UNIVERSITY OF MILANO-BICOCCA
Department of Economics, Management and Statistics
DOCTOR OF PHILOSOPHY IN ECONOMICS AND STATISTICS
CURRICULUM: STATISTICS

ADVANCED ANALYTICS AND MACHINE LEARNING FOR INDUSTRIAL MANUFACTURING APPLICATIONS

Doctoral Dissertation of:
Chiara Galimberti

Supervisor:
Prof. Stefano Peluso

The Chair of the Doctoral Program:
Prof. Matteo Manera

XXXIV Cycle - January 2023

" - Senti, non importa quanto tempo ci vuole. Non devi pensare troppo in là in questo lavoro, se no diventi matto.
- Allora a cosa devo pensare?
- A oggi. Guarda che bella giornata."

P. Cognetti, Le otto montagne

Acknowlegment

This dissertation is the outcome of a four-year industrial PhD programme. I would like to express my gratitude to the University of Milano-Bicocca and Siemens Italy for giving me this opportunity. It has been really inspiring to be exposed to research whilst also facing challenges as a Data Scientist by engaging in company projects.

I am thankful to Prof. S. Peluso for supervising my research and teaching me how to improve my academic skills. My gratitude also goes to Federico Castelletti for supporting me with insightful advice and collaborating on common projects which set the groundwork for this dissertation.

This experience was further enriched also by a visiting period abroad at FAU Univeristy in Nürnberg, for which I have to acknowledge Prof. A. Harth for his hospitality at the Chair of Technical Information Systems. It was intriguing to work together with colleagues from diverse backgrounds and meet a wide array of researchers from different universities.

Moreover, I am grateful for all the conversations and contributions from my colleagues at Siemens, it was really valuable to be in such a dynamic and innovative environment.

A special thanks goes also to my PhD colleagues who shared with me this uphill journey, for all the laughter that made it easier to survive through troublesome days.

Last but not least, my deepest appreciation goes to my family and Cristian. Without your continuous encouragement and support, I would have not been able to endure all the ups and downs of these last years.

Chiara Galimberti
Milano
January 2023

To Cristian,
"I can't carry it for you... but I can carry you!"

Summary

The following list summarizes the scientific contributions included in this dissertation:

1. Co-authored chapter in book (to be published in 2023): **C. Galimberti** & M. Repetto, Editors: F. P. Appio, D. La Torre, F. Lazzeri, H. Masri and F. Schiavone. Impact of AI in Business and Society. *Routledge*

2. Paper: R. Borgoni, **C. Galimberti**, D. Zappa (2021). Identification of spatial defects in semiconductor manufacturing. *Applied Stochastic Models in Business and Industry*. `https://doi.org/10.1002/asmb.2615`

3. Post-Proceeding (accepted for publication): **C. Galimberti**, F. Castelletti, S. Peluso. Bayesian Multivariate Analysis of Mixed data. *Statistical Models and Methods for Data Science - 13th Scientific Meeting of the Classification and Data Analysis Group of the Italian Statistical Society. Springer*

4. Paper (under submission): **C. Galimberti** , S. Peluso, F. Castelletti. Bayesian inference of graph-based dependencies from mixed-type data

Abstract

STARTING from the biggest technology companies, there is a significant and continuous increase of investments in Artificial Intelligence. The main advantage of merging Machine Learning to industrial applications is mainly related to their performance and flexibility, also in complex contexts. This convergence sets the basis for the fourth industrial revolution leading to the *Industry 4.0* era and *smart manufacturing*.

In this dissertation, the focus is on the development of approaches capable to investigate industrial processes in order to address tasks such as predictive maintenance and monitoring.

Firstly, spatial analysis is shown to be effective in detecting and representing structured pattern of defects on integrated circuit fabrication. This issue is crucial to a production line since it can cause damages and yield loss. To address this hard-to-solve problem, the proposed approach is a concatenation of different methods, namely a p-value control chart, a clustering algorithm based on Mininum Spanning Tree and a graphical tool. The suggested procedure proves to be extremely fast and effective, allowing its implementation in-line during the fabrication process.

Secondly, graphical models provide an effective tool to represent conditional independences among variables. Especially in manufacturing, heterogeneous data often occurs (e.g. continuous, categorical) and it is of interest to discover interactions between different settings or measurements. Therefore, the goal is to define a suitable model able to describe a joint distribution of mixed type variables. The novelty of the proposed methodology is the definition of Bayesian graphical model starting from a Conditional Gaussian distribution suitable both for parameter inference and structure learning. Additionally, an MCMC scheme is implemented for approximate posterior inference in two alternative parametrizations, and a structure learning algorithm for related undirected graph models. This method shows outperforming results when compared to alternative state-of-the-art approaches in a simulation environment.

