x_d) \in \mathbb{R}^d, \quad (1.5)$$

and $C$ is unique if the marginals are all continuous. As marginal distributions are relatively straightforward to estimate, the theorem reduces the estimation of the full joint density to the estimation of the dependence between the marginals.

Some well known copula functions include the Gaussian copula, the Frank copula and the Farlie–Gumbel–Morgenstern copula. These copulas were proposed pre-1980; two modern alternatives are the Extreme-Value copula and the Archimax copula, with recent development in Charpentier et al. (2014); Saminger-Platz et al. (2017); Hofert et al. (2018).

Vine copulas

Despite that copulas have brought new insights about the construction of multivariate distribution, finding a high-dimensional copula in general can be as hard as finding the joint distribution. The main difficulty lies with the need of handling high-dimensional function directly. Joe (1996) proposed a new approach later referred to as the vine copulas to circumvent the problem. Vine copulas take advantage of the graph terminology and build a copula by “stitching” together
many bivariate copulas. It is based on the pair-copula decomposition of the joint density (Bedford and Cooke, 2001):

\[
f(x_1, x_2, \ldots, x_d) = \left( \prod_{j=1}^{d-1} \prod_{i=1}^{d-j} c_{i,(i+j)|((i+1),\ldots,(i+j-1))} \right) \prod_{i=1}^{d} f_i(x_i),
\]

where \(c_{i,j|i_1,\ldots,i_k} := c_{i,j|i_1,\ldots,i_k}(F(x_1|x_{i_1},\ldots,x_{i_k}), F(x_j|x_{i_1},\ldots,x_{i_k}))\), \(c_{i,j}(\cdot, \cdot)\) is a bivariate copula density involving \(x_i\) and \(x_j\), and \(f_i(x) = \frac{dF_i(x)}{dx}, i = 1, 2, \ldots, d\). Comparing that with the expression one gets from differentiating the distribution function in (1.5)

\[
f(x_1, x_2, \ldots, x_d) = c(F_1(x_1), \ldots, F_d(x_d)) \cdot \prod_{i=1}^{d} f_i(x_i),
\]

where \(c(y_1, y_2, \ldots, y_d) = \frac{\partial C(y_1, y_2, \ldots, y_d)}{\partial y_1 \cdots \partial y_d}\), it is clear any copula can be built by taking products of bivariate copula.

**Graphical GLM**

Graphical models via Generalised Linear Model (GGLM) (Yang et al., 2012) is a graphical model where each node conditioning on the other nodes is modelled by the exponential family of distribution. Specifically, let \(X = (X_1, X_2, \ldots, X_p)\) be a \(p\)-dimensional random vector and \(G = (V, E)\) be the corresponding graph where \(V, |V| = p\) denotes the vertex set and \(E\) denotes the edge set. Then for each node \(n \in V\),

\[
P(X_s|X_{V\setminus s}) = \exp \left\{ E(X_{V\setminus s})B(X_s) + C(X_s) - \bar{D}(X_{V\setminus s}; \phi) \right\}, \quad (1.6)
\]

where \(E, B, C, \bar{D}\) are some functions.

In the special case where the joint distribution has cliques (i.e. fully connected subgraphs) of size at most two and that \(B(\cdot)\) is such that \(B(X_s) = X_s\), then the conditional distribution simplifies to

\[
P(X_s|X_{V\setminus s}) = \exp \left\{ \sum_s \theta_s X_s + \sum_{t \in N(s)} \theta_{st} X_s X_t + C(X_s) - \bar{D}(X_{V\setminus s}; \phi) \right\} \quad (1.7)
\]
where \(N(s)\) denotes the set of nodes adjacent to node \(s\) (i.e. the neighborhood). Subsequently, this gives a closed form for the joint distribution:

\[
P(X) = \exp \left\{ \sum_{s} \theta_s X_s + \sum_{(s,t) \in E} \theta_{st} X_s X_t + \sum_{s} C(X_s) - A(\phi) \right\}.
\]

In the same paper, Yang et al. (2012) has also provided statistical guarantee about the structure learning, i.e. with high probability, the graphical models can be recovered from data. The proof relies on the idea that (under various conditions) for each node \(s\), the \(L_1\)-regularised conditional likelihood can recover the true neighborhood \(N(s)\), then by stitching together the neighborhoods for all the nodes, the full graph structure can be recovered.

Relation to this work

Vine-copula, copula and GGLM are all models of multivariate distributions alternative to the Structural GLM (SGLM), which is proposed in the third part of this work. Both vine-copulas and SGLM trade off global closed-form expression for a richer family of distributions, and one may draw the connection that SGLM is to GGLM as vine-copulas is to Copula.

<table>
<thead>
<tr>
<th>Modelling based on</th>
<th>marginals</th>
<th>conditionals</th>
</tr>
</thead>
<tbody>
<tr>
<td>Has no global closed form</td>
<td>Vine-copula</td>
<td>Structural GLM</td>
</tr>
<tr>
<td>Has global closed form</td>
<td>Copula</td>
<td>Graphical GLM</td>
</tr>
</tbody>
</table>

Table 1.1: Comparison of multivariate models
Chapter 2

Trend extraction from time series with missing data: How accurate could it be?

Chapter Summary

Extracting trends from time series data is a central task in many fields; when the data have missing parts, it is unclear to what degree accurate estimations of trends are possible. In this work, we answer the question: how accurate could trend estimates be when there are missing data? Considering an imputation approach for the missing data and linear trend extraction methods, we show that the trend estimate has error bounded above by the imputation error times a constant factor. We evaluate the constant factor for various methods, giving a simple way to assess the uncertainty of trend estimates. Examples and simulation studies are also presented to illustrate usage and performance for practical applications.

2.1 Introduction

Extracting trends from data is a central task in many fields, including economics, geophysics, climatology and engineering. When the data have missing parts, due to e.g. faulty equipment or entry error, it is unclear to what degree accurate estimations of trends are possible. In this chapter, we study the problem of missing data in the context of trend extraction; concretely, we answer the question: How accurate are the trend estimates when some data are missing?

To facilitate trend estimations, we need to first handle the missing data. Broadly speaking, we could fill them in or ignore them. While the latter may
sound vacuous, it is widely used in statistical packages (van Buuren, 2012, p.8). This approach is often realised in two ways: complete-case analysis, where observations with missing parts are simply ignored and analysis proceeds with the complete observations only, and pairwise deletion, where observations are discarded only when the missing parts are related to the variables of interest.

Despite the convenience, this approach has its drawbacks. In theory, it has been shown that the estimation may be inefficient or even biased (Little and Rubin, 2002). In practice, both complete-case analysis and pairwise deletion could waste a lot of data as complete records usually only form a small fraction of the dataset. This waste of data is more pronounced when the data is high-dimensional. In bio-informatics, where each record consists of hundreds of measurements, if each measurement has 1% chance of missing, then complete-case analysis would often discard more than 99% of the data (Heitjan, 2011). Overall, the approach of ignoring the missing data is well-understood and generally not recommended.

The above leads us to the second approach: filling in the missing data. Two widely accepted practices are the EM algorithm (Dempster et al., 1977) and the multiple imputations (Rubin, 1987). They not only produce unbiased results in a wide range of situations, but also have transparent assumptions and easy access to standard errors estimate. In the EM algorithm, missing data are treated as random variables, and inference is performed with the usual maximum likelihood methodology. The maximisation relies on the following recursive relation:

$$
\theta_{k+1} = \arg \max_{\theta} \int_{Y_{mis}} P(Y_{mis}|Y_{obs}, \theta_k) \log P(Y_{mis}, Y_{obs}|\theta) dY_{mis},
$$

where $\theta$ are the parameters of the model considered, $\theta_k$ is the current estimate of the parameters, and $Y_{mis}, Y_{obs}$ denotes the missing data and the observed data respectively. The procedure can be viewed as performing imputations implicitly. To see that, we consider the discrete analog of the integral, the Monte Carlo integral:

$$
\theta_{k+1} = \arg \max_{\theta} \sum_{i=1}^{L} \log \frac{P(Y_{mis}^{(i)}, Y_{obs}|\theta)}{L},
$$

where $Y_{mis}^{(i)}$ is the $i$-th sample drawing from the distribution $P(Y_{mis}|Y_{obs}, \theta_k)$. The “imputations” element comes from the fact that we are drawing $Y_{mis}^{(i)}$ from a distribution and using it to evaluate the quantity of interest, while the “implicitly” element comes from that the original integral is evaluated analytically (when possible), so no actual drawing and substitution at the missing point is done. The alternative to the EM algorithm is multiple imputations. For multiple imputa-
tions, the missing data are handled as follows. First, we specify a model for the missing data, then for each missing data-point, we generate a value from the model and impute the missing point with it. The analysis then proceeds as if there is no missing data. The procedure described up to this point is referred to as single imputation; iterating it multiple times gives the procedure of multiple imputations. Both single and multiple imputations are instances of explicit imputation methods, and this class of methods is what we will consider in the rest of this chapter.

Formally, we now define the problem examined in this chapter, and it begins with three objects: first, suppose we have a dataset where some data are missing, we denote it by $y_{\text{missing}}$; this is the observation we get in reality. Next, suppose we apply statistical methods to fill in the missing parts, we denote the imputed dataset by $y_{\text{imputed}}$. Finally, we refer the idealised dataset where no data are missing from the beginning by $y_{\text{complete}}$. We are concerned with the following problem: given $y_{\text{missing}}$, if we impute the missing data points to get $y_{\text{imputed}}$, is the trend extracted from $y_{\text{imputed}}$ close to the one extracted from $y_{\text{complete}}$? In other words, how much do missing data affect the trend estimation?

Our approach to the problem is to first compute the difference between the trend estimated from $y_{\text{complete}}$ and that from $y_{\text{imputed}}$, then find an analytic expression to bound this difference. Referring the data difference by “imputation error” and the trend difference by “trend error”, it turns out if the trend extraction method is linear, i.e. the output is a linear function of the inputs, then we can bound the trend error by the imputation error multiplied by a constant. The result tells us how much imputation error propagates to the trend estimates.

We now present our main results and discussion in the next two sections, followed by examples, simulation studies and an application. All the technical detail is given in the appendix.

2.2 Main results

2.2.1 The mathematical formulation

Our work uses some basic results from linear operator theory. Suppose we have a linear map $T : X \to Y$, where $X, Y$ are both normed vector space. We define $T$ to be bounded if there exists $M > 0$ such that

$$||T(v)|| \leq M \cdot ||v||, \quad \forall v \in X.$$
Also, we define the smallest $M$ satisfying above to be the operator norm of $T$, denoted by $||T||$. If $X = Y = R^n$ and $T = A$ is a $n \times n$ matrix mapping $R^n$ to $R^n$, then $A$ is bounded and $||A||$ is given by $||A|| = \sqrt{\lambda_L(A^T A)}$, where $\lambda_L(B)$ denotes the largest eigenvalue of a (symmetric) matrix $B$ (Lütkepohl, 1996).

We now apply this result to the problem of extracting trend with missing data. Trend extraction methods, in general, take data as input and return a trend as output. When the method is linear, the extraction of trends can be expressed as:

$$v_{\text{trend}} = A y_{\text{data}},$$

where $v_{\text{trend}} \in R^n$, $y_{\text{data}} \in R^n$, $n$ is the number of data points, and $A$ is the $n \times n$ trend extraction matrix. Recalling that the complete data are denoted by $y_{\text{complete}}$ and the imputed data are denoted by $y_{\text{imputed}}$, we have

\begin{align*}
\text{trend error} & = ||v_{\text{trend}}^{\text{complete}} - v_{\text{trend}}^{\text{imputed}}|| = ||A y_{\text{complete}} - A y_{\text{imputed}}|| \\
& = ||A (y_{\text{complete}} - y_{\text{imputed}})|| \leq ||A|| \cdot ||y_{\text{complete}} - y_{\text{imputed}}||. \quad (2.1)
\end{align*}

This inequality lays the ground for all the results we will present later.

Examining the derivation closely, on the first line, we started off with the difference of the trend estimates, extracted from the complete data and the imputed data respectively; we refer to this by “trend error” (due to missing data). On the second line, we ended with $||y_{\text{complete}} - y_{\text{imputed}}||$; we refer to this by “imputation error”. Overall, the inequality says that the trend error is bounded above by a constant times the imputation error. The constant can be evaluated once a trend extraction method is chosen. It follows that if the imputation error is known, then we know how large the trend error can be. This quantifies the uncertainty of the trend estimates.

From equation (2.2) we also see that the smaller $||A||$ is, the tighter of a bound we get. As a tight bound gives a more confident trend estimate, $||A||$ to some extent measures a method’s robustness against missing data. If $||A||$ is large, then even a small imputation error could lead to a large trend error; if $||A||$ is small, then a large imputation error only leads to a small trend error, i.e. the method is robust to large amount of missing data.

Returning to our main question, “is the trend extracted from $y_{\text{imputed}}$ close to the one extracted from $y_{\text{complete}}$?”, the answer is: it depends on the chosen trend extraction method and the imputation error, and the relation is given by the inequality in equation (2.2). To apply the inequality, we need to evaluate $||A||$ and $||y_{\text{complete}} - y_{\text{imputed}}||$; in the next section, we first evaluate $||A||$ for various trend extraction algorithms, then in the one after, we discuss how to estimate the imputation error.
Table 2.1: Examples of trend extraction methods (Kendall and Ord, 1990, p.30-31) that satisfy the criteria given in Theorem 1

<table>
<thead>
<tr>
<th>Methods</th>
<th>Moving average representations</th>
</tr>
</thead>
<tbody>
<tr>
<td>Henderson’s filter</td>
<td>(-325, -468, 0, 1100, 2475, 3600, 4032) / 16796</td>
</tr>
<tr>
<td><strong>Running polynomial:</strong></td>
<td></td>
</tr>
<tr>
<td>Quadratic and cubic, 7-point</td>
<td>(-2, 3, 6, 7) / 21</td>
</tr>
<tr>
<td>Quadratic and cubic, 13-point</td>
<td>(-11, 0, 9, 16, 21, 24, 25) / 143</td>
</tr>
<tr>
<td>Spencer’s 15-point moving average</td>
<td>(-3, -6, -5, 3, 21, 46, 67, 74) / 320</td>
</tr>
<tr>
<td>Spencer’s 21-point moving average</td>
<td>(-1, -3, -5, -2, 6, 18, 33, 47, 57, 60) / 350</td>
</tr>
<tr>
<td>Quartic and quintic, 7-point</td>
<td>(5, -30, 75, 131) / 231</td>
</tr>
<tr>
<td>Quartic and quintic, 13-point</td>
<td>(110, -198, -135, 110, 390, 600, 677) / 2431</td>
</tr>
</tbody>
</table>

2.2.2 Bounds evaluation

In this section, we evaluate $||A||$ for various trend extraction methods. In particular, we discuss the Moving Average filter and the Seasonal-Trend decomposition based on Loess. The former contains several other methods as special cases, while the latter serves as a more elaborate example.

**Moving Average**

Suppose we have $n$ time series data points, $y_1, y_2, \ldots, y_n$, and we denote a moving-average filter - a function that takes $2k + 1$ inputs $y_{i+j}, j = -k, \ldots, k$ and outputs $\hat{y}_i = \sum_{j=-k}^{k} a_j y_{i+j}$ where $\sum_{j=-k}^{k} a_j = 1$ - by

$$
(a_{-k}, a_{-k+1}, \ldots, a_{-1}, a_0, a_1, \ldots, a_{k-1}, a_k).
$$

To apply the moving average filter near the two ends of the time series, we assume $y_i = 0$ for $i < 1$ and $i > n$. This is in effect the same as truncating the filter at the point where data becomes unavailable. On the other hand, when the weights are symmetric about $a_0$, i.e. $a_i = a_{-i}, i = 1, \ldots, k$, we simply write $(a_{-k}, a_{-k+1}, \ldots, a_{-1}, a_0)$. Some of the popular filters are given in Table 2.1. Noticeably, all the filters share certain structures, e.g. the center weight $a_0$ is always the largest, followed by a series of sizable positive weights. These turn out to be key properties that ensure the corresponding trend extraction matrix have eigenvalues bounded by 1.

Formally, our first result is as follows. Given a moving-average filter (of length $2k + 1 < n$), the corresponding trend extraction matrix, $A : R^n \to R^n$, mapping
\((y_1, y_2, \ldots, y_n)\) to \((\hat{y}_1, \hat{y}_2, \ldots, \hat{y}_n)\) is defined to be:

\[
A = \begin{bmatrix}
a_0 & a_1 & a_2 & \cdots & a_k & 0 & \cdots & 0 \\
a_1 & a_0 & a_1 & \cdots & a_{k-1} & a_k & \cdots & 0 \\
a_2 & a_1 & a_0 & \cdots & a_{k-2} & a_{k-1} & \cdots & 0 \\
\vdots & \vdots & \vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\
0 & \cdots & 0 & a_k & \cdots & a_2 & a_1 & a_0
\end{bmatrix}.
\]

**Theorem 1.** If the moving-average filter satisfies that:

1. \(a_0 > \sum_{j:a_j<0} |a_j| \geq 0\), i.e. the mid-point of the filter carries a significant positive weight,
2. \(a_j = a_{-j}, j = 1, 2, \ldots, k\), i.e. it is symmetric,
3. There exists \(\theta_0 \in (0, \phi)\) such that \(A_d(\theta_0) < 0\), \(d = 2a\) for some \(a \in \mathbb{N}\), where
   \[
   A_d(\theta) = -2 \left[ \sum_{i=1}^{d} \theta^{2i-1} \frac{\sum_{j=1}^{k} (-1)^{i-1} a_j 2^i}{(2i-1)!} + \theta^{2d+1} \frac{\sum_{j>0:a_j<0} a_j 2^{d+2}}{(2d+1)!} \right],
   \]
   and
   \[
   \phi \text{ is the smallest positive real root of } A_d(\theta).
   \]
4. There exists some \(h > 0\) such that \(\lambda(\theta_i) + L \left( \frac{h}{2} \right) \leq 1, \forall i = 0, 1, \ldots, n\), where
   \[
   \lambda(\theta) = \sum_{j=-k}^{k} a_j \cos(j \theta), \quad L = \sum_{j=-k}^{k} |a_j| |j|, \quad \text{and}
   \]
   \[
   \theta_0 = \phi, \quad \theta_i = \phi + ih, i = 1, \ldots, n-1, \quad \theta_n = \phi + nh = 2\pi - \phi.
   \]
then \(||A|| \leq 1\).

**Proof.** See Appendix.

We remark that when the filter has all positive weights (e.g. the simple moving average filter), the trend extraction is merely an interpolation, and the results are easy to prove as one can leverage convexity and positivity. The real challenge here is the handling of negative weights which suggests extrapolation is involved, and the main contribution here is to provide sufficient conditions (that are easy to verify via a program) the filter has to satisfy to have eigenvalues bounded as desired.

**Corollary 1.** All the moving-average filters given in Table 1 satisfy \(||A|| \leq 1\).

**Proof.** Direct evaluation of the criteria given in Theorem 1 would give the result. R code is provided in the supplementary material for easy verification.
Relating back to equation (2.2), we see that if a moving average filter is used to extract trend, then the trend error can at most be as large as the imputation error. It is instructive to deduce further from equation (2.2):

\[
||v_{\text{complete}}^{\text{trend}} - v_{\text{imputed}}^{\text{trend}}|| \leq ||A|| \cdot ||y_{\text{complete}} - y_{\text{imputed}}||
\]  

(2.3)

\[
\Rightarrow \frac{||v_{\text{complete}}^{\text{trend}} - v_{\text{imputed}}^{\text{trend}}||}{\sqrt{n}} \leq ||A|| \cdot \frac{||y_{\text{complete}} - y_{\text{imputed}}||}{\sqrt{n}}
\]  

(2.4)

\[
\Rightarrow \text{RMSE}_{\text{trend}} \leq ||A|| \cdot \text{RMSE}_{\text{imputations}}
\]  

(2.5)

where \( n \) is again the number of data points, and \( \text{RMSE}_{\text{trend}}, \text{RMSE}_{\text{imputations}} \) denote the root-mean-squared errors of the trend and the imputations respectively.

The inequality suggests that at each data point, the trend extracted from imputed data should on average depart from the trend extracted from the complete data by no more than \( \text{RMSE}_{\text{imputations}} \) times a constant factor. This quantifies how much we could trust our trend estimates when there are missing data.

The methods listed in Table 2.1 are commonly used but they are rather simple. We now study a more elaborate method.

**Seasonal-Trend decomposition based on Loess (STL)**

Seasonal-Trend decomposition based on Loess, proposed by Cleveland et al. (1990) and also known as STL, is a non-parametric decomposition method for time-series data. Through iterative smoothing of the data, it decomposes the time series into three components: the trend component, the seasonal component and the remainder component. From a frequency analysis point of view, what STL does is to filter out signals of different frequencies; the signal with the lowest frequency is regarded as the trend, the one with the medium frequency is the seasonal component, and the remaining ones with the highest frequency are the noise. The major advantages of STL are that it has fast convergence and it can handle non-stationary data, missing data and outliers. We give in Figure 2.1 a schematic representation of the algorithm and the corresponding list of symbols in Table 2.2. Readers are referred to the seminal paper (Cleveland et al., 1990) for full details of the STL procedure.

To apply STL, six parameters need to be specified, they are the number of outer loops, \( n_o \), the number of inner loops, \( n_i \), the number of cycle-subseries \( n_p \), the neighbourhood size for seasonal smoothing, \( n_s \), the neighbourhood size for trend smoothing, \( n_t \), and the neighbourhood size for seasonal trend smoothing, \( n_l \). Cleveland et al. (1990) recommends the following choice for the parameters:
Figure 2.1: Schematic representation of STL
Table 2.2: List of symbols

<table>
<thead>
<tr>
<th>STL parameters</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>$n_o$</td>
<td>number of outer loops</td>
</tr>
<tr>
<td>$n_i$</td>
<td>number of inner loops</td>
</tr>
<tr>
<td>$n_p$</td>
<td>number of cycle-subseries</td>
</tr>
<tr>
<td>$n_s$</td>
<td>the neighbourhood size for seasonal smoothing</td>
</tr>
<tr>
<td>$n_t$</td>
<td>the neighbourhood size for trend smoothing</td>
</tr>
<tr>
<td>$n_l$</td>
<td>the neighbourhood size for seasonal trend smoothing</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Loess parameters</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>$d$</td>
<td>the degrees of polynomial</td>
</tr>
<tr>
<td>$q$</td>
<td>the neighbourhood size for smoothing</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Components</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>$T^{(i)}_v$</td>
<td>the trend component at the $i$-th iteration</td>
</tr>
<tr>
<td>$S^{(i)}_v$</td>
<td>the seasonal component at the $i$-th iteration</td>
</tr>
<tr>
<td>$R_v$</td>
<td>the remainder component</td>
</tr>
</tbody>
</table>

- $n_o = 0, n_i = 2$ if resistance to outliers are not needed and $n_o = 5, n_i = 1$ otherwise.
- $n_p$ depends on the application, for example, 12 would be appropriate for monthly climate data.
- $n_s$ is specified by the user to incorporate prior knowledge on the regularity of the seasonal pattern. This must be an odd integer $\geq 7$.
- $n_t = [1.5n_p/(1 - 1.5n_s^{-1})]_{\text{odd}}$ and $n_l = [n_p]_{\text{odd}}$, where $[x]_{\text{odd}}$ denotes the smallest odd integer greater than or equal to $x$.

Now we present our second result, which is based on the setting in Cleveland et al. (1990).

**Theorem 2.** Suppose a time series $Y_i, i = 1, 2, ..., n$ is circular, i.e. $Y_i = Y_j$ if $i \equiv j \pmod{n}$, and the parameters of STL are chosen according to Section 2.2.2, then the trend extraction matrix of STL, $A$, satisfies that $\|A\| \leq 1$.

**Proof.** See Appendix. \qed

The circular assumption about the time series is made for the sake of mathematical convenience, but as it is noted by Cleveland et al. (1990), the setting should apply well to long time series, in particular when $n_t$ and $n_l$ are small relatively to $n$ and $n_s$ is small relatively to $n/n_p$. 

30
2.3 Estimating imputation error

Having evaluated \( ||A|| \) for some trend extraction methods, it remains to estimate the imputation error \( ||y_{\text{complete}} - y_{\text{imputed}}|| \). This together with the result we saw in Section 2 complete our discussion of equation (2.2), answering the question “how accurate could trend estimates be when there are missing data?”.

In general, the estimation of imputation error proceeds differently for different imputation methods. Here we will use local polynomial regression (aka loess) (Cleveland and Devlin, 1988) as an example. We made this choice because smoothing methods are commonly used to handle missing data, and loess serves well as a representative to illustrate usage. Loess assumes a model of \( Y = f(t) + \epsilon \), where \( f \) is a smoother, \( E(\epsilon) = 0 \) and \( \text{Var}(\epsilon) = \sigma^2 \). To facilitate analysis, we assume that \( f \) is twice differentiable and \( f''(t) \) is bounded, i.e. \( |f''(t)| \leq K \) for some \( K > 0 \).

We discuss two ways to estimate the imputation error: a theoretical one and a data-driven one. The former utilizes assumptions about the data and the imputation method - which in our case is the smoothness assumption - to give an analytic upper bound of the error; the latter utilizes properties of the observations, making no model assumptions about the data, to bound the imputation error from above.

2.3.1 Theoretical approach

We consider a case where data are missing every other point, i.e. out of the complete data \( y_1, y_2, y_3, \ldots, y_n \), we have \( y_2, y_4, y_6, \ldots \) missing. We use the same notation as the previous section: denote the time series data by \( y \) and the data with imputed values by \( y_{\text{imputed}} \). We also write \( y_i = f(t_i), i = 1, 2, \ldots, n \) since we have \( E(Y|t) = f(t) \) under the loess model.

We begin by finding the imputation error at a point \( t_j \), i.e. we find \( |y_j - y_{j,\text{imputed}}| = |f(t_j) - \hat{f}(t_j)| \), bearing in mind that we are only interested in a rough aggregated estimate rather than accurate individual estimates. By Taylor’s theorem, we have

\[
f(t_j + \Delta t) = f(t_j) + f'(t_j)\Delta t + \frac{f''(\zeta)}{2} (\Delta t)^2,
\]

where \( \zeta \in (t_j, t_j + \Delta t) \). Next, we express the loess estimator in a form of equivalent kernels. We denote the kernel weight associated with a data point \( t_i \) by \( w_i \), then
we have

\[
\hat{f}(t_j) = \sum_{t_i \in N(t_j)} w_i f(t_i)
\]

\[
= \sum_{t_i \in N(t_j)} w_i \left[ f(t_j) + f'(t_j)(t_i - t_j) + f''(\zeta_i)(t_i - t_j)^2/2 \right],
\]

where \( \sum_i w_i = 1 \), \( \zeta_i \in (\min(t_i, t_j), \max(t_i, t_j)) \) and \( N(t_j) \) is the neighbourhood of \( t_j \).

As time series data have equally spaced data points, and we assume the data are missing every other point, so the neighbourhood and the kernel weights are symmetric about the imputed point. This implies \( \sum_{t_i \in N(t_j)} w_i f'(t_j)(t_i - t_j) = 0 \) giving

\[
|\hat{f}(t_j) - f(t_j)| = \left| \sum_{t_i \in N(t_j)} w_i f''(\zeta_i)(t_i - t_j)^2/2 \right| \leq \sum_{t_i \in N(t_j)} \left| w_i f''(\zeta_i)(t_i - t_j)^2/2 \right| \leq \sum_{t_i \in N(t_j)} K \left| w_i (t_i - t_j)^2 \right| = \sum_{t_i \in N'(t_j)} K \left| w_i (t_i - t_j)^2 \right|
\]

where \( N'(t_j) \) is the one-sided neighbourhood of \( t_j \).

Now suppose the size of the neighbourhood of \( t_j \) is \( 2l \), and note that the kernel weights \( w_i \) is given by \( W(i/l) / \sum_{j=-l}^{l} W(j/l) \), \( i = -l, -l + 1, \ldots, l - 1, l \), where \( W(\cdot) \) is the tricube weight function, then our missing data pattern implies

\[
|\hat{f}(t_j) - f(t_j)| \leq \sum_{i=1}^{l} K \left| \frac{W((2i - 1)/(2l - 1))}{\sum_{j=1}^{l} W((2j - 1)/(2l - 1))} (2i - 1)^2 \right|; \]

with some algebra, the expression on the right simplifies to

\[
K \cdot \left[ \frac{217}{1000} - \frac{7l}{25} + \frac{7l^2}{60} + O \left( \frac{1}{l} \right) \right].
\]

We have thus found an upper bound for the imputation error at the point \( t_j \). To get the imputation error over the whole time series, we simply sum over the individual errors to get

\[
||y - y_{\text{imputed}}||^2 = \sum_j (y_j - y_{\text{imputed}})^2 \leq \frac{nK^2}{2} \cdot \left[ \frac{217}{1000} - \frac{7l}{25} + \frac{7l^2}{60} + O \left( \frac{1}{l} \right) \right]^2.
\]
In the above, we assumed the neighborhood size is the same for all points to make the final expression tractable and also used the fact that in our setting, we have \( n/2 \) points missing (ignoring odd-even parity).

Examining the last expression, we see that the parameter \( K \) is yet to be determined. One choice is to use the “empirical” estimate: we first smooth the data, differentiate the resulting curve twice and then take maximum to get \( K \). The \texttt{fda} R package could be used for this task. As a remark, we recommend multiplying the maximum by a factor greater than 1 (e.g. 1.5) to protect against potential underestimation of \( K \) due to smoothing.

In principle, other missing data patterns can be analyzed in the same way as shown in our derivation. The approach we used is common for analyzing linear smoothers, and similar derivations can be found in Hastie and Tibshirani (1990, p.19, 41) and Fan and Gijbels (1996, p.58, 84).

The above concludes our theoretical approach to the estimation of imputation error. We saw that while this approach is rigorous, it is also fairly elaborate. This motivates an easier alternative: a data-driven approach. But before that, we end this section by briefly describing how to put together all the results we have seen so far.

Usage

Here is an example of how one applies our results in practice. Suppose we use loess for imputation and STL for trend extraction, then to get a bound for the trend error, we substitute equation (2.12) and Theorem 2 into equation (2.5) to get:

\[
RMSE_{\text{trend}} \leq \begin{cases} 
\sqrt{2}K \cdot \left[ \frac{217}{1000} - \frac{7l}{25} + \frac{7l^2}{60} + O \left( \frac{1}{l} \right) \right] & \text{if } n_i = 1 \\
\sqrt{8}K \cdot \left[ \frac{217}{1000} - \frac{7l}{25} + \frac{7l^2}{60} + O \left( \frac{1}{l} \right) \right] & \text{if } n_i = 2 
\end{cases}
\] (2.13)

We have thus bounded the mean-squared error of the trend estimate from above. This inequality gives information about the average squared error at a missing point. Moreover, equation (2.13) provides some guidelines on how to pick the neighborhood parameter of loess. Specifically, the (half) neighborhood size \( l \) may be chosen such that the upper bound is tight. However, there is a potential trade-off between (i) choosing a small \( l \) for a tight bound and getting a relatively large error (close to the bound) and (ii) choosing a large \( l \) for a loose bound and getting a relatively small error (away from the bound). The decision should be made based on the research context and needs.
2.3.2 Data-driven approach

The steps involved in the data-driven approach are similar to the theoretical approach: We first find the bound for the imputation error at a point, then multiply that by the number of missing data point to get the bound for the total imputation error.

We suggest two data-driven ways to estimate the individual imputation error. The first one is to use the maximum of the first absolute difference as an estimate, written as max\(_t\) \(|y_t - y_{t-1}|\). This idea is based on two heuristics: (i) data points do not vary more than the largest observed one-step variation most of the time, and (ii) any reasonable imputation methods should do better than the carrying-forward strategy, where one simply uses the last point available and carry that forward to impute the missing data point. This bound is generally conservative. The second way is to remove a couple of data points neighboring the missing data point, then apply the chosen imputation method to impute those points and calculate the “empirical” imputation error.

2.4 Simulation studies

In this section, we verify our results through simulation studies. First we simulate some time series data and randomly remove some data points from it. This gives us a dataset with missing data. The two settings we use to simulate the data are:

**Setting 1.** We simulate data from ARIMA(12,1,0) with Gaussian noise, and add to it a smooth trend, generated by smoothing a random piecewise linear function.

**Setting 2.** We sample some data from the Antarctic Temperature database, available at [https://www.ncdc.noaa.gov/data-access/weather-balloon/integrated-global-radiosonde-archive](https://www.ncdc.noaa.gov/data-access/weather-balloon/integrated-global-radiosonde-archive). The data comes in the form of daily time series, collected at 22 stations, 16 altitude levels 2 times a day over 50 years. We remove the trend from the sampled data using STL, then add to it a smooth trend, again generated by smoothing a random piecewise linear function.

Next, we impute the missing points and extract a trend from the dataset, and finally since we know the complete data and the true trend, we can compute the imputation error and the trend error and check their consistency with the theoretical results we derived. 200 simulations are performed, and the results are summarized in Figures 2.2 and 2.3.

Figure 2.2 shows the simulation studies conducted under setting 1. The trend error and various upper bounds are shown. We refer the “approximate bound” to when the imputation error is estimated by the maximum-absolute-first-difference
Verification and Comparison of upper bounds of trend error
heuristic, the “empirical bound” to when the imputation error is estimated by the remove-then-impute-neighboring-points heuristic and the “true bound” by when the imputation error is computed exactly from the simulated data. We see that the approximative bound is overly conservative while the empirical bound is reasonably close to the true bound. The true bound never crosses the trend error—it is valid for all 200 simulations just as our theorem suggested—while the approximative bound and the empirical bound, which are based on heuristic, only holds for 198 times out of the 200 simulations. Similarly, Figure 2.3 shows results corresponding to setting 2. The results are consistent with setting 1: the true bound and the approximative bound hold true for all 200 simulations while the empirical bound holds for 199 out of the 200 simulations.

We remark that in general, the gap between the true bound and the trend error cannot be reduced because the bound applies to all possible scenarios including the worst-case scenario in particular. When all the individual imputation errors are of the same sign, they cause a serious distortion to the trend estimates. While this case occurs only with very small probability, it is possible and covered by our bound, explaining the gap between the true bound and the (typical) trend error. Overall, we would recommend using the heuristic bounds in practice except for safety-critical systems, in which case the theoretical bound should be used instead.

2.5 Application

Deriving an accurate trend in meteorological data (e.g. temperature) is important for detection and attribution of climate change. To derive plausible trends, long-term time series data—typically 30+ years long—are used. However, these time series often have missing data (e.g. due to failure of instruments) which impact the accuracy of the estimated trend. In remote areas like the Arctic and the Antarctic, the proportion of missing data could be particularly large, but at the same time, accurate trend estimates over these areas are of great importance. For example, response of the Arctic to global warming is one of the major indicators of climate change (IPCC, 2014).

In this section, we apply our analysis to the Antarctic temperature data introduced in section 2.4. In particular, we examine the temperature time series collected at the Novolazaravskaja station at 8 different pressure levels. The missing data in these time series show a high degree of dispersedness, and it is what our analysis is designed for. We use loess smoothing to impute the missing data and STL to extract trends from the time series. In Figure 2.4, we show one of the time series before and after imputations. The circles are the original data points,
and the rhombuses are the imputed data points. The time series is 498 months long and has 42 data points missing, which is equivalent to a missing proportion of 8.4%. Upon the imputation, STL is applied to extract trends from the time series. We do this to each of the 8 time series and generate a profile plot. The result is presented in Figure 2.5. In the figure, the bars and the dots represent the average temperature change per decade at the corresponding pressure level, where the average change refers to the slope coefficient of the OLS line fit on the extracted trend. The ±2 standard errors are provided using error bars, and a smoothed line is plotted to show the dynamics of the average temperature change over the different pressure levels. The result shows that while there is warming from the 400-hPa pressure level and above, there is also significant cooling below the 400-hPa level. Note that pressure level is inversely related to altitude, so the result translates to that warming is only happening below a certain altitude while it is actually cooling at the ultra-high altitudes.

Figure 2.4: An illustration of the Novolazaravskaja time series before and after imputation

2.6 Conclusion

In this work, we studied to what extend do missing data affect trend analysis. We showed that for a large class of linear trend extraction method, the influence of missing data can be characterized by an inequality. We evaluate the inequality for several trend extraction methods and give examples on how to apply our results in practice. Our results translate insight about the missing data to that of the trend estimates, this can be used to settle doubts drawn towards trend analysis with incomplete datasets.
2.7 Appendix

In this section, we present the proofs of theorem 1 and 2. We first state some preliminary results and then the proof.

2.7.1 Preliminaries

Notation:
\[ \lambda_L(A) \] denotes the largest eigenvalue of a symmetric matrix \( A \).
\[ \lambda_S(A) \] denotes the smallest eigenvalue of a symmetric matrix \( A \).

The following definition and lemmas are taken from Lütkepohl (1996, p.104, 113-114, 137).

**Definition 1.** An \((n \times n)\) matrix

\[
circ(a_0, a_1, \ldots, a_{n-1}) = \begin{bmatrix}
  a_0 & a_1 & a_2 & \cdots & a_{n-2} & a_{n-1} \\
  a_{n-1} & a_0 & a_1 & \cdots & a_{n-3} & a_{n-2} \\
  a_{n-2} & a_{n-1} & a_0 & \cdots & a_{n-4} & a_{n-3} \\
  \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\
  a_2 & a_3 & a_4 & \cdots & a_0 & a_1 \\
  a_1 & a_2 & a_3 & \cdots & a_{n-1} & a_0 
\end{bmatrix}
\]
is a circulant matrix.

**Definition 2.** Let $A$ be a $n \times n$ matrix. Any matrix that contains exactly the first $k$ rows and columns of $A$, where $k \in \{1, 2, ..., n - 1\}$, is a principal submatrix of $A$.

**Definition 3.** Let $A = [a_{i,j}]$ be a $n \times n$ matrix. The $k$-diagonal entries are the set of entries:

$$\{a_{i,i+k}, i = 1, ..., n-k\} \text{ if } k \geq 0, \text{ and } \{a_{i-k,i}, i = 1, ..., n+k\} \text{ if } k < 0.$$

**Lemma 1.** Let $A$ be a matrix and consider $A : x \mapsto Ax, x \in \mathbb{R}^n$ as an operator, then $||A|| = \sqrt{\lambda_L(A^T A)}$.

**Lemma 2.** If $A = \text{circ}(a_0, a_1, ..., a_{n-1})$ and $B = \text{circ}(b_0, b_1, ..., b_{n-1})$ are $n \times n$ circulant matrices, then

(a) $A + B$ and $AB$ are circulant;

(b) $\lambda_j = \sum_{k=0}^{n-1} a_k w_j^k$, where $w_j = \exp(2\pi ij/n), j = 0, 1, ..., n - 1$, are the eigenvalues of $A$.

(c) $\lambda_j(AB) = \lambda_j(A)\lambda_j(B)$ and $\lambda_j(A + B) = \lambda_j(A) + \lambda_j(B)$, where $\lambda_j(M)$ is the $j$-th eigenvalue of a matrix $M$.

**Lemma 3.** If $A$ is a $n \times n$ Hermitian matrix with eigenvalues $\lambda_1(A) \leq \ldots \leq \lambda_n(A)$, $D$ ($k \times k$) Hermitian, $C$ ($n \times k$) and

$$B = \begin{bmatrix} A & C \\ C^T & D \end{bmatrix},$$

with eigenvalues $\lambda_1(B) \leq \ldots \leq \lambda_{n+k}(B)$, then

$$\lambda_i(B) \leq \lambda_i(A) \leq \lambda_{k+i}(B), i = 1, \ldots, n.$$ In particular, later we will use the case where $i = n$ giving $\lambda_L(A) \leq \lambda_L(B)$.

**2.7.2 Proof of theorem 1**

**Proof**

For readers’ convenience, we repeat the statement of the theorem here. Suppose we have $n$ data points and the weights of a moving average filter (of length $2k+1$
< n) are given by \( w = [a_k, a_{k-1}, ..., a_1, a_0, a_{k-1}, ..., a_k] \), where \( \sum_{j=-k}^{k} a_j = 1 \), we define the transformation matrix \( A : R^n \rightarrow R^n \), corresponding to \( w \), to be

\[
A = \begin{bmatrix}
    a_0 & a_1 & a_2 & \cdots & a_k & 0 & \cdots & 0 \\
a_1 & a_0 & a_1 & \cdots & a_{k-1} & a_k & \cdots & 0 \\
a_2 & a_1 & a_0 & \cdots & a_{k-2} & a_{k-1} & \cdots & 0 \\
    \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\
    0 & \cdots & 0 & a_k & \cdots & a_2 & a_1 & a_0
\end{bmatrix}.
\]

**Theorem.** If the moving-average filter satisfies that:

1. \( a_0 > \sum_{j \neq 0} |a_j| \geq 0 \), i.e. the mid-point of the filter carries a significant positive weight,

2. \( a_j = a_{-j}, j = 1, 2, ..., k \), i.e. it is symmetric,

3. There exists \( \theta_0 \in (0, \phi) \) such that \( A_d(\theta_0) < 0, d = 2a \) for some \( a \in N \), where

\[
A_d(\theta) = -2 \left[ \sum_{i=1}^{d} \theta^{2i-1} \sum_{j=1}^{k} (-1)^{j-1} a_j j^{2i} \right] + \theta^{2d+1} \sum_{j=0-a_j<0} a_j j^{2d+2} \]

and

\[
\phi \text{ be the smallest positive real root of } A_d(\theta).
\]

4. There exists some \( h > 0 \) such that \( \lambda(\theta_i) + L \left( \frac{h}{2} \right) \leq 1, \forall i = 0, 1, ..., n \), where

\[
\lambda(\theta) = \sum_{j=-k}^{k} a_j \cos(j\theta), \quad L = \sum_{j=-k}^{k} |a_j| |j|, \quad \text{and}
\]

\[
\theta_0 = \phi, \quad \theta_i = \phi + i h, \quad i = 1, ..., n-1, \quad \theta_n = \phi + n h = 2\pi - \phi.
\]

then \( ||A|| \leq 1 \).

**Proof.** Since \( ||A|| = \lambda_L(A^T A) = \lambda_L(A^2) \) (as \( A \) is symmetric) which is equal to either \( \lambda_L(A)^2 \) or \( \lambda_S(A)^2 \), it suffices to show \( |\lambda_j(A)| \leq 1, \forall j \in \{0, 1, ..., n-1\} \). The key observation here is that given \( A \ (n \times n) \), we can construct a \((n+k) \times (n+k)\) circulant matrix \( C \) such that \( |\lambda_j(C)| \leq 1, \forall j \in \{0, 1, ..., n+k-1\} \) and that it contains \( A \) as a principal submatrix. Then by Lemma 3, we know the absolute values of all eigenvalues of \( A \) must be less than or equal to 1.

Formally, consider a matrix \( C = \begin{bmatrix} A & B \\ B^T & D \end{bmatrix} \), where

\[
B = \begin{bmatrix}
    a_k & a_{k-1} & a_{k-2} & \cdots & a_2 & a_1 \\
    0 & a_k & a_{k-1} & \cdots & a_3 & a_2 \\
    \vdots & \vdots & \ddots & \ddots & \vdots & \vdots \\
    0 & 0 & 0 & \cdots & 0 & a_k
\end{bmatrix}
\]
and $D$ is the $k$-th order principal submatrix of $A$; $C$ is circulant by construction.

Next, we want to show the absolute values of all eigenvalues of $C$ are less than or equal to 1, i.e. $|\lambda_h(C)| \leq 1$ for all $h = 0, 1, 2, ..., n + k - 1$. We write $m = n + k$ and $\lambda_h(C)$ as $\lambda_h$ for convenience.

We first show $\lambda_h > -1$. From condition (1), we have

$$a_0 + \sum_{j:a_j<0} a_j > 0 > -a_0 - \sum_{j:a_j<0} a_j$$

$$\Rightarrow \left[ a_0 + \sum_{j:a_j<0} a_j \right] - \sum_{j:a_j>0,j\neq 0} a_j > \left[ -a_0 - \sum_{j:a_j<0} a_j \right] - \sum_{j:a_j>0,j\neq 0} a_j$$

$$\Rightarrow a_0 + \sum_{j\neq 0} |a_j|(-1) > -1,$$

(on the RHS, we used the fact that $\sum_j a_j = 1$) and then from Lemma 2b, we have

$$\lambda_h = a_0 + \sum_{j\neq 0} a_j \cos \left( \frac{2\pi j h}{m} \right) > a_0 + \sum_{j\neq 0} |a_j|(-1),$$

so $\lambda_h > -1$.

It remains to show $\lambda_h < 1$. To show $\lambda_h = \sum_{j=-k}^{k} a_j \cos \left( \frac{2\pi j h}{m} \right) < 1$, note that $\frac{2\pi h}{m} \in [0,2\pi]$ for all $m \in \mathbb{N}$, $h = 0, 1, ..., m - 1$, and consider the continuous extension $\lambda(\theta) = \sum_{j=-k}^{k} a_j \cos (j\theta)$; if we can show $\lambda(\theta) \leq 1, \forall \theta \in [0,2\pi]$, then we are done. We need the following two lemmas:

**Lemma 4.** Let

1. $A_d(\theta) = -2 \left[ \sum_{i=1}^{d} \theta^{2i-1} P^{(i)} + \theta^{2d+1} P^{(d-)} \right]$, where $d = 2a$ for some $a \in \mathbb{N}$;

   $$P^{(i)} = \sum_{j=1}^{k} (-1)^{i-j} a_j \frac{j^{2i}}{(2i-1)!}, \quad P^{(d-)} = \sum_{j>0,a_j<0} a_j \frac{j^{2d+2}}{(2d+1)!},$$

2. $\phi$ be the smallest positive real root of $A_d(\theta)$,

then if there exists $\theta_0 \in (0,\phi)$ such that $A_d(\theta_0) < 0$, then $\lambda(\theta) < 1, \forall \theta \in (0,\phi) \cup (2\pi - \phi, 2\pi)$.