Contents

List of Figures

List of Tables

List of Acronyms

AI	Artificial Intelligence
AIC	Akaike Information Criterion
ANN	Artificial Neural Networks
BFGS	Broyden-Fletcher-Goldfarb-Shanno algorithm
BGM-MD	Bayesian Graphical Model Mixed Data
BIC	Bayesian Information Criterion
CG	Conditional Gaussian distribution
CMP	Chemical Mechanical Polishing/Planarization
CSR	Complete Spatial Randomness
CUSUM	CUmulative SUM control chart
DAG	Directed Acyclic Graph
DBSCAN	Density-Based Spatial Clustering of Applications with Noise
DL	Deep Learning
EMST	Euclidean Minimum Spanning Tree
EWMA	Exponentially Weighted Moving Average control chart
FN	False Negative
FP	False Positive
FPR	False Positive Rate
HCG	Homogeneous Conditional Gaussian
HMGM	High dimensional Mixed Graphical Model
HPP	Homogeneous Poisson Process
IDC	International Data Corporation
IoT	Internet of Things

MCC	Matthews Correlation Coefficient
MCMC	Markov Chain Monte Carlo
MH	Metropolis Hastings algorithm
MISR	MISspecification Rate
ML	Machine Learning
MN	Matrix-Normal distribution
MNW	Matrix-Normal-Wishart distribution
PRE	PREcision
RMSE	Root Mean Square Error
SEM	Scanning Electron Microscope
SEN	SENsitivity
SPC	Statistical process control
SPE	SPEcificity
TN	True Negative
TP	True Positive
UG	Undirected Graph

CHAPTER 1

Introduction [1]

1.1 Artificial Intelligence for business

Artificial Intelligence (AI) and all its related fields are in the midst of an optimism cycle called AI Summer [79]. The term describes the current period of increased investment and activity in AI following the AI winter that happened in the late 1990s. The present AI boom began about in the 10s with no signs of abating. The use of AI is becoming increasingly popular among big technology corporations, with Google, Meta, IBM, and Microsoft among those making significant investments. Manufacturing companies are not lagging behind in AI adoption. According to the research report by the International Data Corporation (IDC), the worldwide spending on AI and cognitive services in manufacturing companies is expected to reach 26.6 billion dollars in 2025. AI is expected to bring about significant changes in manufacturing. Some companies are using AI to help with product design. For example, Autodesk offers a program called Autodesk Fusion 360 that uses AI to help designers create products. The program can recommend designs based on what has been designed before, and it can also suggest changes to a design based on customer feedback. Other companies are using AI to help with production. A worth noting example is the MindSphere platform owned by Siemens that enables companies to use AI to optimize their production processes [70]. Several alternative tasks

[1] Section 1.1 is extracted from a co-authored chapter in *F.P. Appio, D. La Torre, F. Lazzeri, H. Masri and F. Schiavone (2023).Impact of AI in Business and Society. Routledge*

can be performed using built-in applications or developing new ones. For example, it is possible to build a recommender system to suggest optimal settings for production processes based on data collected from machines. In addition to these projects run by well-established companies, there is an emerging ecosystem of startups. These companies are developing AI technologies on par with their larger competitors and are receiving record amounts of investments. For example, in the car industry, Comma AI aims to build a full self-driving system like the one provided by Tesla [92]. What is setting the difference in comparison to the previous AI summers is the pervasive usage of Deep Learning (DL) techniques. In fact, during this time period, DL techniques experienced a significant expansion. DL is a subset of AI that is concerned with algorithms that can learn to represent and model data through multiple layers of abstraction, resulting in systems that can learn to perform tasks that are difficult to explicitly program. In DL, Artificial Neural Networks (ANNs) are an essential component. Originally ANNs were modeled after the human brain in that they are composed of a vast number of interconnected nodes, also called neurons. Given this complex structure, an ANN is capable of recognizing highly nonlinear patterns in the input data [17]. Moreover, ANNs can learn faster than canonical Expert Based Systems since they require less human intervention. Nowadays we know that ANNs are by no means mimicking the inner working of a biological brain. In fact, ANNs take from the biological brain only two components: neurons and axons. The first difference resides in how biological neurons work, which is more similar to the Perceptron as proposed by Rosenblatt than the current activation functions. Another worth noting difference is the learning process. As surveyed by Lillicrap et al. [71], there is no scientific evidence that the brain uses a backpropagation algorithm for learning. Regardless of these considerations, there has been an increase in the number of ANN applications to provide services and produce goods that we use daily due to the vast amount of data generated daily and the need to analyze it quickly and accurately. Another critical factor to consider from a practitioner's perspective is the democratization of such technology. ANNs were formerly reserved for specialists only. As new software is developed, they are becoming more user-friendly, which is encouraging. It means that a more significant number of people will be able to use them and benefit from their features. Furthermore, recent advances in software have established an environment that allows non-expert programmers to design and deploy complicated ANN structures without the need for specialized knowledge. These software comprehend ANN frameworks such as Tensorflow [1], Pytorch [85] and Flux [53, 54]. As well as models and results management tools, such as Weights and Biases, for managing the distribution and reproducibility of such models as well as their findings [7]. Several companies have started to use artificial neural networks in their daily operations. Services such as Spotify, which employs an ANN