**Proof.** (An illustration of the proof strategy is provided in Figure 2.6 to aid reading.) To show $\lambda(\theta) < 1, \forall \theta \in (0,\phi) \cup (2\pi - \phi, 2\pi)$, first note that we only need to show $\lambda(\theta) < 1, \forall \theta \in (0,\phi)$, then use the symmetry of the cosine function to get the $(2\pi - \phi, 2\pi)$ part, and second, as $\lambda(0) = 1$, it is sufficient to show $\lambda'(\theta) < 0, \forall \theta \in (0,\phi)$.  

42
Figure 2.6: An illustration of the proof strategy (using the Henderson’s filter). Our goal is to show $\lambda(\theta) \leq 1, \forall \theta \in \mathbb{R}$. The plot at the top shows that if there exists a set of points where the evaluated values are well below 1, then by Lipschitz continuity, the entire interval is below 1. However, this reasoning cannot be applied to the situation near the boundary. So at the bottom plot, we bound $\lambda'(\theta)$ from above using Taylor approximation, then show that if $\lambda(0) = 1$ and the approximate derivative is entirely negative near the left boundary (therefore, so is $\lambda'(\theta)$), then $\lambda(\theta) \leq 1$ near the left boundary. Then by symmetry of $\lambda(\theta)$, we also have $\lambda(\theta) \leq 1$ near the right boundary. Finally, by periodicity, this is true for $\theta \in \mathbb{R}$. 
To show $\lambda'(\theta) < 0, \forall \theta \in (0, \phi)$, we will show $A_d(\theta)$ is constructed to be an upper bound of $\lambda'(\theta)$. If there is a $\theta_0$ between the two roots 0 and $\phi$ such that $A_d(\theta_0) < 0$, then all points between the two roots have the same sign, i.e. $A_d(\theta_0) < 0, \forall \theta \in (0, \phi)$. Then since $\lambda'(\theta) \leq A_d(\theta)$, we have the result.

Now we present the proof $\lambda'(\theta) \leq A_d(\theta)$, $\theta \geq 0$. Using Taylor series expansion, we have that for $d = 2a$, $a \in \mathbb{N}$

$$\sum_{i=1}^{d} (-1)^{i-1} \frac{x^{2i-1}}{(2i-1)!} \leq \sin(x) \leq \sum_{i=1}^{d+1} (-1)^{i-1} \frac{x^{2i-1}}{(2i-1)!}, \forall x \geq 0$$

Next, $\lambda(\theta) = \sum_{j=-k}^{k} a_j \cos(j \theta) \Rightarrow \lambda'(\theta) = \sum_{j=-k}^{k} -a_j j \sin(j \theta)$, and we have that for $d = 2a$, $a \in \mathbb{N}$

$$\lambda'(\theta) = -2 \sum_{j=1}^{k} a_j j \sin(j \theta) = \sum_{j>0:a_j>0} -2a_j j \sin(j \theta) + \sum_{j>0:a_j<0} -2a_j j \sin(j \theta)$$

$$\leq \sum_{j>0:a_j>0} -2a_j j \left( \sum_{i=1}^{d} (-1)^{i-1} \frac{(j \theta)^{2i-1}}{(2i-1)!} \right) + \sum_{j>0:a_j<0} -2a_j j \left( \sum_{i=1}^{d+1} (-1)^{i-1} \frac{(j \theta)^{2i-1}}{(2i-1)!} \right)$$

$$= -2 \left[ \left( \sum_{i=1}^{d} (-1)^{i-1} \frac{\theta^{2i-1}}{(2i-1)!} \sum_{j=1}^{k} a_j j^{2i} \right) + \left( \frac{\theta^{2d+1}}{(2d+1)!} \sum_{j>0:a_j<0} a_j j^{2d+2} \right) \right]$$

$$= -2 \left[ \sum_{i=1}^{d} \theta^{2i-1} P^{(i)} + \frac{\theta^{2d+1}}{(2d+1)!} \sum_{j>0:a_j<0} a_j j^{2d+2} \right] = A(\theta)$$

As a remark, generally setting $d = 2$ (together with the next lemma) would be enough to verify the theorem in practice, and there is a closed-form formula for $d = 2$. When $d = 2$, the root of $A(\theta)$ is given by

$$\theta = 0 \quad \text{or} \quad \theta = \pm \sqrt{\frac{P^{(2)} \pm \sqrt{P^{(2)}^2 - 4P^{(3-)}P^{(1)}}}{2P^{(3-)}}}.$$ 

Furthermore, when a filter is generated by fitting a local polynomial of degree $2d_0 + 1$, $P^{(i)}$ is usually 0 for $i = 1, 2, \ldots, d_0 - 1$. Then similar closed-form formula of the root exists and it is given by:

$$\theta = 0 \quad \text{or} \quad \theta = \pm \sqrt{\frac{P^{(2d_0)} \pm \sqrt{P^{(2d_0)^2} - 4P^{(2d_0+1-)}P^{(2d_0-1)}}}{2P^{(2d_0+1-)}}}.$$ 

**Lemma 5.** $\lambda(\theta)$ is Lipschitz continuous with Lipschitz constant $\sum_{j=-k}^{k} |a_j||j|$. 

44
Proof. For $\theta_1 < \theta_2$,

$$|\lambda(\theta_1) - \lambda(\theta_2)| = \left| \sum_{j=-k}^{k} a_j (\cos(j\theta_1) - \cos(j\theta_2)) \right|$$

(2.14)

$$\leq \sum_{j=-k}^{k} |a_j| \cdot |\cos(j\theta_1) - \cos(j\theta_2)|$$

(2.15)

$$= \sum_{j=-k}^{k} |a_j| \cdot | - j \sin(j\theta^*)(\theta_1 - \theta_2) | \text{ for some } \theta^* \in (\theta_1, \theta_2)$$

(2.16)

$$\leq \left( \sum_{j=-k}^{k} |a_j| \cdot |j| \right) \cdot |\theta_1 - \theta_2| \text{ (since } \sin(x) \leq 1, \forall x \in \mathbb{R})$$

(2.17)

Continuing from above, let $L = \sum_{j=-k}^{k} |a_j||j|$, and suppose we evaluate $\lambda(\theta)$ on a grid $[a, a+h, a+2h, ..., a+nh = b]$ and get $\{((\theta_i, \lambda(\theta_i))\}^n_{i=1}$. Note that for any fixed $i_0$, if $\lambda(\theta_{i_0}) + L \left( \frac{h}{2} \right) \leq 1$, then $\lambda(\theta) \leq 1, \forall \theta \in [\theta_{i_0} - \frac{h}{2}, \theta_{i_0} + \frac{h}{2}]$ by Lipschitz continuity. Therefore, if $\lambda(\theta_i) + L \left( \frac{h}{2} \right) \leq 1, \forall i$, then $\lambda(\theta) \leq 1, \forall \theta \in [a, b]$. In the theorem, we have chosen $a = \phi$ and $b = 2\pi - \phi$. Overall, we know $\lambda(0) = 1$, Lemma 4 establishes $\lambda(\theta) < 1, \forall \theta \in (0, \phi) \cup (2\pi - \phi, 2\pi)$ and Lemma 5 establishes $\lambda(\theta) \leq 1, \forall \theta \in [\phi, 2\pi - \phi]$. Hence, $\lambda(\theta) \leq 1, \forall \theta \in [0, 2\pi]$, and the proof is complete.

\[\square\]

2.7.3 Proof of theorem 2

We first state the settings and consequences given in Cleveland et al.(1990) and then the proof of the proposition.

Settings: The time series $Y_i, i = 1, 2, ..., is circular, i.e. $Y_i = Y_j$ if $i \equiv j \ (mod \ N)$, and the parameter choices follow the recommendation as given in section 2.2.2.

Consequences: Denote the operator matrices associated with the operations in step 2 and 3 of section 2.2.2 by $S$ and $T$ respectively. To be clear, $S$ is the operator matrix that takes the input $Y^{\text{detrend}}$ and outputs the seasonal component $S$, and $T$ is the operator matrix that takes the input $Y^{\text{Deseasonalised}}$ and outputs the revised trend $T$. Given the above, we have

\begin{itemize}
  \item [(C1)] $Y_i$ is circular implies $S$ and $T$ are circulant matrices;
\end{itemize}
(C2) Enforcing the parameter choices in section 2.2.2 implies that \( \forall k, |s_k| \leq 1 \) and \( |t_k| \leq 1 \) with at most one of \( |s_k| = 1 \) and \( |t_k| = 1 \) occurring, where \( s_k \) and \( t_k \) denote the eigenvalues of the operator matrices \( S \) and \( T \).

Now we present the proof of Theorem 2.

**Proof.** We denote the operator matrix of the trend filter of STL after \( k \) iterations of inner loop by \( T_k \); the expression for \( T_k \) is given by (Cleveland et al.,1990)

\[
T_k = \sum_{m=1}^{2k} (-1)^{m-1} B_m, \quad \text{where } B_m = \begin{cases} 
(TS)^{m/2} & \text{for } m \text{ even} \\
(TS)^{(m-1)/2}T & \text{for } m \text{ odd}
\end{cases}
\]  

(2.18)

The proposition to be shown is that \( ||T_k|| \leq 2k \). We simply need to evaluate \( ||T_k|| \) explicitly to prove that. First from (2.18), we have

\[
T_k = \sum_{m=1}^{2k} (-1)^{m-1} B_m
\]

\[= T - TS + (TS)T - (TS)^2 + (TS)^2T - (TS)^3 + \ldots + (TS)^{k-1}T - (TS)^k
\]

\[= [T + (TS)T + (TS)^2T + \ldots + (TS)^{k-1}T]
\]

\[- [TS + (TS)^2 + (TS)^3 + \ldots + (TS)^k]
\]

\[= [I + TS + (TS)^2 + (TS)^3 + \ldots + (TS)^{k-1}]T
\]

\[- [TS + (TS)^2 + (TS)^3 + \ldots + (TS)^k]
\]

\[= [I + TS + (TS)^2 + (TS)^3 + \ldots + (TS)^{k-1}] (T - TS).
\]  

(2.19)

(2.20)

(2.21)

Note that \( T, S \) (by (C1)) and \( I \) are circulant matrices, so lemma 2(a) implies \( T_k \) is circulant. It then follows from lemma 2(c) that \( \lambda_j(T_k^T T_k) = \lambda_j(T_k^T) \lambda_j(T_k) \). Next, note that

\[A = circ(a_0, a_1, \ldots, a_{n-1}) \implies A^T = circ(a_0, a_{n-1}, a_{n-2}, \ldots, a_1),\]

so denoting the eigenvalues of \( A \) by \( \lambda_j \) and the eigenvalues of \( A^T \) by \( \hat{\lambda}_j \), we have by lemma 2(b),

\[
\lambda_j = a_0 + \sum_{k=1}^{n-1} a_k \exp \left( \frac{2\pi i j k}{n} \right)
\]

and

\[
\hat{\lambda}_j = a_0 + \sum_{k=1}^{n-1} a_{n-k} \exp \left( \frac{2\pi i j k}{n} \right) = a_0 + \sum_{l=1}^{n-1} a_l \exp \left[ \frac{2\pi i j (n - l)}{n} \right] = \lambda_{n-j},
\]

46
where the last equality uses that
\[
\exp\left[\frac{2\pi ij(n-l)}{n}\right] = \exp\left[\frac{2\pi ij(-l)}{n}\right] = \exp\left[\frac{2\pi i(-j)l}{n}\right] = \exp\left[\frac{2\pi i(n-j)l}{n}\right].
\]

Hence, \(\lambda_j(T_k^T T_k) = \lambda_j(T_k^T)\lambda_j(T_k) = \lambda_{n-j}(T_k)\lambda_j(T_k) = |\lambda_j(T_k)|^2\). We will need this later to get \(||T_k|||\). Continuing from (2.19) and applying lemma 2(c) give

\[
\lambda_j(T_k) = \lambda_j \left( I + TS + (TS)^2 + (TS)^3 + \ldots + (TS)^{k-1} \right) \cdot \lambda_j(T - TS)
\]

\[
= \left[ \lambda_j(I) + \lambda_j(TS) + \lambda_j((TS)^2) + \lambda_j((TS)^3) + \ldots + \lambda_j((TS)^{k-1}) \right] \cdot [\lambda_j(T) - \lambda_j(TS)]
\]

\[
= [1 + t_j s_j + (t_j s_j)^2 + \ldots + (t_j s_j)^{k-1}] t_j (1 - s_j).
\]

where \(\lambda_j(T) = t_j\), \(\lambda_j(S) = s_j\) are the \(j\)-th eigenvalues of \(T\) and \(S\) respectively and \(\lambda_j(I) = 1\). Note that the indexing strictly follows that of lemma 2(b). Now we take modulus on both sides to get

\[
|\lambda_j(T_k)| = \left| [1 + t_j s_j + (t_j s_j)^2 + \ldots + (t_j s_j)^{k-1}] t_j (1 - s_j) \right|.
\]

Let \(M\) be the index where \(|\lambda_j(T_k)\)| is maximised, i.e. \(\max_j |\lambda_j(T_k)| = |\lambda_M(T_k)|\) and let \(t = t_M\) and \(s = s_M\) be the corresponding eigenvalues of \(T\) and \(S\). Then we have

\[
|\lambda_M(T_k)| = \left| [1 + ts + (ts)^2 + \ldots + (ts)^{k-1}] t (1 - s) \right|
\]

\[
\leq \left| 1 + ts + (ts)^2 + \ldots + (ts)^{k-1} \right| \cdot |t| \cdot |(1 - s)|
\]

\[
\leq \left[ 1 + |t||s| + (|t||s|)^2 + \ldots + (|t||s|)^{k-1} \right] \cdot |t| \cdot |(1 - s)|
\]

\[
= \frac{1 - (|t||s|)^k}{1 - |t||s|} \cdot |t| \cdot |(1 - s)| \quad \therefore (C2) \text{ implies } |t||s| < 1
\]

\[
\leq 1
\]

In the last inequality above, we have used (C2) in the following ways:

1. (first numerator) \(0 \leq |t||s| < 1 \Rightarrow 1 \geq 1 - (|t||s|)^k\),

2. (second term) \(|t| \leq 1\),

3. (the rest) \(|t| \leq 1 \Rightarrow |t| \cdot |s| \leq |s| \Rightarrow 1 - |t| \cdot |s| \geq 1 - |s| \Rightarrow 1 \geq \frac{1 - |s|}{1 - |t||s|}\)
Finally, by lemma 1 and the result we referred to earlier, \( \lambda_j(T_k^T T_k) = |\lambda_j(T_k)|^2 \), we have
\[
||T_k||^2 = \lambda_L(T_k^T T_k) = |\lambda_M(T_k)|^2 \leq 1 \Rightarrow ||T_k|| \leq 1
\]
as required.

\[\square\]

### 2.7.4 R code

This section contains an R implementation of Theorem 1 and Corollary 1.

**Helper functions**

```r
# install.packages("purrr")  # Requires the "purrr" package

# Helper functions
mid_pt <- function(x) ceiling(length(x) / 2)
# Example
# > mid_pt(c(-2, -1, 0, 1, 2))
# [1] 3

defux <- function(x) c(x, rev(head(x, -1)))  # Example
# > defux(c(0.05, 0.20, 0.50))
# [1] 0.05 0.20 0.50 0.20 0.05

full_to_right_half <- function(x) x[(mid_pt(x) + 1):length(x)]
# Example
# > full_to_right_half(c(0.05, 0.20, 0.50, 0.20, 0.05))
# [1] 0.20 0.05
```

**Implementation of Theorem 1**

```r
# Main functions
eigenfun <- function(ws) {
  Vectorize(function(theta) {
    k <- (length(ws) - 1) / 2
    sum(ws * cos(-k:k * theta))
  })
}
```
A_theta <- function(ws, d) {
  Vectorize(function(theta) {
    k <- (length(ws) - 1) / 2
    a_j <- full_to_right_half(ws)
    P_i <- function(i) {
      (-1)^(i-1) * sum(a_j * seq(k)^(2*i)) / factorial(2*i-1)
    }
    P_d <- function(d) {
      ind <- which(a_j < 0)
      sum(a_j[ind] * ind^(2*d + 2)) / factorial(2*d + 1)
    }
    res_sum <- 0
    for (i in 1:d) {
      res_sum <- res_sum + theta^(2*i-1) * P_i(i)
    }
    res_sum <- res_sum + theta^(2*d + 1) * P_d(d)
    -2 * res_sum
  })
}

# This function takes a half filter as input and checks if the
# filter satisfies the conditions given in Theorem 1. See the
# usage in the next section.

test <- function(hs) {
  ws <- left_half_to_full(hs)
  safe_uniroot <- purrr::safely(uniroot)
  d <- 2
  phi <- safe_uniroot(A_theta(ws, d), lower = 0.01, upper = 2*pi)
  while (!is.null(phi$error)) {
    d <- d + 2
    phi <- safe_uniroot(A_theta(ws, d), lower = 0.01, upper = 2*pi)
  }
  phi <- phi$result$root

  s <- seq(phi, 2*pi - phi, length.out = 100)
  h <- s[2] - s[1]
  lambda_f <- eigenfun(ws)
  grid_pts <- purrr::map_dbl(s, lambda_f)
}
k <- (length(ws) - 1) / 2
L <- sum(abs(ws) * abs(-k:k))
test_result <- all(grid_pts + L * (h / 2) <= 1)

# Plot Diagnostic
curve(lambda_f, 0, 2 * pi,
      xlab = expression(theta),
      ylab = expression(lambda(theta)))
points(s, grid_pts, pch = 19)
abline(v = phi)
abline(v = 2*pi - phi)

list(
    kernel_weights = ws,
    evaluation = cbind(s, grid_pts),
    Lipschitz_constant = L,
    phi = phi,
    eigenfun = eigenfun(ws),
    A_theta = A_theta(ws, d),
    grid_size = h,
    test_result = test_result
)

# Extra diagnostic
d_eigenfun <- function(ws) {
  Vectorize(function(theta) {
    k <- (length(ws) - 1) / 2
    sum((-1) * (-k:k) * ws * sin(-k:k * theta))
  })
}
derivative_diagnostic <- function(v1) {
  par(mfrow = c(2, 1))
  test_res <- test(v1)
  dL <- d_eigenfun(test_res$kernel_weights)

  s <- seq(0, 2 * pi, 0.01)
Implementation of Corollary 1

# This section confirms that the linear filters listed
# in Corollary 1 satisfies that $||A|| \leq 1$, where $A$ is
# the corresponding transformation matrix.
#
# The `test` function returns a list of attributes. The
# `test_result` attribute indicates whether Theorem 1
# holds for the particular filter.

# Henderson's filter
v1 <- c(-325, -468, 0, 1100, 2475, 3600, 4032) / 16796
test(v1)
derivative_diagnostic(v1)

# Quadratic and cubic, 7 point
v2 <- c(-2, 3, 6, 7) / 21
test(v2)
derivative_diagnostic(v2)

# Quadratic and cubic, 13 point
v3 <- c(-11, 0, 9, 16, 21, 24, 25) / 143
test(v3)
derivative_diagnostic(v3)
# Spencer's 15-point moving average

```r
v4 <- c(-3, -6, -5, 3, 21, 46, 67, 74) / 320
test(v4)
derivative_diagnostic(v4)
```

# Spencer's 21-point moving average

```r
v5 <- c(-1, -3, -5, -5, -2, 6, 18, 33, 47, 57, 60) / 350
test(v5)
derivative_diagnostic(v5)
```

# Quartic and quintic, 7-point

```r
v6 <- c(5, -30, 75, 131) / 231
test(v6)
derivative_diagnostic(v6)
```

# Quartic and quintic, 13-point

```r
v7 <- c(110, -198, -135, 110, 390, 600, 677) / 2431
test(v7)
derivative_diagnostic(v7)
```
Chapter 3

Combinatorial Model Selection via Gibbs stochastic search

Chapter Summary
We propose a novel model selection procedure based on the Gibbs sampler and classical limit theorems in probability theory. The algorithm belongs to the class of stochastic-search algorithms and is designed to solve problems with exponentially large solution space. Looking for a single model in an exponentially large space is difficult, the core idea of this chapter is that every “wrong” model that we evaluate should tell us something about the true model, and by aggregating all these information, one should be able to recover the true model without exhausting the entire model space. This distinguishes our algorithm from other existing ones in that our algorithm utilizes the hierarchical structure of the model space and all the models it has evaluated to infer what the best model should be, so the selected model is often a model that has not been evaluated. We show the statistical consistency of such selection and benchmark our method against Generalised Linear Model with Elastic Net (glmnet).

3.1 Introduction
Classical statistics is founded on asymptotic analysis, where validity of a statistical model is considered in the limit. In practice, applying asymptotic methods to finite data could introduce a form of bias called over-fitting, i.e. a model fits the existing data too well that it fails to generalise to new data. Model selection was motivated to address this problem; it adjusts the methods for finite data by balancing between model fit and model complexity.
Much work on model selection were done in the 1970s and 1980s with heightened interest around 2000s, driven by the need to improve predictive performance, handle high-dimensional (and potentially sparse) data and produce more interpretable results. In the context of linear regression / classification models, three popular selection principles are subset selection, information criterion and regularisation. Subset selection searches through the model space and finds a best model with respect to some performance metric. As the model space has cardinality $2^p$, where $p$ is the number of features, it is prohibitive to do an exhaustive search even for a moderate size of $p$, and it is common to use greedy methods. This includes forward-stepwise selection, forward-stagewise selection and backward-step selection for example. On the other hand, modern adaptation includes stochastic methods like simulated annealing, genetic algorithm, swarm optimisation and cross-entropy. These are powerful general-purpose optimisers and they have a wide range of applications in many areas. Information criterion is developed based on information theory (specifically via different approximations of Kullback-Leibler divergence); it balances model fit with model complexity by explicitly penalising the former by the latter. Usage in practice involves pre-specifying a set of candidate models, then selecting the one with the optimised information criterion. This class of methods includes the well-known Akaike Information Criterion(AIC), Bayesian Information Criterion(BIC), Deviance Information Criterion(DIC) and Minimum Description Length(MDL). Regularisation places constraints on the size of the regression coefficients, introducing bias towards simpler model to achieve balance between model fit and complexity. Two representative in this class of methods are the $L_p$ regularisation (including important sub-cases $L_1$ Lasso regression and $L_2$ Ridge regression) and elastic-net regularisation. Excellent reviews of the above methods can be found in Claeskens et al. (2008), Friedman et al. (2001) and Spall (2005).

In this chapter, we are concerned with the following model selection problem: consider the generalised linear model (GLM) in the usual regression or classification setting, and suppose we have $n$ data points and $p$ predictor variables. As each subset of the $p$ predictors defines a model, there are $2^p$ models in total. We apply a likelihood-based metric (e.g. maximised likelihood, AIC or BIC) on the models, and we are interested in finding the model with the highest score, referred to as the ‘best model’. When the true data generating process is in the model class considered, then the best model is the true model; otherwise, this model is closest (within the model class) to the true model in terms of Kullback-Leibler divergence.
In general, it is computationally infeasible to evaluate all the models. In this chapter, we propose a stochastic search algorithm based upon the Gibbs sampler and classical limit theorems in probability theory. The key idea is that the likelihood function of a GLM places a hierarchical structure on the model space. As a result, models that are “close” to the true model (e.g. models that nests the true model) have performance comparable to the true model. We examine closely and make precise this comparability. In particular, “good” models have log-likelihood $O(\log \log n)$ away from the best model, and “bad” models have log-likelihood $O(n)$ away from the best model. The model hierarchy and the explicit log-likelihood order give us an algorithm with the distinguishing feature that it could select a model that has not been evaluated. We introduce the algorithm and show the (statistical) consistency of such selection in Section 3.2 and conduct numerical studies benchmarking our method against the Generalised Linear Model with Elastic Net (glmnet) in Section 3.3. Full details of the technical proof are given in the Appendix.

3.1.1 Notations

We use the following notations for the rest of this chapter.

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>$X$</td>
<td>Data matrix</td>
</tr>
<tr>
<td>$n$</td>
<td>Number of data points</td>
</tr>
<tr>
<td>$p$</td>
<td>Number of features / covariates</td>
</tr>
<tr>
<td>$M = [a_1 a_2 \cdots a_p]$</td>
<td>A model that specifies the inclusion/exclusion of predictors, encoded in a binary sequence (1 is inclusion and 0 otherwise)</td>
</tr>
<tr>
<td>$s(A)$ or $s_A$</td>
<td>Goodness-of-fit score of a model $A$</td>
</tr>
<tr>
<td>$\mathcal{M}_c$</td>
<td>The set of all correct models</td>
</tr>
<tr>
<td>$\mathcal{M}_w$</td>
<td>The set of all wrong models</td>
</tr>
<tr>
<td>$\mathcal{M}_{cs}^{(i)}$</td>
<td>The set of all correct sub-models in block $i$</td>
</tr>
<tr>
<td>$\mathcal{M}_{ws}^{(i)}$</td>
<td>The set of all wrong sub-models in block $i$</td>
</tr>
<tr>
<td>$B_0$</td>
<td>The set of predictors in the best model</td>
</tr>
<tr>
<td>$\beta_0$</td>
<td>The regression coefficients of the best model</td>
</tr>
<tr>
<td>$</td>
<td></td>
</tr>
</tbody>
</table>

55
3.2 The Gibbs stochastic search algorithm

We consider a sampling-based approach for the model selection problem. First, we recast the model selection problem into a density estimation problem in Section 3.2.1. Then, we use the Gibbs sampler to estimate the required density; we introduce the Gibbs sampler in Section 3.2.2 and formally construct a Markov Chain with stationary distribution equal to the target density in Section 3.2.3. Based on that, we develop our model selection algorithm and show the consistency of the model selected by our algorithm in Section 3.2.4 and 3.2.5.

3.2.1 From model selection to density estimation

Continuing with our previous set-up of selecting the best model out of $2^p$ models, suppose we have defined a measure of fit, evaluated all $2^p$ models and get a score for each model. We index the models by $\{i\}_{i \in I}$ and the corresponding scores by $\{s_i\}_{i \in I}$, where $I = \{1, 2, ..., 2^p\}$ is the index set.

By using a softmax function, we could turn these scores into probabilities. Specifically, we define

$$\hat{s}_i = \frac{e^{\lambda s_i}}{\sum_{i \in I} e^{\lambda s_i}}, \quad i \in I,$$

(3.1)

where $\lambda$ is a tuning parameter, and this gives us a probability mass function (pmf) over the $2^p$ models. We denote this pmf by $T$ and refer to it by the target density. It follows that selecting the best model corresponds to finding the point where the probability is maximised, and this can be readily solved if we know the pmf $T$ completely. Keen readers may object here that finding the full density seems much more difficult than finding an extrema, indeed it is, but we remark that our algorithm only tries to find the extrema “on the way” to finding the density, and accurate estimation of the density is not required.

3.2.2 Gibbs sampler

To estimate the required density, we will use the Gibbs sampler to produce a Markov Chain with stationary distribution equal to the target density. Gibbs sampler was proposed by Geman and Geman (1984) to model spatial data and later adapted to do Bayesian inference. The algorithm produces samples from the target joint distribution by sampling a sequences of samples from the conditional distributions. Suppose $(X_1, X_2, ..., X_p)$ has joint density $f(x_1, ..., x_p)$ with
full conditionals $f_1, f_2, ..., f_p$, then the Gibbs sampler simulates a sample as follows (Robert and Casella, 2004, p.372):

- Initialise $x^{(0)} = (x^{(0)}_1, x^{(0)}_2, ..., x^{(0)}_p)$, then

- for $t = 0, 1, 2, ...

  1. Sample $X_{1}^{t+1} \sim f_1(x_1 | x^{(t)}_2, x^{(t)}_3, ..., x^{(t)}_p)$

  2. Sample $X_{2}^{t+1} \sim f_2(x_2 | x^{(t+1)}_1, x^{(t)}_3, ..., x^{(t)}_p)$

  \vdots

  p. Sample $X_{p}^{t+1} \sim f_p(x_p | x^{(t+1)}_1, ..., x^{(t+1)}_{p-1})$

For sufficiently large $t$, $x^t = (x^t_1, x^t_2, ..., x^t_p)$ is a sample from $f(x_1, ..., x_p)$.

### 3.2.3 MCMC Construction

Now we construct a Markov Chain that has stationary distribution equal to the target density $\mathcal{T}$. First, we need to define the states and the transition matrix of the Markov Chain. Since the target density is defined on the space of models, where each point represents a model, so accordingly the states of our Markov Chain are the models. We encode the models by binary sequences, where 1 represents inclusion of a predictor and 0 represents exclusion of a predictor. For example, suppose the first three of the $p$ predictors are included to form a model, then the model is represented by $(x_1 = 1, x_2 = 1, x_3 = 1, x_4 = 0, ..., x_p = 0)$ and abbreviated as $\{11100000\}$. Given a model $M$, we also define the conjugate model $\overline{M}_k$ to be the model that has a binary representation same as $M$ except for the $k$-th bit.

For the transition matrix, as the Gibbs sampler updates only one bit in each step and $p$ bits over one iteration of $p$ steps, so it is possible to transit from one model to any other model in one iteration. Suppose $M_i = \{a_1 a_2 \cdots a_p\}$ and $M_j = \{b_1 b_2 \cdots b_p\}$ are two model sequences. For $k = 0, ..., p$, we define the transition sequence $T^{(k)}_{ij}$ from $M_i$ to $M_j$ by

$$T^{(k)}_{ij} = \begin{cases} M_i, & \text{for } k = 0 \\ \{b_1 b_2 \cdots b_k a_{k+1} \cdots a_p\}, & \text{for } k = 1, ..., p - 1 \\ M_j, & \text{for } k = p \end{cases} \tag{3.2}$$

in other words, $T^{(k)}_{ij}$ is the sequence of models one has to get through to go from $M_i$ to $M_j$:

$$M_i = T^{(0)}_{ij} \rightarrow T^{(1)}_{ij} \rightarrow T^{(2)}_{ij} \rightarrow ... \rightarrow T^{(p-1)}_{ij} \rightarrow T^{(p)}_{ij} = M_j$$
To transit from model $i$ to model $j$, one starts at $T_{ij}^{(0)}$, then transits $p$ times to reach $T_{ij}^{(p)}$. If we define the probability of going from $T_{ij}^{(k-1)}$ to $T_{ij}^{(k)}$ to be $t_k, k = 1, ..., p$, then the probability of transiting from model $i$ to model $j$ is given by the product $p_{ij} = \prod_{k=1}^{p} t_k$. Keeping in mind that we are constructing a Markov Chain with target density $\mathcal{T}$, we set $t_k$ to be

$$t_k = \frac{e^{\lambda s(T_{ij}^{(k)})}}{e^{\lambda s(T_{ij}^{(k)})} + e^{\lambda s(T_{ij}^{(k)})}},$$

where $\lambda > 0$ is some tuning parameter, and then the required transition matrix is given by

$$
\begin{pmatrix}
p_{11} & p_{12} & p_{13} & \cdots & p_{1m} \\
p_{21} & p_{22} & p_{23} & \cdots & p_{2m} \\
p_{31} & p_{32} & p_{33} & \cdots & p_{3m} \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
p_{m1} & p_{m2} & p_{m3} & \cdots & p_{mm}
\end{pmatrix}, \quad p_{ij} = \prod_{k=1}^{p} t_k
$$

where $m = 2^p$ and again $p_{ij}$ is the transition probability from model $i$ to model $j$.

**Theorem 3.** The stationary distribution of the Markov Chain constructed above is the target density $\mathcal{T}$.

The existence and convergence to the joint (stationary) distribution of MCMC algorithms constructed from full conditionals (i.e. the convergence of Gibbs’ sampler) are well-known. For sake of completeness, we give a proof specific to our construction in the Appendix. For a general proof, see Geman and Geman (1984) or Roberts and Smith (1994).

### 3.2.4 Gibbs stochastic search

From the construction, as the variables of interest are the inclusion / exclusion of predictors, i.e. only take values 0 or 1, we see that the distribution over the $2^p$
models has Bernoulli distribution as conditionals, and it is simple to write down the Gibbs sampler for it. The algorithm is as follows:

**Algorithm 2: Gibbs sampler for best-subset selection**

**Input:** Number of burn-ins $b$; number of iterations $m$.

Initialise randomly $I_0 = (i_1(0), i_2(0), \ldots, i_p(0))$, $i_k(0) \in \{0, 1\}$ for $k = 1, \ldots, p$.

for $j = 1$ to $b + m$ do

for $k = 1$ to $p$ do

$M_0 = (i_1^{(j)}, i_2^{(j)}, \ldots, i_{k-1}^{(j)}, 1, i_{k+1}^{(j)}, \ldots, i_p^{(j-1)})$,

$M_1 = (i_1^{(j)}, i_2^{(j)}, \ldots, i_{k-1}^{(j)}, 0, i_{k+1}^{(j)}, \ldots, i_p^{(j-1)})$

Draw $i_k^{(j)}$ from $\text{Bern}(p)$, where $p = \frac{e^{s(M_0)}}{e^{s(M_0)} + e^{s(M_1)}}$

Update $I_j = (i_1^{(j)}, i_2^{(j)}, \ldots, i_{k-1}^{(j)}, i_k^{(j)}, i_{k+1}^{(j)}, \ldots, i_p^{(j-1)})$

end

if $j > b$ then

Record $I_j$

end

return \{ $I_j, j = b + 1, \ldots, b + m$ \}

So far, we have merely adapted Gibbs sampler to the problem of model selection, and one can view this simply as a random walk in the model space driven by the performance landscape. The novelty actually lies in the model selection methodology: it selects a model based on the (full) path the sampler has travelled through. We shall discuss this in full in the next section.

### 3.2.5 Model selection criteria

The Gibbs sampler gives us a way to simulate a sequence of models from the model space, we still need to specify how a model is selected. We describe three ways for this task and show the (statistical) consistency of the selection. (We remark that the first two criteria are the naive approach, and readers looking for novelty can skip ahead to the third criterion directly.)

**Best score**

A simple way to select a model is to adopt the ‘best score’ strategy, i.e. as we run the Markov Chain, we keep track of the model that has the best score. This is basically brute-force except that the search is guided by the score rather than blindly evaluating the whole model space.

**Theorem 4.** The model selected by the best-score strategy converges to the best model almost surely as the number of MCMC steps, $m$, goes to infinity.
Proof. First it is clear once the best model is reached, the algorithm would never update again since no other model has better score than the best model. It remains to show the probability that the best model cannot be reached is 0. In other words, the best model must be attained with probability 1, but we know the model space will eventually be exhausted, so this is clearly true.

Formally, we order the models by their performance so that the best model has the last index $J = 2^p$. Given that the chain is in state $i$, the probability of reaching the best model is $p_{iJ}$; if we define $q = \min_{i \in I} p_{iJ}$, then it follows the probability of not reaching the best model in $m$ steps is at most $r_m = (1 - q)^m$. Since by construction $p_{ij} > 0, \forall i, j \in I$, which implies $(1 - q) < 1$, so $r_m \to 0$ as $m \to \infty$. The rate of convergence (in probability) is geometric, and applying Borel-Cantelli lemma yields the almost-sure convergence.

Highest frequency

Given that the best model has the highest score, we see from (3.1) that it also has the highest probability mass, so as we sample from the stationary distribution, we could pick the model that occurs most frequently.

Theorem 5. The model selected by the highest-frequency strategy converges to the best model almost surely as the number of MCMC steps, $m$, goes to infinity.

Proof. By construction.

Thresholding marginal relative frequencies

Here comes the final strategy which is also the selection strategy we advocate. The issue with the last two strategies is that while they are correct, they do not solve the problem of exploring an exponentially large space in a practical sense as they are to some degree akin to the brute-force method. To describe the third strategy, we shall start with two definitions.

Definition 4. A correct model is any model that nests the best model. In other words, a correct model includes all predictors of the best model and potentially more.

Definition 5. A wrong model is any model that is not a correct model, i.e. a wrong model is a model that misses at least one of the predictors of the best model.

We now describe the model selection procedure. Suppose we perform $m$ iterations of update and get $m$ models, each of which is represented by the binary
sequence: \( M^{(k)} = (a_1^{(k)} a_2^{(k)} \cdots a_p^{(k)}) \), \( k = 1, 2, \ldots, m \). We compute the marginal relative frequencies as follows:

\[
f_i = \frac{1}{m} \sum_{j=1}^{m} a_i^{(j)}, \quad i = 1, 2, \ldots, p.
\]

It is useful to imagine a matrix where each row is the model \( M^{(k)} \), then the marginal relative frequencies is simply the column mean. Next, we threshold these relative frequencies by \( \gamma \) and select

\[
M^{final} = (1_{\{f_1 > \gamma\}} 1_{\{f_2 > \gamma\}} \cdots 1_{\{f_p > \gamma\}})
\]

as the final model, where \( 1_{\{\cdot\}} \) is the indicator function. We claim this model converges to a correct model (and also eventually to the best model). Before we state the consistency theorem and its proof, we describe the key feature of this strategy.

The most interesting feature and also the main contribution of this chapter is that the final model selected may be (and in practice it is usually the case) a model that has not been evaluated. This distinguishes our method from the existing optimisation methods like simulated annealing and genetic algorithms, where the model selected must have been evaluated. The fundamental difference is that the other algorithms do not know what the best model would look like and keep on searching using good heuristic and trial-and-error, while our algorithm (using the third strategy) over iterations would build up information about the best model; in fact, it utilises the model space hierarchy and the full path of the sampler to identify a correct model. To elaborate, simulated annealing is essentially a random walk in the model space with a (temperature) parameter to control the scope of the exploration. The search of models is based solely on the performance of the models, and it uses no structural information from them. Genetic algorithm creates ‘offspring’ models from the high-performing models using some genetically inspired mechanisms, and structural information from the ‘parents’ models are used to create new models. Our algorithm, Gibbs stochastic search, aggregates the information from all the evaluated models (i.e. the full path, excluding the burn-ins) to produce one model. Unlike the other algorithms, we know without evaluation that this model is guaranteed in a statistical sense to be a good model. Intuitively, every wrong model we evaluated tells us something about the best model, and Gibbs sampler provides the right mathematical formulation and tools we need.
**Theorem 6.** Let \( p_c \) be the probability that a model sampled from the stationary distribution is a correct model, and \( p_w = 1 - p_c \) be the probability that the sampled model is a wrong model. Then, \( p_c/p_w \) is unbounded as the number of datapoints \( n \) goes to infinity.

**Proof outline:** (The detailed proof is given in the Appendix.) The key observation is that since all the correct models nest the best model so they should have goodness-of-fit comparable to the best model, while the wrong model missed at least one covariate of the best model, hence as the number of samples increases, the goodness-of-fit would suffer. Specifically, we will show (in the Appendix) that the difference in the likelihood scores between the best model and any correct model is of order \( \log \log n \), while between the best model and any wrong model, it is of order \( n \), suggesting that the probability of drawing a specific correct model is much greater than that of drawing a specific wrong model. Then it remains to show this is still the case after adjusting for the number of models in each class.

Suppose out of the \( p \) covariates, the best model contains \( q \leq p \) of them, then the number of correct models and wrong models are \( 2^p - q \) and \( 2^p - 2^p - q \) respectively; we denote the two classes of models by \( \mathcal{M}_c \) and \( \mathcal{M}_w \). Since \( p \) and \( q \) are fixed, hence for some constants \( k_1, k_2 > 0 \) we have

\[
\frac{p_c}{p_w} = \frac{\sum_{i \in \mathcal{M}_c} \hat{s}_i}{\sum_{j \in \mathcal{M}_w} \hat{s}_j} > \frac{2^{p-q} \cdot \min_{i \in \mathcal{M}_c} \hat{s}_i}{(2^p - 2^p - q) \cdot \max_{j \in \mathcal{M}_w} \hat{s}_j} \geq e^{\lambda(n_k - k_2 \log \log n)} \tag{3.6}
\]

is unbounded. (Note that in the above we defined \( \hat{s}_c_{\min} := \min_{i \in \mathcal{M}_c} \hat{s}_i, \hat{s}_w_{\max} := \max_{j \in \mathcal{M}_w} \hat{s}_j \).)

**Corollary 2.** For a sufficiently large number of datapoints \( n \), the selection strategy of thresholding marginal relative frequencies would select a correct model as the number of MCMC steps / samples \( m \) goes to infinity.

Corollary 2 follows from Theorem 6 because if the ratio of probabilities in (3.6) is unbounded, then we would expect to draw (from the stationary distribution) many more correct models than wrong models. This results in high marginal frequencies of the correct predictors, which upon thresholding would yield the result.

Formally, for a fixed threshold \( \gamma \) and a sufficiently large \( n \), we have \( p_c > \gamma \). It then follows that if we draw \( m \) samples (after stationarity is reached) from the model space and select the model \( M_{\text{selected}} \) by thresholding the marginal frequen-
cies, then

\[ P(\text{selected is correct}) \geq P\left( \frac{\sum_{i=1}^{m} Z_i}{m} > \gamma \right) \xrightarrow{a.s.} 1, \tag{3.7} \]

where \( Z_i \sim \text{Bern}(p = p_c), i = 1, 2, \ldots, m \) are i.i.d. Bernoulli random variables, representing whether a correct model is drawn, and for the convergence, we used that \( \sum_{i=1}^{m} Z_i m \xrightarrow{a.s.} p_c > \gamma \) by the Strong Law of Large Number.

Furthermore, since all the correct models must contain the best model, the marginal frequencies would eventually also reveal the best predictors. To see this, note that Theorem 6 says for sufficiently large data, one would almost always draw a correct model. For the sake of illustration, suppose the samples we get are all correct models. Then since every correct model by definition must contain all the best predictors, the best predictors must have the highest marginal frequencies of 1, while the other predictors have marginal frequencies (generally) less than 1. Hence the best model can be identified by thresholding again. As a side note, this effect would be more pronounced in the sparse situation, i.e. when \( q << p \).

**Discussion**

The result above gives important insights about the nature of finding a particular solution in a exponentially large space. The takeaways are as follows:

- Randomly searching for one particular model out of an exponentially large space using only local information - with Simulated Annealing in mind - is very difficult. Even driven by the landscape, it is unlikely one runs into the best model as the space is too large.

- The selection, crossover and mutation mechanism used by Genetic Algorithm does incorporate structural information, and it works well in practice. However, it does not use information from the optimisation landscape, so generally there is no guarantee the crossed-over-and-mutated offspring is good. In fact, the non-convergence is shown in Rudolph (1994).

- Having acknowledged the weakness above, we advocate the approach that expands the target, from the single best model to the entire class of correct models (i.e. from size 1 to size \( 2^{p-q} \)), and then infer the original target from there (by thresholding marginal frequencies). This breaks the original hard search problem into a much simpler search problem plus a “deduction” problem. Our approach uses both the local information (during the MCMC
transition) and the structural information (thresholding the model path), and it comes with consistency guarantee.

• Taking a high-level point of view, the key message is that one does not need to actually reach the best model to find it, it is sufficient to reach a large number of its neighbors (w.r.t. the Hamming metric in our case). Perhaps this may be summarised by the quote “If you cannot define yourself, then define your surroundings”.

• The proposed approach may be extended to other regression models and score functions provided that similar structure can be established out of the model space. The key construction is to find $M_c$ such that

$$m_{\text{best}} = \bigcap_{m \in M_c} m \quad \text{and} \quad s(m_{\text{best}}) > s(c) >> s(w), \quad \forall c \in M_c, w \in M_w = M_{c'},$$

where $m_{\text{best}}$ denotes the best model.

### 3.2.6 Parallelisation

MCMC iterations are generally computationally intensive. It is natural to consider parallelisation to speed up performance. While parallelisation has not been possible for MCMC procedure due to its sequential nature, it is possible to parallelise the model selection component to achieve significant speed up.

Our proposal is as follows. First, given $p$ predictors, we partition them into $k$ groups, run our Gibbs stochastic search (in parallel) on all of them and collect the selected predictors. Then, we pool together the predictors from all the groups to form the final model. From a model space perspective, grouping the predictors into blocks is essentially restricting each sampler to explore the respective local subspace. It turns out the structure imposed by the likelihood function is so strong that the model selection would still work when applied to each of the subspace, despite that the proof is more involved.

We require additional definitions to show the consistency of the selected model. Continuing with our set-up of partitioning $p$ predictors into $k$ blocks, for each block, the full sub-model is (defined to be) the model that includes all the predictors in that block (i.e. it does not contain any other predictors in other blocks). A correct sub-model is any model that includes all the true predictors in that block, while a wrong sub-model is any model that misses at least one true predictors in the block. Note that a correct sub-model can be (and usually will be) a wrong model as defined in Definition 5; it is only “correct” with respect to the block that it is in. Some examples are given in Table 3.1 to aid reading.
<table>
<thead>
<tr>
<th>Models</th>
<th>Model encoding</th>
</tr>
</thead>
<tbody>
<tr>
<td>The best model</td>
<td>-1----1-11-</td>
</tr>
<tr>
<td>The full model (i.e. include all predictors)</td>
<td>11111111111</td>
</tr>
<tr>
<td>A correct model (i.e. include all predictors in the best model and potentially more)</td>
<td>-11--11-111</td>
</tr>
<tr>
<td>A wrong model (i.e. misses at least one predictors in the best model)</td>
<td>111--1-11--</td>
</tr>
<tr>
<td>A correct sub-model in 1st block (say block size is 4)</td>
<td>11-1---------</td>
</tr>
<tr>
<td>A wrong sub-model in 1st block</td>
<td>1--1---------</td>
</tr>
<tr>
<td>A correct sub-model in 2nd block</td>
<td>----1111----</td>
</tr>
<tr>
<td>A wrong sub-model in 2nd block</td>
<td>----1--1----</td>
</tr>
</tbody>
</table>

Table 3.1: Examples of encoding of models (of 11 predictors) (‘0’ is replaced by ‘-’ to improve readability)

The main modification needed to make Theorem 6 work for the parallelised case is this: instead of comparing the correct models and the wrong models through the best model, we compare the correct sub-models with the wrong sub-models directly. It turns out that a correct sub-model has a score higher than a wrong sub-model by a order of $n$; in fact, this property holds more generally, i.e. models at a higher position of the hierarchy would have score larger than the ones at a lower position by a order of $n$. This gives us just enough to establish the result, despite that we do not have a complete ordering among the models. The full proof is given in the Appendix. We will now turn to the numerical studies.

### 3.3 Numerical studies

The goal of this section is to test whether the proposed algorithm can identify the true predictor variables in a large pool of variables. Ideally, the algorithm identifies all true predictors without including any false ones. In the imperfect case, we generally prefer more predictors including all the true ones over less predictors missing some of the true predictors. For example, suppose the number of true predictors is 10, then between

(i) a set of 15 predictors including all 10 true predictors and 5 false predictors and

(ii) a set of 10 predictors including 9 true predictors and 1 false predictor,

we would favor (i) over (ii). The motivation here is to avoid leaving out any true predictors as any one of them could be important. For comparison, we benchmark
our method against glmnet (Zou and Hastie (2005)), a popular variable selection method for high-dimensional data.

We conduct our study under two scenarios. In the first case, both the predictor and response variables are simulated; we refer to this case as the fully-simulated case. In the second case, the predictors are taken from a real dataset, and only the response variable is simulated; we refer to this case as the mixed-simulated case.

The simulation consists of two parts, simulating the data and selecting the model. In the fully-simulated case, assuming sparsity, we first simulate 10 predictor variables from some joint distributions, then we generate the response variable using the Generalized Linear Model (GLM). This gives us a dataset containing the response variable and the true predictors. Next, we add in 1990 false predictors, i.e. predictors unrelated to the response variable, and “hide” the true ones among them by permuting the columns of the dataset. Figure 3.1 illustrates this setup. Note that both the true and false predictors are simulated using the same procedure.

In the mixed-simulated case, instead of simulating 10 predictor variables, we randomly draw 10 predictor variables from a real dataset, then we generate the response variable using a GLM and mix in the remaining predictor variables in the dataset. The dataset we use is related to breast cancer, as it is seen in Qian et al. (2016). There are 229 response records sampled from the Australian Breast Cancer Family Study (ABCFS) (Dite et al., 2003) and Australian Mammographic Density Twins and Sisters Study (AMDTSS) (Odefrey et al., 2010), each encoded by 0 and 1 where 1 indicates presence of cancer. Each record comes with 366 SNP predictor variables, encoded by 0, 1 or 2, representing the number of the minor alleles at the SNP loci. One can consult Qian et al. (2016) for more details about the dataset.

Multiple datasets are simulated to assess how well our method performs in different situations. These datasets differ in the following choices:

A. The type of predictor variables: We consider four settings, all independent numerical predictors, all independent categorical predictors, all dependent numerical predictors and all dependent categorical predictors. Numerical predictors are drawn from the multivariate normal distribution, and the categorical predictors are drawn from the discrete uniform distribution.

B. Distribution of the response variable: Gaussian family with the identity link, and binomial family with the logistic link.
C. Signal noise ratios: Each predictor $x_i$ is associated with a coefficient $\beta_i$ and a corresponding signal-to-noise ratio (SNR), defined to be $\frac{|\hat{\beta}_i|}{\sqrt{n \cdot \text{Var}(\hat{\beta}_i)}}$, where $\hat{\beta}_i$ denotes the coefficient estimate. We classify different values of SNR into three classes, ‘Large’, ‘Medium’ and ‘Small’, based on some pre-specified ranges of values. In general, one expects predictors with large SNRs are identified more consistently while predictors with small SNRs are harder to detect. We consider four SNRs settings in our simulation: (i) all Large SNRs (10L), (ii) all Medium SNRs (10M), (iii) half Large and half Medium (5L5M), and (iv) a mixture of Large, Medium and Small SNRs (4L4M2S).

Overall, we have $(2 \times 2) \times 2 \times 4 = 32$ settings.

Following the data simulation, we perform model selection. We apply glmnet and Gibbs stochastic search to each dataset and record the number of variables selected and the number of true predictors identified. Both algorithms have hyperparameters that need to be specified. For glmnet, the key tuning parameters are $\alpha$ and $\lambda$. The former controls the balance between $L_1$ and $L_2$ regularization, while the latter controls the amount of penalization. As we want to demonstrate how our method could serve as a strong alternative to glmnet, we tune the hyperparameters to glmnet’s favor. Specifically, as the sparsity assumption is made, we set $\alpha = 1$ which corresponds to LASSO regression. For $\lambda$, we consider two ways of tuning. The first is to start with the default sequence of $\lambda$’s provided by glmnet, then choose the best one according to the following procedure.
1. Pick the $\lambda$ that includes the most number of true predictors.

2. If there are ties, pick among them the one that selects the least number of variables.

3. If there are still ties, pick the one that maximises the deviance ratio.

These criteria require knowledge of the true predictors and should give a result indicative of the theoretical limit of glmnet. The second way to tune $\lambda$ is to simply use cross validation; this gives a result that reflects the performance of glmnet in practice.

For the Gibbs stochastic search, we do not tune the hyper-parameters and use the default as given by the algorithm:

1. Number of runs: 500
2. Number of burns-in: 10
3. Number of blocks: 50
4. Measure of fit: BIC
5. Selection method: Thresholding marginal frequency
6. Threshold: 0.9 (for comparison purposes, we also use 0.5 and 0.999).

### 3.3.1 Results

The results for the fully-simulated case are given in Tables 3.2 and 3.3. Table 3.2 corresponds to the independent-covariates case. We see that when $n = 10000$, both methods recover the true predictors almost perfectly although cv_min tends to choose too many irrelevant predictors, especially in the case of GLM with binomial family. When $n = 500 < 2000 = p$, both methods still do well for Gaussian case, but not so well for binomial case. In general, cv_1se seems to be a good heuristic and often outperforms cv_min; our method (at the default threshold of 0.9) is comparable to cv_1se in all settings.

Table 3.3 corresponds to the dependent-covariates case. When $n = 10000$, again cv_min tends to choose too many irrelevant predictors, while cv_1se and gibbs.0.9 perform well. When $n = 500$, the performance of both methods suffer from the lack of data. cv_1se and gibbs.0.9 have comparable performance, but gibbs.0.9 (along with gibbs.0.5 and gibbs.0.999) tends to be more stable across different scenarios.
The results for the mixed-simulated case are given in Table 3.4. Interestingly, the Gibbs class of methods outperform the ‘best’ glmnet model either by identifying more true predictors or identifying less irrelevant predictors. We remark that one example certainly does not prove anything, but together with the results from Table 3.2 and 3.3, we do see good indication that our method could complement glmnet where it falls short, may it be in the case of large $p$, small $n$ or in the case of dependent predictors.

3.4 Conclusion

In this chapter, we proposed a novel model selection procedure based upon the Gibbs sampler and classical limit theorems in probability theory. Our method belongs to the class of stochastic search algorithms and it is designed to solve problems with exponentially large solution space. Unlike its rivalries, the simulated annealing and the genetic algorithm, the model selected by our method does not have to be evaluated. Our method achieves this by utilising the model space hierarchy and aggregating information from (all) the evaluated models. We showed the consistency of the method and demonstrated numerically that it has comparable accuracy as Generalised Linear Model with Elastic Net (glmnet) and more stability across different situations.
Table 3.2: Model selection results for the independent-covariates case. The numbers shown are the number of correctly identified variables divided by the total number of identified variables (and the adjusted Rand index in bracket). Columns 5-7 are obtained with glmnet. **cv_min**: the regularisation parameter, \( \lambda \), is chosen such that the cross-validation error is minimised; **cv_1se**: \( \lambda \) is chosen such that the cross-validation error is within 1 standard error of the minimum; **best**: the approximate theoretical limit. Columns 8 - 12 are obtained using Gibbs stochastic search with different marginal frequency thresholds.
Table 3.3: Model selection results for the dependent-covariates case. The numbers shown are the number of correctly identified variables divided by the total number of identified variables (and the adjusted Rand index in brackets). Columns 5-7 are obtained with glmnet. \textbf{cv\_min}: the regularisation parameter, \(\lambda\), is chosen such that the cross-validation error is minimised; \textbf{cv\_1se}: \(\lambda\) is chosen such that the cross-validation error is within 1 standard error of the minimum; \textbf{best}: the approximate theoretical limit. Columns 8-12 are obtained using Gibbs stochastic search with different marginal frequency thresholds.
Table 3.4: Model selection results for the mixed-simulated case. The true data generating process consists of 229 records with 10 true predictors and 356 irrelevant predictors.

### 3.5 Appendix

#### 3.5.1 Preliminaries

**Theorem 7.** *(Uniform Law of Large Number (Ferguson, 1996, p.107-108))* Suppose $X_1, X_2, ...$ is a sequence of i.i.d. random variables with a common distribution, $F(x)$, let $V(x, \theta)$ be a measurable function of $x$ for all $\theta$ in some parameter space $\Theta$ and write $\mu(\theta) = \int V(x, \theta) dF(x)$.

If the parameter space $\Theta$ is compact, $V(x, \theta)$ is continuous in $\theta$ for all $x$, and there exists a function $K(x)$ such that $\mathbb{E}K(X) < \infty$ and $|V(x, \theta)| \leq K(x)$ for all $x$ and $\theta$, then

$$P \left( \lim \sup_{n \to \infty} \frac{1}{n} \sum_{i=1}^{n} V(X_i, \theta) - \mu(\theta) = 0 \right) = 1$$

**Theorem 8.** *(Law of Iterated Logarithm (Petrov, 1995, p.248))* Let $\{X_n\}$ be a sequence of independent identically distributed random variables with zero mean and finite variance $\sigma^2$. Let $S_n = \sum_{i=1}^{n} X_i$, then

$$\lim \sup_{n \to \infty} S_n/\sqrt{2n \log \log n} = \sigma \quad a.s. \quad (3.8)$$

$$\lim \inf_{n \to \infty} S_n/\sqrt{2n \log \log n} = -\sigma \quad a.s. \quad (3.9)$$

and every point of the closed interval $[-\sigma, \sigma]$ is the limit point (in the sense of almost sure convergence) for the sequence $S_n/\sqrt{2n \log \log n}$.

**Definition 6.** A transition matrix $A$ is regular if there exists $n \in \mathbb{N}$ s.t. $A^n$ has all positive entries.

**Theorem 9.** *(Kemeny and Snell, 1983, p.71)* If $P$ is a regular transition matrix and $A := \lim_n P^n$ with first row denoted as $\alpha$, then

1. for any probability vector $\pi$, $\lim_n \pi \cdot P^n = \alpha$;
2. the vector $\alpha$ is the unique probability vector such that $\alpha P = \alpha$.

3.5.2 Proof of MCMC construction (Theorem 3)

Here we show the Markov Chain we constructed has stationary solution equal to the target density $\mathcal{T}$. We show the result in two steps:

1. the Markov Chain constructed has a stationary solution, and

2. the stationary solution is the target density $\mathcal{T}$.

To show the first part, from equation (3.3) we see that $t_k$ is positive, so $p_{ij}$ is positive and the transition matrix $P = [p_{ij}]$ is regular. It then follows from Theorem 9 that the regular chain has an unique stationary distribution.

For the second part, knowing that the stationary distribution of the Markov Chain constructed from the conditional distribution is the corresponding joint distribution, it suffices to show that the full conditional distributions corresponding to the target density $\mathcal{T}$ are the same as our construction in (3.3).

Recall that the target density $\mathcal{T}$ has probability mass function given by:

$$P(M_i = \{a_1 a_2 \cdots a_p\}) = \frac{e^{\lambda s_i}}{\sum_{i \in I} e^{\lambda s_i}},$$

where $\{a_1 a_2 \cdots a_p\}$ is a model sequence. For $i = 1, 2, ..., p$, the full conditional distributions are then:

$$P(x_i = a_i | x_{1:i-1} = a_{1:i-1}, x_{i+1:p} = a_{i+1:p})$$

$$= \frac{P(x_i = a_i, x_{1:i-1} = a_{1:i-1}, x_{i+1:p} = a_{i+1:p})}{P(x_{1:i-1} = a_{1:i-1}, x_{i+1:p} = a_{i+1:p})}$$

$$= \frac{P(x_i = a_i, x_{1:i-1} = a_{1:i-1}, x_{i+1:p} = a_{i+1:p})}{P(x_i = a_i, x_{1:i-1} = a_{1:i-1}, x_{i+1:p} = a_{i+1:p}) + P(x_i = 1 - a_i, x_{1:i-1} = a_{1:i-1}, x_{i+1:p} = a_{i+1:p})}$$

$$= \frac{e^{\lambda s(M_i)}}{e^{\lambda s(M_i)} + e^{\lambda s(M_i')}}$$

which is the same as (3.3).

3.5.3 Proof of convergence of the model selection criterion (Theorem 6 - Thresholding marginal frequencies)

To complete the proof outline given in the main text, we need to establish the following two theorems. Denote the log-likelihood by $l_n(\cdot)$.
Law of Iterated Logarithm
(Petrov, 1995, p.248)

Uniform Law of Large Number
(Ferguson, 1996, p.107-108)

For fixed $\beta$, $||i_n(\beta)||$ is $O(n)$ a.s.
(Qian and Wu, 2006)

Lemma 6:
$||u_n(\beta_0)||$ is $O(\sqrt{n \log \log n})$ a.s.

+ Taylor’s expansion

Lemma 7:
$||\beta_0 - \beta||$ is $O(\sqrt{n^{-1} \log \log n})$ a.s.

Lemma 8:
$||l_n(\beta_0) - l_n(\hat{\beta})||$ is $O(\log \log n)$ a.s.

+ nested model structure

Theorem 10: $||l_n(\beta_0) - l_n(\hat{\beta}_c)||$ is $O(\log \log n)$ a.s.

Lemma 9:
For any $\beta \neq \beta_0$, $||E \left[\left(\frac{F'(X^T_0\beta_0) - F'(X^T_0\hat{\beta})}{X^T_0(\beta_0 - \beta)}\right)\right]|| > 0$

Theorem 11: $\liminf_{n \to \infty} \frac{1}{n}||l_n(\beta_0) - l_n(\hat{\beta}_w)|| > 0$ a.s.

Figure 3.2: A roadmap for proving Theorems 10 and 11
Theorem 10. For any correct model $\alpha_c$, denote the associated coefficients estimates by $\hat{\beta}_c$, then $\|l_n(\hat{\beta}_c) - l_n(\beta_0)\| = O(\log \log n)$ a.s. ($\beta_0$ is as given in Section 3.1.1, i.e. it is the coefficients of the best model.)

Theorem 11. For any incorrect model $\alpha_w$, denote the associated coefficients estimates by $\hat{\beta}_w$, we have

$$\liminf_{n \to \infty} \frac{1}{n} \|l_n(\beta_0) - l_n(\hat{\beta}_w)\| > 0 \text{ a.s.}$$

To establish Theorem 10, we require further the following lemmas. Denoting the coefficients estimates of the full model by $\hat{\beta}$,

1. $\|u_n(\beta_0)\|$ is $O(\sqrt{n \log \log n})$ a.s.
2. $||\beta_0 - \hat{\beta}||$ is $O(\sqrt{n^{-1} \log \log n})$ a.s.
3. $||l_n(\beta_0) - l_n(\hat{\beta})||$ is $O(\log \log n)$ a.s.

For readers’ convenience, a roadmap for proving the results is given in Figure 3.2.

We first present the setup, then the proof for the lemmas and finally the proof for the two theorems. Consider the usual regression setting of i.i.d. $(Y_i, X_i)$ for $i = 1, 2, ..., n$ with density $f_{Y_i, X_i}(y, x|\beta)$, where $Y_i$ is a scalar, $X_i$ and $\beta$ are both $p \times 1$ vectors. When the context is clear, we may omit the index $i$ for a cleaner presentation. We factorise the joint density $f_{Y, X}(y, x|\beta) = f_Y(y|X, \beta) \cdot f_X(x|\beta)$ and suppose $f_Y(y|x, \beta)$ follows the Generalised Linear Model (GLM), and $f_X(x|\beta) = f_X(x)$ has some fixed multivariate distribution. The inference problem is about estimating $\beta$ given the data $(y_i, x_i), i = 1, 2, ..., n$. Since $f_X(x)$ does not depend on $\beta$, it is clear to find the maximum likelihood estimate, we have

$$\hat{\beta}_{MLE} = \arg \max_{\beta} \prod_{i=1}^{n} f_{Y, X}(y_i, x_i|\beta) = \arg \max_{\beta} \prod_{i=1}^{n} f_Y(y_i|x_i, \beta).$$

Hence, the distribution of $X_i$ plays no role in the estimation, which is a relief as we can avoid dealing with the complex multivariate distributions. However, while the distribution does not take part in the estimation, it does restrict when estimation is possible and so some regularity conditions on $X_i$ are still needed. Without loss of generality, we assume $E(X) = 0$, then the (regularity) conditions required are:

1. $P(X^Tv \neq 0) > 0$ for all $v \neq 0$ in $\mathbb{R}^p$,
2. $E(XX^T)$ is (entry-wise) finite and has full-rank,

3. $\beta_0$ is identifiable.

We use the simple linear regression to illustrate the reasoning behind the conditions. Consider $y_i|x_i = x_i^T\beta + \epsilon_i$, $\epsilon_i \sim N(0,\sigma_i^2)$, where $i = 1, 2, ..., n$, $\beta$ are non-zero. Suppose we are in a situation where $x_i$ can only take value 0 (i.e. $X_i$ has a degenerate distribution at 0), then the data we ever get are $(x_i, y_i) = (0, 0)$.

It’s clear in such case, the recovery of $\beta$ is impossible: if we inspect the usual Maximum-Likelihood estimator $\hat{\beta}_{MLE} = (X^TX)^{-1}X^Ty$, we see there is a problem with the $(X^TX)^{-1}$ as the matrix is a 0 matrix and does not have an inverse. In related situations where $X_i$ has a distribution converging to the degenerate distribution (at 0), similar problem would occur. One expects at some point the estimate would stop improving since the new data points $(0, 0)$ are non-informative (which corresponds to a block of 0 in the data matrix. It may be instructive to imagine $X_2 = \begin{bmatrix} X \\ 0 \end{bmatrix}$ $\Rightarrow X_2^TX_2 = \begin{bmatrix} X & 0 \\ 0 & 0 \end{bmatrix}$). Hence, the estimate will not converge to the true parameters. In practice, this situation would occur when one has a measurement device that degrades over time. In this work, we assume condition 1 to rule out these abnormal cases.

On the other extreme, if $X_i$ “explodes”, in the sense of having an infinite variance, then the matrix $(X^TX)^{-1}$ would give $\beta = 0$ (except for the intercept as the associated covariates have zero variance, rather than infinite variance). Hence, the recovery of $\beta$ is impossible. This problematic case and all the related ones are ruled out by condition 2.

We need the last condition because even when $X$ has a well-behaved (e.g. no degeneracy or infinite variance) distribution, there still could be issue with the maximum likelihood estimation when $\beta_0$ is not identifiable. Continuing with our previous example, suppose we fixed another $\beta_1$ and construct a distribution of $X$ taking values $x$ only on the hyperplane defined by $x^T(\beta_0 - \beta_1) = 0$. Then given the observations generated by the model $y_i|x_i = x_i^T\beta + \epsilon_i$, we can’t tell whether the estimated $\hat{\beta}$ is $\beta_0$ or $\beta_1$ as both of them generate the same observations. This is also known as the problem of (multi-)collinearity.

We will use the following notation for the rest of this section.

$$U(\beta) = \frac{1}{n} \sum_{i=1}^n \frac{\partial \log f_Y(y_i|x_i, \beta)}{\partial \beta}, \quad I(\beta) = \frac{\partial^2 \log f_Y(y|x, \beta)}{\partial \beta \partial \beta^T}$$

$$u_n(\beta) = \frac{\partial \log f_Y(y_i|x_i, \beta)}{\partial \beta}, \quad i_n(\beta) = \frac{\partial^2 \log f_Y(y_i|x_i, \beta)}{\partial \beta \partial \beta^T}$$

(3.10) (3.11)
and consider the Generalised Linear Model (GLM),

\[ f_Y(y|x, \beta) = \exp \left( yx^T \beta - F(x^T \beta) + G(y) \right), \]

where \( F(\cdot) \) is the log-cumulant generating function. Hence, we have

\begin{align*}
U(\beta) &= yx - F'(x^T \beta)x, & I(\beta) &= -x^TF''(x^T \beta)x, & (3.12) \\
u_n(\beta) &= \sum_{i=1}^n y_ix_i - F'(x_i^T \beta)x_i, & i_n(\beta) &= -\sum_{i=1}^n x_i^TF''(x_i^T \beta)x_i. & (3.13)
\end{align*}

\( F'(\cdot) \) is also known as the canonical inverse link function (which is monotone and satisfies that \( E(Y|X) = F'(X^T \beta_0) \)), and we assume this to be continuously differentiable. Next, for a fixed \( \beta \), by the Law of Large Number we have

\[ \frac{u_n(\beta)}{n} \overset{a.s.}{\to} E[U(\beta)] \quad \text{and} \quad \frac{i_n(\beta)}{n} \overset{a.s.}{\to} E[I(\beta)] \quad (3.14) \]

Note that \( u_n(\beta) \) is a vector, and \( i_n(\beta) \) is a matrix; the convergence referred to should be understood as element-wise. For most practical purposes, we can assume \( E[I(\beta)] \) exists and is of full-rank, then the above convergence implies (for a fixed \( \beta \)) there exists \( K_1, K_2 > 0 \) s.t.

\[ K_1n \leq \lambda_1(i_n(\beta)) \leq \ldots \leq \lambda_p(i_n(\beta)) \leq K_2n \quad a.s., \quad (3.15) \]

where \( \lambda_k(A) \) is the \( k \)-th eigenvalue (in increasing order of magnitude) of a symmetric matrix \( A \). Subsequently, this implies

\[ K_1n \leq ||i_n(\beta)|| \leq K_2n \quad a.s.. \quad (3.16) \]

Similar construction has been used in Qian and Wu (2006) to prove strong limit results for binomial-family GLM.

**Proof of Theorem 10.** With the above setup, recall that \( \beta_0 \) is the regression coefficients of the best model, and \( \hat{\beta} \) is the coefficient estimates of the full model (i.e. the model that includes all the predictors), we are ready to show the following lemmas:

1. \( ||u_n(\beta_0)|| \) is \( O(\sqrt{n \log \log n}) \) a.s.
2. \( ||\beta_0 - \hat{\beta}|| \) is \( O(\sqrt{n^{-1} \log \log n}) \) a.s.
3. \( ||l_n(\beta_0) - l_n(\hat{\beta})|| \) is \( O(\log \log n) \) a.s.
For notational tidiness, the ‘a.s.’ suffix may be dropped in the paragraphs unless it is the main focus. All $O(.)$ results should be understood as they hold almost surely.

**Lemma 6.** $||u_n(\beta_0)||$ is $O(\sqrt{n \log \log n})$ a.s.

**Proof.** To begin with, we analyse the vector $u_n(\beta_0)$ term-by-term. Denote the $j$-th term of $u_n(\beta_0)$ by $u_n(\beta_0)_j$ and let

$$\zeta_i^{(j)} = y_i x_{ij} - F'(x_i^T \beta_0)x_{ij}, \quad i = 1, 2, \ldots, n, \quad j = 1, 2, \ldots, p$$

then we have $u_n(\beta_0)_j = \sum_{i=1}^n \zeta_i^{(j)}$, where the mean and variance of $\zeta_i^{(j)}$ are given by:

$$E\zeta_i^{(j)} = E\left[ (y_i - F'(x_i^T \beta_0)) x_{ij} \right] = 0$$

$$Var\zeta_i^{(j)} = Var(\zeta_i^{(j)}|x_i) + Var(E(\zeta_i^{(j)}|x_i))$$

$$= E\left( Var(y_i|x_i)x_{ij}^2 \right) + 0 < \infty$$

(The conditional variance is finite by the existence of mgf as mentioned in 1.3.3, and then $E(x_{ij}^2)$ is finite by regularity condition 2.)

Now, we can apply Theorem 8 to get

$$u_n(\beta_0)_j = \sum_{i=1}^n \zeta_i^{(j)} \text{ is } O(\sqrt{n \log \log n}) \text{ a.s.}$$

It then follows that $||u_n(\beta)|| = \sqrt{\sum_{j=1}^p u_n(\beta)_j^2}$ is also $O(\sqrt{n \log \log n})$, and the lemma is established.

\[\square\]

**Lemma 7.** $||\beta_0 - \hat{\beta}||$ is $O(\sqrt{n^{-1} \log \log n})$ a.s.

**Proof.** Applying Taylor’s expansion, for some $\zeta$ between $\beta_0$ and $\hat{\beta}$ (entry-wise),

$$u_n(\beta_0) = u_n(\hat{\beta}) + i_n(\zeta)(\beta_0 - \hat{\beta}) \quad (3.17)$$

$$\Rightarrow \quad i_n(\zeta)^{-1}u_n(\beta_0) = \beta_0 - \hat{\beta} \quad (3.18)$$

$$\Rightarrow \quad ||\beta_0 - \hat{\beta}|| \leq ||i_n(\zeta)^{-1}|| \cdot ||u_n(\beta_0)|| \quad (3.19)$$

By Lemma 6, $||u_n(\beta_0)||$ is $O(\sqrt{n \log \log n})$, if we can show $||i_n(\zeta)^{-1}||$ is $O(n^{-1})$, then combining them gives the required $O(\sqrt{n^{-1} \log \log n})$. To show $||i_n(\zeta)^{-1}||$ is
\( O(n^{-1}) \), let \( i(\beta_0) = E[I(\beta_0)] \), we need to strengthen (3.14) to
\[
\frac{1}{n} i_n(\hat{\beta}) \rightarrow i(\beta_0) \quad a.s. \tag{3.20}
\]
to give a version of results (3.15) and (3.16) corresponding to \( \hat{\beta} \). It then follows from \( \zeta \) is between \( \beta_0 \) and \( \hat{\beta} \) that \( ||i_n(\zeta)^{-1}|| \) is \( O(n^{-1}) \).

It is standard (e.g. see Ferguson (1996, p.108)) to establish (3.20) by showing that, for some \( \rho > 0 \),
\[
\sup_{\beta \in B[\beta_0, \rho]} \left| \frac{1}{n} i_n(\beta) - i(\beta_0) \right| \rightarrow 0, \tag{3.21}
\]
where \( B[\beta_0, \rho] \) is a closed ball centered at \( \beta_0 \) with some radius \( \rho > 0 \), since this implies
\[
\left| \frac{1}{n} i_n(\hat{\beta}) - i(\beta_0) \right| \leq \left| \frac{1}{n} i_n(\hat{\beta}) - i(\hat{\beta}) \right| + \left| i(\hat{\beta}) - i(\beta_0) \right|
\leq \sup_{\beta \in B[\beta_0, \rho]} \left| \frac{1}{n} i_n(\beta) - i(\beta) \right| + \left| i(\hat{\beta}) - i(\beta_0) \right| \quad a.s.
\rightarrow 0 \quad a.s.
\]
where the second term \( \left| i(\hat{\beta}) - i(\beta_0) \right| \) goes to zero almost surely by \( \hat{\beta} \rightarrow \beta_0 \), continuity of \( i(\cdot) \) and the continuous mapping theorem.

To show (3.21), we simply need to check the conditions of Theorem 7. Let
\( V(x, \beta) = -x^T F''(x^T \beta) x \), which is \( I(\beta) \) in (3.12).

1. Since \( F'(\cdot) \) is by assumption continuously differentiable, so \( F''(x^T \beta) \) is continuous in \( \beta \). Hence, \( V(x, \beta) \) is continuous in \( \beta \) for all \( x \).

2. For the \((j,k)\)-entry of \( V(x, \beta) \), we can restrict \( \beta \) to the compact set \( B[\beta_0, \rho_{jk}] \) to get uniform continuity. Next, taking intersection of all the balls, we have that all entries of \( V(x, \beta) \) are uniformly continuous on \( B[\beta_0, \rho_m] \), where \( \rho_m = \min_{j,k} \rho_{jk} \). This together with our assumption that \( E[I(\beta)] \) exists imply that there is an integrable function \( K(x) \) such that \( |V(x, \beta)| \leq K(x) \) (entry-wise) for all \( \beta \in \Theta_{\rho_m} \).

3. Let \( \Theta = B[\beta_0, \rho_m] \), as given in the last point, it is compact by construction.

This completes the proof of lemma 7. \( \square \)

**Lemma 8.** \( ||l_n(\beta_0) - l_n(\hat{\beta})|| \) is \( O(\log \log n) \).
Proof. Applying Taylor’s expansion gives that for \( \zeta \) between \( \beta_0 \) and \( \hat{\beta} \),

\[
\ln(\beta_0) = \ln(\hat{\beta}) + u_n(\hat{\beta})(\beta_0 - \hat{\beta}) + \frac{1}{2}(\beta_0 - \hat{\beta})i_n(\zeta)(\beta_0 - \hat{\beta})^T
\]

\[
\Rightarrow \quad \ln(\beta_0) - \ln(\hat{\beta}) = \frac{1}{2}(\beta_0 - \hat{\beta})i_n(\zeta)(\beta_0 - \hat{\beta})^T
\]

\[
\Rightarrow \quad ||\ln(\beta_0) - \ln(\hat{\beta})|| \leq \frac{1}{2}||\beta_0 - \hat{\beta}|| \cdot ||i_n(\zeta)|| \cdot ||(\beta_0 - \hat{\beta})^T||
\]

On the right hand side, both \( ||\beta_0 - \hat{\beta}|| \) and its transpose is \( O(\sqrt{n^{-1}\log \log n}) \) by Lemma 7, and again by (3.20) \( ||i_n(\zeta)|| \) is \( O(n) \). Overall, this gives \( O(\log \log n) \).

Now we are ready to proceed with the proof of Theorem 10:

Proof. First, if the full model satisfies the theorem, then all other correct models also satisfy the theorem. This relies on the nested structure of models:

\[
\alpha_0 \subset \alpha_c \subset \alpha_{full} \quad \Rightarrow \quad l_n(\beta_0) \leq l_n(\hat{\beta}_c) \leq l_n(\hat{\beta}) \quad (3.22)
\]

\[
\Rightarrow \quad 0 \leq l_n(\hat{\beta}_c) - l_n(\beta_0) \leq l_n(\hat{\beta}) - l_n(\beta_0) \quad (3.23)
\]

\[
\Rightarrow \quad ||l_n(\hat{\beta}_c) - l_n(\beta_0)|| \leq ||l_n(\beta) - l_n(\beta_0)|| \quad (3.24)
\]

which suggests the full model, among all the correct models, is the furthest away from the true model in terms of likelihood, hence it suffices to show the theorem is true for the full model. But we have shown this in Lemma 8, so we are done.

Now we give the proof of Theorem 11.

Proof. For a fixed \( \beta \),

\[
\frac{1}{n}(l_n(\beta_0) - l_n(\beta)) = \frac{1}{n}\left\{ \sum_{i=1}^{n} y_i x_i^T(\beta_0 - \beta) - [F(x_i^T \beta_0) - F(x_i^T \beta)] \right\}
\]

\[
\Rightarrow \quad \text{a.s.} \quad E(Y_iX_i^T(\beta_0 - \beta) - [F(X_i^T \beta_0) - F(X_i^T \beta)])
\]

Next, applying the Taylor’s expansion, for \( \zeta \) s.t. \( x_i^T \zeta \) is (strictly) between \( x_i^T \beta_0 \) and \( x_i^T \beta \), we have

\[
F(x_i^T \beta) = F(x_i^T \beta_0) + F'(x_i^T \zeta)x_i^T(\beta - \beta_0)
\]

\[
\Rightarrow \quad F(x_i^T \beta) - F(x_i^T \beta_0) = F'(x_i^T \zeta)x_i^T(\beta - \beta_0)
\]

80
Substituting this into above, we have

\[ E (Y_i X_i^T (\beta_0 - \beta) + F'(X_i^T \zeta) X_i^T (\beta - \beta_0)) \]
\[ = E (E (Y_i (\beta_0 - \beta) + F'(X_i^T \zeta) X_i^T (\beta - \beta_0) | X_i)) \]
\[ = E (E(Y_i | X_i) X_i^T (\beta_0 - \beta) - F'(X_i^T \zeta) X_i^T (\beta_0 - \beta)) \]
\[ = E (F'(X_i^T \beta_0) - F'(X_i^T \zeta)) X_i^T (\beta_0 - \beta) \]

Note that \( \zeta \) is merely a device to help us express the power series of \( F(\cdot) \) in terms of its first derivative. While it depends on \( X_i \) and \( \beta \), it does not add complications to our derivation (above and below). Formally, it should be read as \( \zeta = \zeta(X_i, \beta) \).

Now we will need the following lemma.

**Lemma 9.** For any \( \beta \neq \beta_0 \),

\[ \| E (\left[ F'(X_i^T \beta_0) - F'(X_i^T \zeta) \right] X_i^T (\beta_0 - \beta)) \| > 0 \quad (3.25) \]

*Proof.\* By the regularity conditions and the setting \( \beta \neq \beta_0 \), we have that \( X_i^T (\beta_0 - \beta) \) cannot be zeroes for all \( i = 1, 2, \ldots, n \). When they are not zeroes, we now show

\[ [F'(X_i^T \beta_0) - F'(X_i^T \zeta)] X_i^T (\beta_0 - \beta) \]

are either all positive or all negative.

First, since for any fixed \( x_i \), \( \zeta \) is such that \( x_i^T \zeta \) is (strictly) between \( x_i^T \beta_0 \) and \( x_i^T \beta \), so \( x_i^T (\beta_0 - \beta) \) has the same sign as \( x_i^T (\beta_0 - \zeta) \). Next, \( F'(\cdot) \) is monotone by assumption, so if \( F'(\cdot) \) is increasing (decreasing), then \( [F'(X_i^T \beta_0) - F'(X_i^T \zeta)] \) has the same (opposite) sign as \( x_i^T (\beta_0 - \beta) \) for all \( i = 1, 2, \ldots, n \). Multiplying them together always gives a positive (negative) sign, and hence

\[ \| E (\left[ F'(X_i^T \beta_0) - F'(X_i^T \zeta) \right] X_i^T (\beta_0 - \beta)) \| > 0, \]

and the proof of the lemma is complete. \( \square \)

Summarising all the progress so far, we have shown that for \( \beta \neq \beta_0 \), we have

\[ \liminf_{n \to \infty} \frac{1}{n} \| l_n(\beta_0) - l_n(\beta) \| > 0 \quad a.s. \]
Since a wrong model $\alpha_w$ by definition must miss at least one true predictor, so the associated coefficients $\beta_w$ satisfies $\beta_w \neq \beta_0$, and we have

$$\liminf_{n \to \infty} \frac{1}{n} ||l_n(\beta_0) - l_n(\beta_w)|| > 0 \ \text{a.s..} \tag{3.26}$$

Finally, we need to strengthen this result from $\beta_w$ to $\hat{\beta}_w$. First, observe that if we restrict $\beta$ to those in the class of wrong model $\mathcal{M}_w$, then (3.26) can be tightened to that for some $K > 0$,

$$\liminf_{n \to \infty} \frac{1}{n} ||l_n(\beta_0) - l_n(\beta_w)|| > K \ \text{a.s..} \tag{3.27}$$

This is done by associating to each $\beta_w$, corresponding to $\alpha_w \in \mathcal{M}_w$, the constant

$$K_w = \liminf_{n \to \infty} \frac{1}{n} ||l_n(\beta_0) - l_n(\beta_w)||$$

(i.e. the LHS of (3.26)) and letting $K = \inf_{\alpha_w \in \mathcal{M}_w} K_w$ (note that there are only finite number of wrong models).

Next, we take an open ball of radius $\rho > 0$ around $\beta_w$, $B(\beta_w, \rho)$. Since $\hat{\beta}_w \Rightarrow \beta_w$, it will enter the ball almost surely, and if we can show that (3.27) holds uniformly for $B(\beta_w, \rho)$, i.e. for some $K_2 > 0$,

$$\inf_{\beta \in B(\beta_w, \rho)} \liminf_{n \to \infty} \frac{1}{n} ||l_n(\beta_0) - l_n(\beta)|| > K_2 \ \text{a.s.}$$

then $\beta_w$ may be replaced by $\hat{\beta}_w$ in (3.26).

Let the radius of the open ball around $\beta_w$ be s.t. $0 < \rho_n < \frac{1}{n} ||\beta_0 - \beta_w||$, then for $n \geq 2$, we have $\beta \neq \beta_0$, $\forall \beta \in B(\beta_w, \rho_n)$ (because every point in the ball is at least $\frac{1}{2} ||\beta_0 - \beta_w||$ away from $\beta_0$). Next, write $f(\beta) = ||E \left[ \left[ F'(X_i^T \beta_0) - F'(X_i^T \zeta) \right] X_i^T (\beta_0 - \beta) \right] ||$. Note that as $n$ goes to $\infty$ (i.e. $\rho_n \to 0$), we have

$$\inf_{\beta \in B(\beta_w, \rho_n)} f(\beta) \nearrow f(\beta_w).$$

Since we have shown in 3.27 that $f(\beta_w) > K$ for some $K > 0$, we know for $0 < \epsilon < K$, there exists some $N$ such that if $n > N$, then

$$\left| \inf_{\beta \in B(\beta_w, \rho_n)} f(\beta) - f(\beta_w) \right| < \epsilon \Rightarrow \left| \inf_{\beta \in B(\beta_w, \rho_n)} f(\beta) \right| > K - \epsilon > 0. \tag{3.28}$$

Take $\rho = \rho_{N+1}$, then the proof is complete.

□
3.5.4 Proof of convergence of the parallelisation extension

Consider the setting given in Section 3.2.6, i.e. partition the predictors into \( k \) groups, apply Gibbs stochastic search to each block and combine the results from all the blocks.

**Theorem 12.** For a group of predictors, let \( p_c \) be the probability that a model sampled from the stationary distribution is a correct sub-model, and \( p_w = 1 - p_c \) be the probability that the sampled model is a wrong sub-model. Then, \( p_c/p_w \) is unbounded as the number of datapoints \( n \) goes to infinity.

**Proof.** The proof uses the same idea as for the non-parallel case except that we have to use a more refined models hierarchy to handle the partitioning.

Suppose block \( i \) has \( q_i \) predictors, of which \( q_i \) of them are from \( B \), the set of predictors of the best model, first we will prove the following lemma:

**Lemma 10.** Given a fixed block, for any \( \alpha_{cs} \in M_{cs} \), we define the set of wrong sub-models associated with \( \alpha_{cs} \) to be the wrong sub-models that can be obtained by changing any number of the \( q_i \) predictors of \( \alpha_{cs} \), and we denote this set by \( M_{\alpha_{cs} ws} \). Then for a fixed \( \alpha_{cs} \in M_{cs} \) and any \( \alpha_{ws} \in M_{\alpha_{cs} ws} \), we have

\[
\liminf_{n \to \infty} \frac{1}{n} \left( l_n(\hat{\beta}_{cs}) - l_n(\hat{\beta}_{ws}) \right) > 0 \quad a.s.
\]

Here is an example of wrong sub-models associated with a correct sub-model. Suppose the best model is 1100 1010 00, and the block size is 4. Consider a correct sub-model in block 1, \( \alpha_{cs} = 1110 0000 00 \), the associated set of wrong sub-models are \( M_{\alpha_{cs} ws} = \{0010 0000 00, 1010 0000 00, 0110 0000 00\} \).

**Proof.** By the strong consistency of maximum likelihood estimator, we have

\[
\frac{1}{n} \left( l_n(\hat{\beta}_{cs}) - l_n(\hat{\beta}_{ws}) \right) \overset{a.s.}{\to} l(\beta_{cs}) - l(\beta_{ws}).
\]

This is positive since the wrong sub-model by construction is nested in the correct sub-model, i.e. \( \alpha_{ws} \subset \alpha_{cs} \), and it must differ from the correct sub-model by at least one of the true predictors.

This lemma suggests the log-likelihood score of any correct sub-model is greater than the log-likelihood of any of the associated wrong sub-model by an order of \( n \). We will show that similar result holds after adjusting for the number of models in each class. Formally, we show that the ratio of the probability of choosing a correct sub-model, \( p_c^{(i)} \), to the probability of choosing a wrong sub-model, \( p_w^{(i)} \), is
unbounded as \( n \) goes to infinity. Hence, if we sample from the model subspace according to the log-likelihood, we expect to get many more correct sub-models than wrong sub-models. Then combining them would recover a correct model.

Now we return to the main proof. Recall that in each block, there are in total \( 2^{p_i} \) models, which consist of \( 2^{p_i-q_i} \) correct sub-models and \( 2^{p_i} - 2^{p_i-q_i} \) wrong sub-models. For each correct sub-model, there are \((2^{q_i} - 1)\) ways to modify it to produce a wrong sub-model; we can group the wrong sub-models according to the correct sub-models. Denoting \( \hat{s}_{\alpha ws} := \arg \max_{\gamma \in \mathcal{M}_{ws}} \hat{s}_{\gamma} \), there exists \( K_\alpha > 0, \alpha \in \mathcal{M}_{cs} \) such that

\[
\frac{p_c}{p_w} = \frac{\sum_{\alpha \in \mathcal{M}_{cs}} \hat{s}_\alpha}{\sum_{\alpha \in \mathcal{M}_{ws}} \hat{s}_\alpha} \geq \frac{\sum_{\alpha \in \mathcal{M}_{cs}, \gamma \in \mathcal{M}_{ws}} \hat{s}_\gamma}{\sum_{\alpha \in \mathcal{M}_{ws}} \hat{s}_\alpha} = \frac{\sum_{\alpha \in \mathcal{M}_{cs}} \hat{s}_\alpha}{(2^{q_i} - 1) \sum_{\alpha \in \mathcal{M}_{cs}} \hat{s}_{\alpha ws}} \geq \frac{\sum_{\alpha \in \mathcal{M}_{cs}} e^{\lambda_{s ws}}}{(2^{q_i} - 1) \sum_{\alpha \in \mathcal{M}_{cs}} e^{\lambda_{s ws}}} \geq \frac{\sum_{\alpha \in \mathcal{M}_{cs}} e^{\lambda_{s ws} + K_\alpha n}}{(2^{q_i} - 1) \sum_{\alpha \in \mathcal{M}_{cs}} e^{\lambda_{s ws}}} \geq \frac{\sum_{\alpha \in \mathcal{M}_{cs}} e^{\lambda_{s ws} + K_\alpha}}{(2^{q_i} - 1) \sum_{\alpha \in \mathcal{M}_{cs}} e^{\lambda_{s ws}}} \geq \frac{\sum_{\alpha \in \mathcal{M}_{cs}} e^{\lambda_{s ws} + K_\alpha n}}{(2^{q_i} - 1) \sum_{\alpha \in \mathcal{M}_{cs}} e^{\lambda_{s ws}}} \geq \frac{\sum_{\alpha \in \mathcal{M}_{cs}} e^{\lambda_{s ws} + K_\alpha}}{(2^{q_i} - 1) \sum_{\alpha \in \mathcal{M}_{cs}} e^{\lambda_{s ws}}} \geq \frac{\sum_{\alpha \in \mathcal{M}_{cs}} e^{\lambda_{s ws} + K_\alpha}}{(2^{q_i} - 1) \sum_{\alpha \in \mathcal{M}_{cs}} e^{\lambda_{s ws}}} \geq \frac{\sum_{\alpha \in \mathcal{M}_{cs}} e^{\lambda_{s ws} + K_\alpha n}}{(2^{q_i} - 1) \sum_{\alpha \in \mathcal{M}_{cs}} e^{\lambda_{s ws}}},
\]

which is unbounded (as \( n \to \infty \)), and the proof is completed. \( \square \)
Chapter 4

Structural Generalised Linear Model

Chapter Summary

In this chapter, we propose a flexible class of models for multivariate data, constructed using a Directed Acyclic Graphs (DAG) and the Generalised Linear Model (GLM), and show the statistical consistency in recovering the best approximating model from data under different constraints. Since the model is constructed using a graph, to estimate the model, we need to first estimate the graph. This requires us to search through the graph space, which has cardinality greater than $2^{(2^{p})}$ where $p$ is the number of variables in the data, and it can be very challenging. The main contribution of this work is to carefully arrange a hierarchical structure out of the model space so that the Gibbs Stochastic Search algorithm from the last chapter can be applied, making the recovery of the target model possible. Other applications like imputation, simulation of realistic data and structural inference are also investigated.

4.1 Introduction

Structure learning is a class of techniques for discovering systemic relationships among a set of variables. It helps scientists understand the mechanisms behind phenomena and has applications in many areas including climatology, system biology and genomics. From a statistical point of view, the systemic relationship is expressed by the joint distribution of the variables, and there are in general two approaches to learn the joint distribution from data. The first is the global approach, where a single statistical model is specified for the variables all at once. Two popular classes of models are the copula models and the graphical models. The former is based on the Sklar’s theorem, and it models a joint distribution
using a copula function and marginal distributions. Some examples in this class are the Gaussian copula, the Frank copula and the Farlie–Gumbel–Morgenstern copula. The latter class is based on the Hammersley-Clifford theorem, and it models a joint distribution using a graph. Common examples in this class are the Gaussian model (Yuan and Lin, 2007), the Ising model (Ravikumar et al., 2010) and Multinomial model (Geiger and Meek, 1998).

The global approach is often used when there is a natural candidate for the joint distribution suggested by the problem at hand. In other cases, it can be hard to specify a global model and justify such choice. This motivates the alternative - the local approach. The local approach constructs a model by specifying and stitching together local relationships about the variables. Two notable instances are the Vine-copula models (Joe, 1996; Bedford and Cooke, 2001; Aas et al., 2009; Czado, 2010), which use pairwise copulas to build up the full copula (and subsequently the joint distribution), and the GLM-based graphical models (Yang et al., 2012, 2013, 2014, 2015), which specify a conditional distribution for each node on a graph to build up the joint distribution. The local approach has greater flexibility and covers a wider class of distributions than the global approach, but it comes at a cost that the estimation is harder to perform. This is because the number of possible ways to construct a global model from local relationships grows exponentially with the number of variables, and it can be difficult to find the best way out of the combinatorially large number of possibilities.

In this work, we take the local graphical approach and propose a flexible multivariate model specified via a Directed Acyclic Graph (DAG) and the Generalised Linear Models (GLM). The DAG captures the dependencies between variables and the GLM specifies the distributional assumptions underlying the dependencies. We aim to recover both the structure (i.e. the graph) and the coefficients estimate (of the GLMs) given the data. This is challenging because given \( p \) variables / nodes, the number of possible graphs is at least \( 2^{\binom{p}{2}} \), and it is infeasible to enumerate all the graphs even for a moderate size of \( p \). To tackle this, we recast structure learning into a form similar to the best subset model selection, and we establish a hierarchical structure out of the model space so that the combinatorial complexity can be handled using the Gibbs Stochastic Search introduced in the last chapter. Applications like knowledge discovery, imputations and simulation are also presented. We present the model formulation and theoretical results in Section 4.2 and 4.3, followed by numerical studies, applications and conclusion in Section 4.4, 4.5 and 4.6.
4.2 Structural Generalised Linear Model

In this section, we formulate the Structural Generalised Linear Model (SGLM). The classical Generalised Linear Model (GLM) is first introduced, then generalised to the structural counterpart. This arrangement highlights how the classical asymptotic results would naturally carry forward to the structural case.

4.2.1 Generalised Linear Model (GLM)

Consider i.i.d. random variables \((Y_i, X_i), i = 1, 2, ..., n\), where each \(Y_i\) follows an exponential family of distributions with density function

\[
f(y_i; \theta_i, \phi) = \exp(\alpha(\phi)[y_i \theta_i - b(\theta_i) + h(y_i)] + \beta(\phi, y_i)),
\]

and it is related to \(X_i\) through the link function \(g: \mathbb{R} \to \mathbb{R}, g(E(Y_i|X_i)) = X_i \beta\), where \(\beta\) are the parameters of the model. For notational simplicity, let \(\mu_i = E(Y_i|X_i)\) and \(\eta_i = X_i \beta\) so that we have \(g(\mu_i) = \eta_i\).

The parameters are estimated using Maximum Likelihood Estimation (MLE). The consistency and asymptotic normality of the estimator were established by Fahrmeir and Kaufmann (1985), and the law of iterated logarithm was given by Xiao and Liu (2008). The result in Xiao and Liu (2008) assumed bounded regressors, we relaxed that assumption in the previous chapter, but only proved the case for univariate response variable. Based on that, we also developed new results in the context of model selection. In particular, the theorem of interest to this chapter is:

**Theorem 13.** Consider the setting of generalised linear model with a set of \(p\) predictors, denoted by \(P\), and suppose the true data generating process (DGP) uses \(q < p\) of them, denoted by \(Q\). Let \(l_n(S)\) be the maximised likelihood of the model associated with the subset of predictors \(S\), then for any subset \(C : Q \subset C \subset P\), we have

\[
|l_n(C) - l_n(Q)| \text{ is } O(\log \log n) \quad \text{a.s.} \quad (4.1)
\]

where \(n\) is the number of data. And for any subset \(W \subset Q\), we have

\[
\liminf_{n \to \infty} \frac{1}{n} |l_n(Q) - l_n(W)| > 0 \quad \text{a.s.} \quad (4.2)
\]

This theorem quantifies the model space hierarchy by comparing the models in terms of their asymptotic properties.
4.2.2 Structural Generalised Linear Model (SGLM)

A graphical model is a statistical model specified by a graph and distributional assumptions. A graph consists of nodes and edges: nodes represent variables, and an edge between two nodes represents dependence between two variables. We will only work with a special type of graph in this work, the Directed Acyclic Graph (DAG), which is a graph with edges that are directed and satisfy that none of them form a cycle. An example of a graph is shown in Figure 4.1.

A graph specifies the dependence between the variables and gives us a way to simplify a joint distribution. Take Figure 4.1 as an example, starting with the joint distribution \( f(x_1, x_2, x_3, x_4, x_5) \), we factorise it into product of conditional distributions:

\[
\begin{align*}
    f_5(x_5|x_1, x_2, x_3, x_4) & \cdot f_4(x_4|x_1, x_2, x_3) \\
    & \cdot f_3(x_3|x_1, x_2) \\
    & \cdot f_2(x_2|x_1) \\
    & \cdot f_1(x_1),
\end{align*}
\]

then it can be simplified according to the graph to yield

\[
\begin{align*}
    f_5(x_5|x_1, x_3, x_4) & \cdot f_4(x_4|x_1, x_2) \\
    & \cdot f_3(x_3|x_1, x_2) \\
    & \cdot f_2(x_2|x_1) \\
    & \cdot f_1(x_1).
\end{align*}
\]

To complete the specification of a graphical model, we need to state the distributional assumption for each component of the (simplified) factorisation. Consider the setup of i.i.d. random variables \( \mathbf{X}_i = (X_{i1}, X_{i2}, \ldots, X_{ip}), i = 1, 2, \ldots, n \), we use GLM to model the relationship between the variables. Specifically, denoting \( \{X_{i1}, X_{i2}, \ldots, X_{ip}\}\setminus\{X_{ik}\} \) by \( \mathbf{X}_{(i,-k)} \), the conditional distribution \( X_{ik}|\mathbf{X}_{(i,-k)} \) follows the exponential family of distributions \( f_j \) with mean given by

\[
g_k(E(X_{ik}|\mathbf{X}_{(i,-k)})) = \mathbf{X}_{(i,-k)} \beta_k, \quad i = 1, 2, \ldots, n, \quad k = 1, 2, \ldots, p,
\]
where \( \{ \beta_k, k = 1, 2, ..., p \} \) are the parameters, and \( g_k \) are the link functions capturing the relationship between the (conditional) means and the covariates. For notational simplicity, we let \( \mu_{ik} = E(X_{ik}|X_{(i,-k)}) \) and \( \eta_{ik} = X_{(i,-k)}\beta_k \), and we have \( g_k(\mu_{ik}) = \eta_{ik} \).

We refer to the model specified via a graph with dependence modelled by GLM as the Structural Generalised Linear Model (SGLM). Here is a summary:

- Data: \( n \) data points, \( p \) measurements, \((X_{i1}, X_{i2}, ..., X_{ip})\), \( i = 1, 2, ..., n \).

- Model: SGLM = Directed Acyclic Graph (DAG) + GLM distributional assumptions.
  1. start with a general multivariate density \( f(x_1, x_2, ..., x_p) \);
  2. factorise it into \( p \) components
     \[
     f(x_1, x_2, ..., x_p) = f_1(x_1|x_2, ..., x_p) f_2(x_2|x_3, ..., x_p) \cdots f_{p-1}(x_{p-1}|x_p) f_p(x_p);
     \]
  3. simplify it with a graph, which has \( p \) nodes and at most \( \binom{p}{2} \) edges; edges are directed and do not form any cycles;
  4. model the components by parametric models \( f_1, f_2, ..., f_p \) following the GLM
     \[
     f_k(x_k|x_{(-k)}) \sim \text{ExpFamily} (g_k(\mu_k) = \eta_k).
     \]

We remark that the multivariate distribution proposed above is equivalent in distribution to the clique-2 mixed graphical models in Yang et al. (2014, 2015) and it may also be considered as a special case of Fahrmeir and Tutz (1994). Given that the idea of using GLMs as conditionals to build joint distributions is not new, some clarifications about our contribution are perhaps needed. Yang et al. (2014, 2015) takes the traditional Markov Random Field approach (i.e. it is based on undirected graph) and their construction is at the root founded on Hammersley-Clifford theorem; the theorem is needed to settle the graph identification issue. In contrast, our formulation uses Directed Acyclic Graph (DAG) and the result relies solely on classical (asymptotic) statistics. We take the viewpoint that structure learning is a regression model with special structure; the graph is used merely as a device to construct the multivariate distribution, and the identification is studied from the distributional point of view. The distinction is important since our method, for one, gives an entirely different structure learning algorithm, and more importantly, allows for generalisation to other regression models, essentially giving a framework to build generative models using discriminative models. This is not
possible via Yang et al. (2014, 2015)’s formulation. On the other hand, Fahrmeir and Tutz (1994) proposed a general framework of using GLMs as conditionals to build any kind of joint distribution, e.g. time series models, panel data models and survival models. The focus of their work is, however, different to ours as the structure of their models is generally assumed to be known a priori. What distinguishes this work from theirs is that our work centers around the structure learning component, and the main contribution is to establish the consistency of the structure recovery.

4.3 Theoretical results

We discuss the theoretical result in terms of three questions.

1. Given the data and the true graphical model, can we recover all the parameter values?

2. Given the data, can we recover the graph structure assuming the underlying model is in our class of models?

3. Given the data, if the underlying model is outside of our class of models, what does the fitted model mean?

The answer to the first question is positive. This follows directly from the classical result as we will see in Section 4.3.1. Skipping ahead, the third question is the classical problem of model misspecification, and it has been answered in Akaike (1973): when the data generating process (DGP) is outside of the specified class of models, the fitted model is the model (within the class) that is closest to the true DGP in the sense of Kullback-Leibler divergence. For completeness sake, a concise derivation is given in Section 4.3.3. The difficult question above is the second one, which involves inferring the graph structure from data. The challenge comes from that the number of possible graphs grows exponentially fast with the number of nodes $p$ – in fact, it grows at least as fast as $2^{\binom{p}{2}}$ – so the space of model is very large even for a moderate size of $p$, and it is hard to identify the (single) best fitted model in such a space. To tackle this, we use the idea from the last chapter which incorporates the hierarchical structure of the model space into a stochastic search algorithm; this allows us to find a consistent model in an exponentially large space. Some modifications need to be made as the problem setting in this chapter is different; in particular, a new hierarchical structure needs to be constructed, and we will do this in Section 4.3.2.
4.3.1 Parameter consistency

Given the data and the underlying graphical model, it is straightforward to derive parameter consistency as this comes directly from the classical result. The key observation is that maximising the likelihood of the joint density is equivalent to maximising the likelihood of each component of the factorised density. Mathematically, we start with

\[ f(x_1, x_2, ..., x_p) = f_1(x_1|x_2, ..., x_p) \cdot f_2(x_2|x_3, ..., x_p) \cdots f_{p-1}(x_{p-1}|x_p) \cdot f_p(x_p). \]

Note that \( f_k(x_k|x_{k+1}, ..., x_p) \) has \((p - k + 1)\) parameters (accounted for the intercept), and we label them by \( \{\alpha_k, \beta_{kj}, j = (k + 1), ..., p\} \) or in vector notation by \( \beta_k \). Next, denote the log-likelihood function corresponding to \( f_k \) by \( l_k \) and let \( \theta = (\beta_1, \beta_2, ..., \beta_p) \), we have

\[ l(\theta) = l_1(\beta_1) + l_2(\beta_2) + \cdots + l_{p-1}(\beta_{p-1}) + l_p(\beta_p). \]

This gives us

\[ \arg \max_{\theta \in \mathbb{R}^{p\times p}} l(\theta) = \bigcup_{k=1,2,...,p} \arg \max_{\beta_k \in \mathbb{R}^{p-k+1}} l_k(\beta_k) \]

showing that the likelihood maximisation can be done component-wise. This implies that given the correct model specification, the graphical model is consistent because each component is consistent by the classical result.

4.3.2 Structural consistency

Given the data, if the underlying model is within the GGLM class, but the graph structure is unknown to us, could we recover it? Noting that the number of possible directed acyclic graphs (DAG) is given by the recursive formula

\[ a_p = \sum_{k=1}^{p} (-1)^k \binom{p}{k} 2^{k(n-k)} a_{p-k}, \quad a_0 = a_1 = 1, \]

and it satisfies \( 2^{\binom{p}{2}} \leq a_p \leq 3^{\binom{p}{2}} \), we see that our problem is of the combinatorial-optimisation type which in general is hard to solve (in fact many of them are NP-hard). However, as shown in the last chapter, if the model space has structure that can be exploited, then the estimation of graph from data could become feasible. We first present an algorithm, then show that it produces a consistent estimate of the underlying graph structure.
Algorithm

First, we encode the graph by an adjacency matrix $A$. Denote the $(i, j)$-th entry of $A$ by $a_{i,j}$, $i = 1, 2, ..., p, j = 1, 2, ..., p$, then $a_{i,j} = 0$ represents there is no relationship between node $i$ and node $j$, $a_{i,j} = 1$ represents an arrow pointing from $i$ to $j$, and $a_{i,j} = 2$ represents an arrow pointing from $j$ to $i$. Note that it suffices to describe each graph using only the class of upper triangular matrices.

For ease of presentation, we vectorise the matrix by joining all the rows of the upper triangular part of the matrix. For example,

$$
vec \begin{pmatrix}
0 & 1 & 2 \\
0 & 0 & 3 \\
0 & 0 & 0
\end{pmatrix} = (1, 2, 3)
$$

The length of the vector is $s = p \cdot (p - 1)/2$. We denote the inverse operation that takes the vector back to the upper triangular adjacency matrix by $vec^{-1}$; it satisfies that $vec \circ vec^{-1} = vec^{-1} \circ vec = id$, where $id$ is the identity function.

Denote the goodness-of-fit score function by $l(\cdot)$, the number of nodes again by $p$ and $s = p \cdot (p - 1)/2$, the graph-estimation algorithm is given in Algorithm 3. The algorithm assumes a fixed family of distribution for each node; in practice, it is more convenient to assume a fixed family for a particular data type, e.g., the gamma family for positive data, the poisson family for count data and the binomial family for binary data. On the other hand, if one restricts the Gibbs update to only 0 and 1, i.e. one can only add or remove an arrow from a graph but cannot reverse an arrow, then it is in effect imposing an ordering on the nodes. To see this, since we are only changing the upper triangular part of the adjacency matrix, if we only allow the entries to take values 0 or 1, then on the graph, there can only be arrows from node $i$ to node $j$ for $i < j$, and hence an ordering of nodes is imposed. As a result, this eliminates the evaluation of $M_2$, and since under the new updating scheme, the adjacency matrix always corresponds to a DAG after each update, so this further eliminates the step of checking if the graph is acyclic. Overall, the performance is improved. We also remark that the idea of fixing a particular permutation / ordering of the nodes relates closely to the Chow–Liu tree (Chow and Liu, 1968).

Theory

What does the algorithm do underneath? To answer that, we need some auxiliary terminologies. First we associate each graph $G_i, i \in \mathcal{I}$ with its goodness-of-fit
Algorithm 3: Gibbs stochastic search on graphs

**Input:** Number of burn-ins $b$, number of iterations $m$, tuning parameter $\lambda > 0$.

- Initialise an adjacency matrix $G_0$ randomly.
- Vectorise it to get $I_0 = \text{vec}(G_0) = (i_1^{(0)}, i_2^{(0)}, \ldots, i_s^{(0)})$, where $i_k^{(0)} \in \{0, 1, 2\}$ for $k = 1, \ldots, s$.

for $j = 1$ to $b + m$ do
  for $k = 1$ to $s$ do
    $M_0 = (i_1^{(j)}, i_2^{(j)}, \ldots, i_k^{(j)}, 0, i_{k-1}^{(j)}, \ldots, i_s^{(j-1)})$,
    $M_1 = (i_1^{(j)}, i_2^{(j)}, \ldots, i_{k-1}^{(j)}, 1, i_{k+1}^{(j)}, \ldots, i_s^{(j-1)})$,
    $M_2 = (i_1^{(j)}, i_2^{(j)}, \ldots, i_{k-1}^{(j)}, 2, i_{k+1}^{(j)}, \ldots, i_s^{(j-1)})$.
  Compute for $i = 0, 1, 2$, the goodness-of-fit score $s_i = \begin{cases} l(M_i), & \text{if } \text{vec}^{-1}(M_i) \text{ is acyclic}, \\ -\infty, & \text{if } \text{vec}^{-1}(M_i) \text{ is cyclic}. \end{cases}$

Let $p_i = e^{\lambda s_i}/\sum_i e^{\lambda s_i}$, $i = 0, 1, 2$.

Draw $i_k^{(j)}$ from $\text{Multinomial}(p_0, p_1, p_2 = 1 - p_0 - p_1)$ and update $I_j = (i_1^{(j)}, i_2^{(j)}, \ldots, i_{k-1}^{(j)}, i_k^{(j)}, i_{k+1}^{(j)}, \ldots, i_s^{(j-1)})$.

end

if $j > b$ then
  Record $I_j$
end

- Form a matrix $B$ with row $r$ set to $I_{b+r}$, $r \in 1, 2, \ldots, m$.
- Compute the marginal distribution for each covariate, i.e. for each column of $B$, count the number of $\{0, 1, 2\}$ and divide each count by $m$. The result is a $3 \times s$ matrix $C$.
- Form $I_{\text{selected}} = (\arg\max_r c_{r,1}, \arg\max_r c_{r,2}, \ldots, \arg\max_r c_{r,s})$, i.e. collecting the element with the highest count in each column of $C$.

return the estimated graph $G_{\text{selected}} = \text{vec}^{-1}(I_{\text{selected}})$.

score, $s_i$ and construct a probability mass function $\mathcal{P}$ over the space of graphs by defining $p_i = \exp(\lambda s_i)/\sum_i \exp(\lambda s_i)$, $\lambda > 0$. Next, we define:

**Definition 7.** Graphical model $A$ nests graphical model $B$ if

1. all the edges presented in $B$ are present in $A$, and their orientations match with the ones of $A$;

2. for each node, the conditional distribution (given the other nodes) in Model $A$ matches the one in Model $B$. 

93
**Definition 8.** The best model is the model within the SGLM class that is closest to the true model in terms of KL divergence. When the true model is within the model class, then the best model is the true model.

**Definition 9.** A correct model is any model that nests the true model.

**Definition 10.** An incorrect / wrong model is any model that is not a correct model. For two wrong models A and B, A is less wrong if A nests B.

An example is shown in Figure 4.2. Suppose diagram (a) is the true DGP, then (b) is a correct model since it has all the edges in (a) and more (as indicated by the dotted arrows); (c) is an incorrect model since it misses two edges in (a) (from $x_1$ to $x_4$ and from $x_4$ to $x_5$). Here we assumed the arrows have matching distributional specifications across the diagrams.

Examining the algorithm closely, the for-loops in the algorithm sample from the space of graphs in a MCMC fashion with stationary distribution (of visits to each graph) equal to $P$. Each sample is stored in a row of the matrix $B$. As we sample according to the goodness-of-fitness score, correct models would show up more often than wrong models. The last three steps (in point-form) of the algorithm is equivalent to a majority vote. Since $B$ is expected to be occupied by more correct models than incorrect models, the majority vote would favor the correct ones. Finally, since all correct models nest the best model, the majority vote would reveal the best model.

Now we formally establish the theorems that make the algorithm work. The first theorem is that correct models have log-likelihood of order $\log \log n$ better than the best model almost surely, while wrong models have log-likelihood of order $n$ worse than the best model almost surely. We use an example to illustrate the idea before proceeding to the formal statement and proof. Consider Figure 4.2 again, diagram (a) corresponds to a factorisation of joint density:

$$f^{(a)}(x_1, x_2, x_3, x_4, x_5) = f_3^{(a)}(x_3|x_2, x_4, x_5) \cdot f_2^{(a)}(x_2|x_1) \cdot f_1^{(a)}(x_1) \cdot f_5^{(a)}(x_5|x_4) \cdot f_4^{(a)}(x_4|x_1),$$
diagram (b) corresponds to
\[ f^{(b)}(x_1, x_2, x_3, x_4, x_5) = f^{(b)}_3(x_3|x_2, x_4, x_5) \cdot f^{(b)}_2(x_2|x_1, x_4) \cdot f^{(b)}_1(x_1) \cdot f^{(b)}_5(x_5|x_1, x_2, x_4) \cdot f^{(b)}_4(x_4|x_1), \]
and diagram (c) corresponds to
\[ f^{(c)}(x_1, x_2, x_3, x_4, x_5) = f^{(c)}_3(x_3|x_2, x_4, x_5) \cdot f^{(c)}_2(x_2|x_1) \cdot f^{(c)}_1(x_1) \cdot f^{(c)}_5(x_5) \cdot f^{(c)}_4(x_4). \]

Denote the log-likelihood function corresponding to \( f^{(s)} \) by \( l^{(s)} \), and \( f^{(s)}_i \) by \( l^{(s)}_i \), our result says

(i) \( l^{(b)} \) has log-likelihood of order \( \log \log n \) better than \( l^{(a)} \), and

(ii) \( l^{(c)} \) has log-likelihood of order \( n \) worse than \( l^{(a)} \).

To see this, it suffices to match the components of one factorisation to those of another factorisation. In the first case,
\[ l^{(a)}_k = l^{(b)}_k, \quad k = 1, 3, 4 \quad \text{and} \quad l^{(a)}_k \leq l^{(b)}_k, \quad k = 2, 5 \]
By (4.1), \( l^{(b)}_k - l^{(a)}_k \) is \( O(\log \log n) \) a.s., \( k = 2, 5 \), hence adding the components would give \( \sum_{i=1}^5 (l^{(b)}_k - l^{(a)}_k) = l^{(b)} - l^{(a)} \) is \( O(\log \log n) \) a.s..

Similarly, in the second case,
\[ l^{(a)}_k = l^{(c)}_k, \quad k = 1, 2, 3 \quad \text{and} \quad l^{(c)}_k \leq l^{(a)}_k, \quad k = 4, 5 \]
By (4.2), \( n^{-1} \lim \inf (l^{(a)}_k - l^{(c)}_k) > 0 \) a.s., \( k = 4, 5 \), hence adding the components would give \( n^{-1} \lim \inf \sum_{i=1}^5 (l^{(a)}_k - l^{(c)}_k) = n^{-1} \lim \inf (l^{(a)} - l^{(c)}) > 0 \) a.s..

In summary, adding an extra arrow to the best graph (i.e. the graph associated to the best model, which is graph (a) in this case) would give an extra conditional term in the components, then the log-likelihood would increase by an order of \( \log \log n \) by (4.1). So the difference between the best model and a correct model is \( O(\log \log n) \). On the other hand, when an arrow is removed from the best graph, then the log-likelihood would decrease by an order of \( n \) by (4.2). So the difference between the best model and a wrong model is \( O(n) \). Here are the formal statement and proof:

**Theorem 14.** Denote the parameter associated with a correct model by \( \hat{\theta}_c \), the parameter associated with a wrong model by \( \hat{\theta}_w \) and the parameter associated with
the best model by $\theta_0$, then

$$||l_n(\theta_0) - l_n(\hat{\theta}_c)|| \text{ is } O(\log \log n) \quad a.s. \quad (4.3)$$

$$\liminf_{n \to \infty} \frac{1}{n} \left( l_n(\theta_0) - l_n(\hat{\theta}_w) \right) > 0 \quad a.s. \quad (4.4)$$

The first result quantifies the difference between the best model and the correct models, which is of order $\log \log n$. The second result quantifies how much better the best model is compared to a wrong model; the difference is of order $n$. Intuitively, it is easy to distinguish the best model and the bad model, but it is harder to identify the best one among the good ones.

**Proof.** The key idea has been illustrated earlier, here we simply put the argument in the general form. Denote the log-likelihood by

$$\log f^*(x; \theta) = \sum_{i=1}^{n} \sum_{k=1}^{p} \log f_k \left( x_{i,k} | x_{(i,k)}, \beta_{(i,k)} \right),$$

where the index $s \in \{*, c, w\}$ represents the best model, the correct model and the wrong model respectively.

By the definition of a correct model, we have $x^*_{(i,k)} \subseteq x^c_{(i,k)}, \forall k$. Applying (4.1) gives us

$$\sum_{i=1}^{n} \left[ \log f_k \left( x_{i,k} | x^c_{(i,k)}, \beta^c_{(i,k)} \right) - \log f_k \left( x_{i,k} | x^*_{(i,k)}, \beta^*_{(i,k)} \right) \right] \text{ is } O(\log \log n) \quad a.s., \quad \forall k$$

$$\Rightarrow \sum_{i=1}^{n} \sum_{k} \left[ \log f_k \left( x_{i,k} | x^c_{(i,k)}, \beta^c_{(i,k)} \right) - \log f_k \left( x_{i,k} | x^*_{(i,k)}, \beta^*_{(i,k)} \right) \right] \text{ is } O(\log \log n) \quad a.s. \quad (4.3)$$

This proves (4.3). Next, by the definition of a wrong model, there exists some index $k$ such that $x^w_{(i,k)} \subset x^*_{(i,k)}$. Group all the indexes satisfying this into a group labelled by $G_1$, and put the remaining indexes into another group $G_2$. Then by (4.2), we have for $k \in G_1$,

$$\liminf_{n \to \infty} \frac{1}{n} \sum_{i=1}^{n} \left[ \log f_k \left( x_{i,k} | x^w_{(i,k)}, \beta^w_{(i,k)} \right) - \log f_k \left( x_{i,k} | x^*_{(i,k)}, \beta^*_{(i,k)} \right) \right] > 0, \quad \forall k$$

$$\Rightarrow \liminf_{n \to \infty} \frac{1}{n} \sum_{i=1}^{n} \sum_{k} \left[ \log f_k \left( x_{i,k} | x^w_{(i,k)}, \beta^w_{(i,k)} \right) - \log f_k \left( x_{i,k} | x^*_{(i,k)}, \beta^*_{(i,k)} \right) \right] > 0,$
whereas for \( k \in G_2 \) we have a result similar to the earlier ones.

\[
\sum_{i=1}^{n} \sum_{k \in G_2} \left[ \log f_k \left( x_{i,k} | x_{-k}^w, \beta_{-k}^w \right) - \log f_k \left( x_{i,k} | x_{-k}^c, \beta_{-k}^c \right) \right] = O(\log \log n)
\]

Overall, the gain is of order \( \log \log n \), but the loss is of order \( n \). Hence, we have a loss of order \( n \) in total. This proves (4.4), and the proof is complete.

The second theorem establishes the consistency of the model selected by the algorithm.

**Theorem 15.** (Structural consistency) Let \( p_c \) be the probability that a sample from the stationary distribution \( P \) is a correct model and \( p_w = 1 - p_c \) be the probability that the sample is a wrong model. Then,

1. \( \frac{p_c}{p_w} \) is unbounded as the number of datapoints \( n \) goes to infinity,
2. for a sufficiently large \( n \), \( P(\text{the selected model is the best model}) \xrightarrow{a.s.} 1 \) as the number of MCMC steps / samples \( m \) goes to infinity.

**Proof.** Suppose we have \( p \) nodes and the number of edges in the best model is \( q \), then for some positive constant \( k_1, k_2 \),

\[
p_c \leq \frac{\sum_{i \in M_c} p_i}{\sum_{j \in M_w} p_j} = \frac{\sum_{i \in M_c} e^{s_i}}{\sum_{j \in M_w} e^{s_j}} \geq \frac{|M_c| \cdot \min_{i \in M_c} e^{s_i}}{|M_w| \cdot \max_{j \in M_w} e^{s_j}} \geq \frac{|M_c|}{|M_w|} \cdot e^{s_{\text{min}}} \cdot e^{k_1 \log \log n + k_2 n}
\]

where \( s_{\text{min}} := \min_{i \in M_c} s_i \), \( s_{\text{max}} := \max_{j \in M_w} s_j \), \(| \cdot |\) denotes the cardinality of a set, and in the last inequality, we used that \( s_{\text{min}} - s_{\text{max}} = (s_{\text{min}} - s_{\text{best}}) + (s_{\text{best}} - s_{\text{max}}) \geq -k_1 \log \log n + k_2 n \) for some positive \( k_1, k_2 \) by Theorem 14.

It follows that if we draw \( m \) samples (after stationarity is reached) from the model space and select the model \( M_{\text{selected}} \) by the majority vote, then

\[
P(\text{\( M_{\text{selected}} \) is correct}) \geq P \left( \frac{\sum_{i=1}^{m} Z_i}{m} > 0.5 \right) \xrightarrow{a.s.} 1,
\]

where \( Z_i \sim \text{Bern} \left( p = \sum_{i \in M_c} p_i \right) \), \( i = 1, 2, \ldots, m \) are i.i.d. Bernoulli random variables, representing whether a correct model is drawn. The first inequality is due to the fact that to win a majority vote, it is sufficient to gain 50% of the votes. And for the convergence, we used that \( \sum_{i=1}^{m} Z_i \xrightarrow{a.s.} \sum_{i \in M_c} p_i \) by the Strong Law of Large Number, and \( \sum_{i \in M_c} p_i > 0.5 \) for sufficiently large \( n \).
4.3.3 Model misspecification

When the model class is misspecified, the Maximum Likelihood Estimation (MLE) approach would asymptotically recover a model that minimises the Kullback-Leibler (KL) divergence with respect to the true model. We present a concise argument below, and readers are referred to (Claeskens et al., 2008, p.24) or (Boucheron et al., 2013, p.84) for more detailed derivations.

Suppose \( f \) is the density of the true model, and \( g_\theta \) represents the density of the specified model class, indexed by the parameter \( \theta \) in some parameter space \( \Theta \).

First, let

\[
\ell_n(\theta) = \frac{1}{n} \sum_{i=1}^{n} \log g_\theta(x_i), \quad \ell(\theta) = \int f(x) \log g_\theta(x) dx,
\]

then by the Strong Law of Large Numbers (SLLN), \( \ell_n(\theta) \xrightarrow{a.s.} \ell(\theta) \), and under appropriate natural conditions, \( \arg\max_{\theta \in \Theta} \ell_n(\theta) \xrightarrow{a.s.} \arg\max_{\theta \in \Theta} \ell(\theta) \). This suggests the MLE approach asymptotically maximises the negative cross-entropy, \( \ell(\theta) \). Next, note that

\[
\arg\max_{\theta \in \Theta} \ell(\theta) = \arg\min_{\theta \in \Theta} -\ell(\theta) = \arg\min_{\theta \in \Theta} \int f(x) \log f(x) dx - \ell(\theta)
\]

\[
= \arg\min_{\theta \in \Theta} \int f(x) \log f(x) - \int f(x) \log g_\theta(x) dx
\]

\[
= \arg\min_{\theta \in \Theta} \int f(x) \log \frac{f(x)}{g_\theta(x)} dx
\]

\[
= \arg\min_{\theta \in \Theta} D_{KL}(F||G_\theta)
\]

where \( F, G_\theta \) correspond to the probability measures associated with \( f, g_\theta \) respectively. This suggests maximising the negative cross-entropy is equivalent to minimising the Kullback-Leibler divergence. Combining with the previous result, we have shown MLE asymptotically recovers a model that minimises the KL divergence with respect to the true model as required.

4.4 Numerical studies

In this section, we present numerical studies to confirm the theoretical results, and we discuss some subtleties and practical issues related to the implementation.
The studies are set up as follows. We randomly generate a DAG, associate each node with a (randomly generated) GLM family and parameter coefficients, and then sample from the entire graphical model $n = 10000$ records of $p = 10$ measurements. Note that given $p$ variables, the number of possible DAG is greater than or equal to $2^{\binom{p}{2}} = 2^{45} \approx 10^{15}$. (If it takes 1s to evaluate a model, it will take more than 30 millions years to exhaust all the possibilities). We repeat this process for 1000 simulations, and for each simulation, we record the mean-square-error (MSE) of the estimated parameters (when the true DAG is given) and the proportion of missed edges in the estimated graph (when the true DAG is not given). We report the proportion instead of the number of missed edges because specifying the number of missed edges without the total can be misleading, e.g. missing 2 true edges out of 4 edges is much worse than missing 2 true edges out of 20 edges. A summary of the results are presented in Table 4.1, and the full distribution of the proportion of recovered edges is presented in Figure 4.3.

<table>
<thead>
<tr>
<th>MSE</th>
<th>Proportion of missed edges</th>
</tr>
</thead>
<tbody>
<tr>
<td>Min.</td>
<td>$6.047 \times 10^{-5}$</td>
</tr>
<tr>
<td>1st Qu.</td>
<td>$3.186 \times 10^{-4}$</td>
</tr>
<tr>
<td>Median</td>
<td>$5.305 \times 10^{-4}$</td>
</tr>
<tr>
<td>Mean</td>
<td>$6.722 \times 10^{-4}$</td>
</tr>
<tr>
<td>3rd Qu.</td>
<td>$8.676 \times 10^{-4}$</td>
</tr>
<tr>
<td>Max.</td>
<td>$4.524 \times 10^{-3}$</td>
</tr>
</tbody>
</table>

Table 4.1: SGLM numerical results

Over the 1000 simulations, the numerical results show consistency with the theoretical results. The parameters are reliably estimated when the graph is given, having an average MSE of $6.722 \times 10^{-4}$. Half of the simulated graphs (ground truth) have number of edges between 16 and 20 and an overall average of 17.96 edges; the estimated graph on average misses 29.83% of edges, i.e. about 5 edges on a 17-edge graph. The imperfect recovery is partly due to the finite-ness of the data and MCMC iterations and partly due to the consequent graph adjustment needed, which will be discussed in the second issue of the following.

Now we turn to discuss a few issues related to the implementation. The first is choosing the tuning parameter, $\lambda > 0$. In our simulation, this is set to be $n^{-0.5}$. Recall that $\lambda$ controls the probability of transiting to a new model, $e^{\lambda s_i}/\sum_i e^{\lambda s_i}$, and the smaller the value $\lambda$ takes, the more flat the multinomial distribution will be. If $\lambda$ takes a small value, then it is easy to transit between models, and the sampler would explore more models throughout the MCMC run. However, if $\lambda$ is set to too small a value, then the sampler may also waste a lot of time moving around the region where the goodness-of-fit scores are low. In practice, we
recommend setting $\lambda$ by running a grid search, e.g. $\lambda \in [1, 0.1, 0.01, 0.001]$, and then picking the $\lambda$ that gives the best goodness-of-fit scores.

The second issue is about selecting model by the majority vote. While in theory, the majority vote would (in the limit) select the best model which is a DAG, it may not be the case (i.e. not a DAG) for finite data and MCMC iterations. To solve this problem, we can further apply a threshold to the graph. Consider for example a graph with four nodes 1, 2, 3, 4 with the following marginal frequencies:

<table>
<thead>
<tr>
<th>Edges</th>
<th>{1-2}</th>
<th>{1-3}</th>
<th>{1-4}</th>
<th>{2-3}</th>
<th>{2-4}</th>
<th>{3-4}</th>
</tr>
</thead>
<tbody>
<tr>
<td>No edge (0)</td>
<td>0.1</td>
<td>0.25</td>
<td>0</td>
<td>0.1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Directed (1)</td>
<td>0.5</td>
<td>0.35</td>
<td>1</td>
<td>0.6</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Reverse (2)</td>
<td>0.4</td>
<td>0.4</td>
<td>0.3</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Majority Vote frequency > threshold $\phi$ ?

<table>
<thead>
<tr>
<th>$\phi = 0.4$</th>
<th>True</th>
<th>False</th>
<th>True</th>
<th>True</th>
<th>True</th>
<th>True</th>
</tr>
</thead>
</table>

Final DAG

<table>
<thead>
<tr>
<th>Majority Vote</th>
<th>(1)</th>
<th>(0)</th>
<th>(1)</th>
<th>(1)</th>
<th>(0)</th>
<th>(0)</th>
</tr>
</thead>
</table>

After the majority vote, the resulting graph \{1 $\rightarrow$ 2 $\rightarrow$ 3 $\rightarrow$ 1, 1 $\rightarrow$ 4\} is cyclic. We apply an additional threshold $\phi$, starting with 0 and gradually increasing it until a DAG is obtained. In our case, this happens at $\phi = 0.4$ and it gives us a DAG of \{1 $\rightarrow$ 2 $\rightarrow$ 3, 1 $\rightarrow$ 4\}. Note that it is guaranteed that there is such threshold to trim the cyclic graph into a DAG. This is because in the worst case,
one can always set the threshold to be 1, then the graph becomes a graph with no edge which is a DAG. Overall, an edge is only kept if it wins the majority vote and its marginal frequency is beyond the threshold. This resolves the potential cycles in the graphs, but the trade-off is that it may also remove the true edges “by accident”, so one expects the final graph to potentially miss a few edges in the true graph if trimming is required. We also remark that in case there are many redundant edges in the graph, increasing the threshold \( \phi \) also helps removing them.

The final issue is about the limit of identifying a DAG from data. The theoretical result presented in the previous sections implicitly assumed the existence and uniqueness of a Maximum-Likelihood estimate. In general, likelihood-based inference cannot distinguish models that are observationally equivalent, i.e. models that have equal likelihood to produce the observations, so the assumption is made to push through the arguments and proofs. However, suppose \( X \) and \( Y \) jointly follows the bivariate Gaussian distribution. Then the distribution admits two possible SGLM representation, one with \( X \) pointing toward \( Y \) representing the factorisation \( f_{X,Y}(x,y) = f_X(x)f_{Y\mid X}(y) \), where \( f_X, f_{Y\mid X} \) are Gaussian, and the other going in the opposite direction. In this case, SGLM would in theory fail to identify the structure as the ML estimate is not unique (though in practice, it might just end up in one of the two MLEs). It may seem worrying that SGLM would fail with such a simple example, but the problem actually goes away when the system becomes more complex, e.g. a random variable \( Z \sim \Gamma(1,1) \) pointing toward \( Y \) would break the symmetry and make the representation unique. In gist, the issue should be noted, but there should be little effect on practical applications.

### 4.5 Applications

The SGLM proposed (along with the structure learning algorithm) gives us a way to fit a multivariate distribution to data. In addition to exploring relationships between variables in the data, the fitted (generative) model also has other applications. We present three of them in this section, showing that the SGLM can be useful in confirming expert knowledge or discovering mechanisms of deterministic systems, in imputing missing data to give better parameter estimates, and in simulating realistic data for testing statistical methods.

#### 4.5.1 System identification

SGLM can be used to identify the mechanism underlying a deterministic system. We consider the example of Lorenz System which has been used to model, among
Figure 4.4: Simulated data from the Lorenz system

many things, chemical reactions and electric circuits. The dynamics of the system is governed by the Lorenz equations:

\[
x'(t) = \sigma(y(t) - x(t)) \quad (4.8)
\]
\[
y'(t) = x(t) \cdot (\rho - z(t)) - y(t) \quad (4.9)
\]
\[
z'(t) = x(t) \cdot y(t) - \beta \cdot z(t) \quad (4.10)
\]

We simulate data from the system with the parameter \( \sigma = 10, \beta = \frac{8}{3}, \rho = 28 \) and initial condition \( x(0) = 1, y(0) = 1, z(0) = 1 \) using finite differencing with \( \Delta t = 0.01 \) and 1000 periods. A plot of the simulated data is given in Figure 4.4. And we use SGLM to search for a systemic relation among the original and derived variables (derivatives and second-order terms) \( \{dx_i, dy_i, dz_i, x_i, y_i, z_i, x_i \cdot y_i, y_i \cdot z_i, x_i \cdot z_i\}_{i=0}^{999} \)
The system suggested by the SGLM (Gaussian family) is:

\[
\begin{align*}
    x &= 7.47 \cdot 10^{-15} - 0.1x' + y - 3.71 \cdot 10^{-17} \cdot xy \\
    y' &= 9.05 \cdot 10^{-14} - 2.8x' + 27y - xz \\
    z' &= 5.53 \cdot 10^{-13} + 2.92 \cdot 10^{-15}x - 2.666667z + xy
\end{align*}
\]

It is reasonable to discard coefficients smaller than \(10^{-12}\), and the system becomes

\[
\begin{align*}
    x &= -0.1x' + y \\
    y' &= -2.8x' + 27y - xz \\
    z' &= -2.666667z + xy
\end{align*}
\]

Rearranging gives

\[
\begin{align*}
    x' &= 10 \cdot (y - x), \\
    y' &= x(28 - z) - y, \\
    z' &= xy - 2.66667z
\end{align*}
\]

which agrees with the true system (4.8) - (4.10).

### 4.5.2 Imputations

SGLM may be used to handle missing data. We illustrate with an example where the data is missing-not-at-random (MNAR). We choose MNAR out of the MCAR, MAR and MNAR missing data mechanisms (as introduced in Section 1.3.1) because it is arguably the hardest case to handle. For \(i = 1, 2, ..., n\), suppose the data generating process is given by:

\[
\begin{align*}
    w_i &= \beta_1 x_{i1} + \beta_2 x_{i2} \\
    y_i &= 1_{w_i > 0.5} \\
    z_i &= \gamma_1 + \gamma_2 y_i + \epsilon_i
\end{align*}
\]

where \(\epsilon_i \sim N(0, 1)\). And we only observe \(y_i\) when \(w_i \leq 0.6\); we write

\[
y^*_i = \begin{cases} 
    y_i, & w_i \leq 0.6 \\
    \text{NA}, & w_i > 0.6
\end{cases}
\]

Note that the missing-ness depends on the value \(y_i\) takes (via \(w_i\)), so the missing data mechanism is MNAR.

Given the dataset \(\{x_{i1}, x_{i2}, y^*_i, z_i\}_{i=1,2,\ldots,n}\), suppose we are interested in estimating \(\gamma_1\) and \(\gamma_2\). From the set-up, since \(y_i\) takes value 1 only when \(w_i > 0.5\), but
the data-point will be lost when \( w_i > 0.6 \), so a lot of information in \( y_i \) is lost, and it is more difficult to recover \( \gamma_1 \) and \( \gamma_2 \). (It may be helpful to draw analogy with a electric circuit where it works only when the voltage reach a certain threshold, and blows-up when the voltage exceeds the maximum limit.)

We use SGLM to learn the structure of the data, use it to impute the missing data, then estimate the required parameters with the simple linear regression using \( z_i \) against \( x_{i1}, x_{i2} \) and the imputed \( y_i^* \). To fix ideas, we first present the result of one simulation. The true parameters are:

\[
\begin{align*}
    n &= 100, \quad \beta_1 = 1.944, \quad \beta_2 = -1.452, \quad \gamma_1 = 1.621, \quad \gamma_2 = 0.305
\end{align*}
\]

The parameters estimated ignoring the missing data are:

\[
\begin{align*}
    \gamma_1^{(\text{missing})} &= 1.649, \quad \gamma_2^{(\text{missing})} = 0.819,
\end{align*}
\]

and the parameters estimated using the imputed data are:

\[
\begin{align*}
    \gamma_1^{(\text{imputed})} &= 1.682, \quad \gamma_2^{(\text{imputed})} = 0.273.
\end{align*}
\]

In the first case, the mean-squared-error between the true and estimated parameters is 0.1325, while in the second case, the error is 0.002349.

The above result is expected. From the DGP, \( z_i = \gamma_1 + \gamma_2 y_i + \epsilon_i \), we have \( E(z_i|y_i = 0) = \gamma_1 \), so naturally, \( \hat{\gamma}_1 \) relates to \( \frac{1}{n^{(\text{missing})}} \left( \sum_{i:y_i=0} z_i \right) \), where \( n^{(\text{missing})} \) is the size of the set \( \{ i : y_i = 0 \} \). As \( y_i \) can only be missing when it takes value 1, so we should be able to estimate \( \gamma_1 \) reliably with or without imputing the data. In contrast, \( E(z_i|y_i = 1) = \gamma_1 + \gamma_2 \), so the estimate of \( \gamma_2 \) could improve significantly when we impute the data using the SGLM model. And this is exactly what we see from the above.

The simulation is repeated 1000 times; the average mean-squared-error between the true and estimated parameters in the missing-data case is 0.3566, while in the imputed-data case, the average mean-squared-error is 0.1627, showing that imputation (with our SGLM) reduces the error of the estimated parameters by half.

### 4.5.3 Data simulation

Analysing the performance of statistical methods often requires the knowledge of the ground truth, i.e. the data generating process of the data. Since this is not available in a real-world scenario (in fact, the goal is to recover it from data),
it is common to use simulated data instead. However, if the data are simulated using simplistic setting, it could give the wrong impression that the statistical methods under test work well, while in reality they fail to generalise to real-world situation. It is therefore important to be able to simulate realistic dataset, and we will present an example how SGLM may be used for that.

We use SGLM to simulate data that mimics the “Air Quality Data Set” as appeared in the UC Irvine Machine Learning Repository, available at https://archive.ics.uci.edu/ml/datasets/Air+Quality#. The dataset contains 9358 records with 15 numerical attributes. We exclude the ‘Date’ and ‘Time’ variables and fit a SGLM to the complete observations of the remaining 13 attributes. Then we simulate some data from the model, and plot the correlation matrices of the original data and the simulated data in Figure 4.5.

From the figure, we see that the simulated data capture some key features of the original data, e.g. the negative correlation ‘cross’ in the variable X7, the lack of correlation between the variables X11, X12 and X13 with the rest, the correlation matrix of X11, X12 and X13, and the (positive) correlation cluster in the variables X1 to X6 and X9 to X10. Overall, SGLM provides us a mean to simulate realistic dataset with complex correlation structure, which could be difficult to do manually.
4.6 Concluding remark

In this chapter, we proposed a flexible class of multivariate distribution constructed using a DAG and the GLM, and along with that, we provided a structure learning algorithm and showed its consistency in structure recovery. On the practical side, this gives a new tool for modelling multivariate distribution, and it may also be used to identify underlying mechanisms of deterministic systems, impute missing data for better parameter estimates, and simulate realistic data for testing statistical methods as we have seen in the application section. On the theoretical side, SGLM serves as a good example how the principle of hierarchical structure in the previous chapter can be used to tackle combinatorially hard problem. It also hints at a framework that uses discriminative models to construct generative models. In particular, the idea of using substructures to compose a global structure is not restricted to the use of GLM. Our result relies mainly on the model hierarchy and the corresponding (partial) ordering of goodness-of-fit scores across layers; the GLM assumption is made only to make the convergence proof more tractable. In general, the same idea can be applied to other regression models where similar hierarchical structure exists.
Bibliography


Minerva Access is the Institutional Repository of The University of Melbourne

Author/s: Kwok, Chun Fung

Title: Missing data analysis, combinatorial model selection and structure learning

Date: 2019

Persistent Link: http://hdl.handle.net/11343/228925

File Description: Final thesis file

Terms and Conditions: Copyright in works deposited in Minerva Access is retained by the copyright owner. The work may not be altered without permission from the copyright owner. Readers may only download, print and save electronic copies of whole works for their own personal non-commercial use. Any use that exceeds these limits requires permission from the copyright owner. Attribution is essential when quoting or paraphrasing from these works.