to locate tunes that best match users' preferences, are a good illustration of this pervasiveness in practice [25]. Another example is Tesla's Autopilot, which is capable of providing an aided self-driving experience and has been available in cars since 2015 [27]. The above are only a few examples of how ANNs are becoming increasingly popular. They are often utilized to know how to serve the customers and improve their overall experience effectively. More specifically, ANN usage is becoming increasingly popular in a wide range of business fields. For example, ANNs are used to assess credit risk both of consumers and enterprises to predict whether or not a borrower will default on a loan [16, 67]. This is accomplished by teaching the network to recognize patterns in historical data pertaining to defaulting borrowers. Another application of ANN in business is in the prediction of consumer behavior [60]. Consumers' potential interests in products, websites they might visit, and types of advertisements they might click on can all be predicted using DL techniques. Customers' preferences can be used to create more targeted advertisements and to improve the overall customer experience. ANN is also used in business to improve decision-making. ANN can be used to improve the accuracy of algorithmic predictions, to improve the accuracy of forecasts, and to improve the accuracy of risk assessments, among other things. When businesses have better data, they can make better decisions about where to allocate their resources, how to respond to changing market conditions, and how to reduce the risk of making bad decisions [81]. Finally, ANN can be applied to business processes in order to increase their efficiency. Besides to numerical application of ANN, they can also be used to improve the accuracy of data entry, translations, and text recognition, to name a few applications. Businesses can save time and money by reducing the amount of human error that occurs during business processes as a result of this information.

1.2 Digital transformation in manufacturing

In the previous section, an introduction of a general-purpose AI and Deep Learning applications for Business is given. In this section, the focus is on the industrial framework, especially manufacturing.

As evidenced by a bibliometric analysis by Sahoo [91], data collection and analysis are crucial in the manufacturing industry. The achievement of continuous improvement requires a deep understanding about the various underlying causes of issues. This goal motivates the development of data-driven approaches. As well as business management research is increasingly moving towards business analytics for strategic decision-making, manufacturers are approaching digital transformation driven by (big) data to have a deeper knowledge of their production processes.

In the industrial landscape, during the last decades significant transformations were introduced in the manufacturing environment. The new era,

defined as *Industry 4.0*, is characterized by renovations both in physical and technological terms (e.g., Cyber-Physical systems, Internet of Things (IoT), Digital Twins). Regardless of the enabling technology, the main purpose of industrial transformation is to increase the resource efficiency and productivity to amplify the competitive power of companies, see [98]. Figure 1.1 shows the shifts of manufacturing paradigms through periods of time, highlighting the major milestones achieved [2] . The era of Industry 4.0 corresponds to the definition of *smart manufacturing* [61]. Specifically, after the subsequent implementation of strategies to optimize the organization of a productive system, the fourth industrial revolution is characterized mainly by the introduction of sensors and computational effort within the machinery.

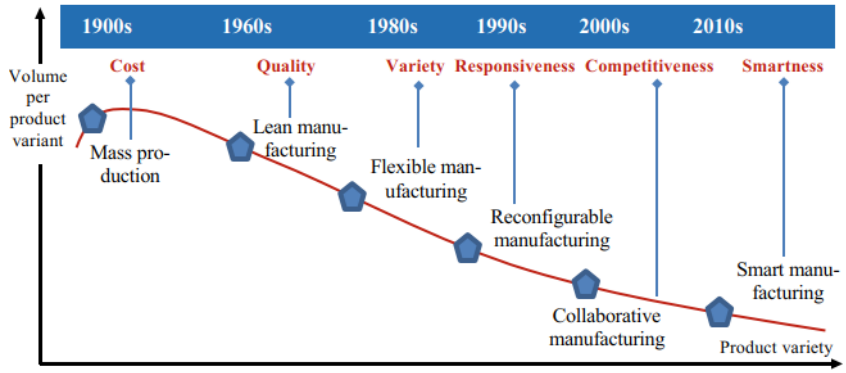


Figure 1.1: *Main paradigm shifts of manufacturing industry (from [68])*

The key for successful solutions for smart manufacturing is to combine and exploit advantages coming from technology and data collection with models and algorithms adapted to manufacturing processes. Given this two-fold perspective between industry and research, the general topic of Industry 4.0 is one of the most popular among industry and academia in recent years.

Within this research area, it is possible to find topics such as *big data analytics* and *predictive analytics*. An interesting discussion on the topic for manufacturing IoT is provided by Dai et al. [21]. The Authors provide a clear explanation of a general framework and lifecycle of analytics solutions in manufacturing, pointing out necessities and challenges in this field. Examples of research challenges are mainly related to the presence of heterogenous data and computational constraints for real time solutions. Consequently, also analytics applications need to undergo a transformation to solve new requirements from business. Additionally, there is a drift also in terms of purpose of analytics solutions: from a descriptive approach

[2] Figure 1.1 included with permission from the journal

the development aims to reach a prescriptive level, see Figure 1.2. This means that the focus is now on "what will happen" rather than "what happened", [95]. An an example, Figure 1.3 shows a classification of some typical use cases for analytics grouping the examples in terms of approaches and applications [3].

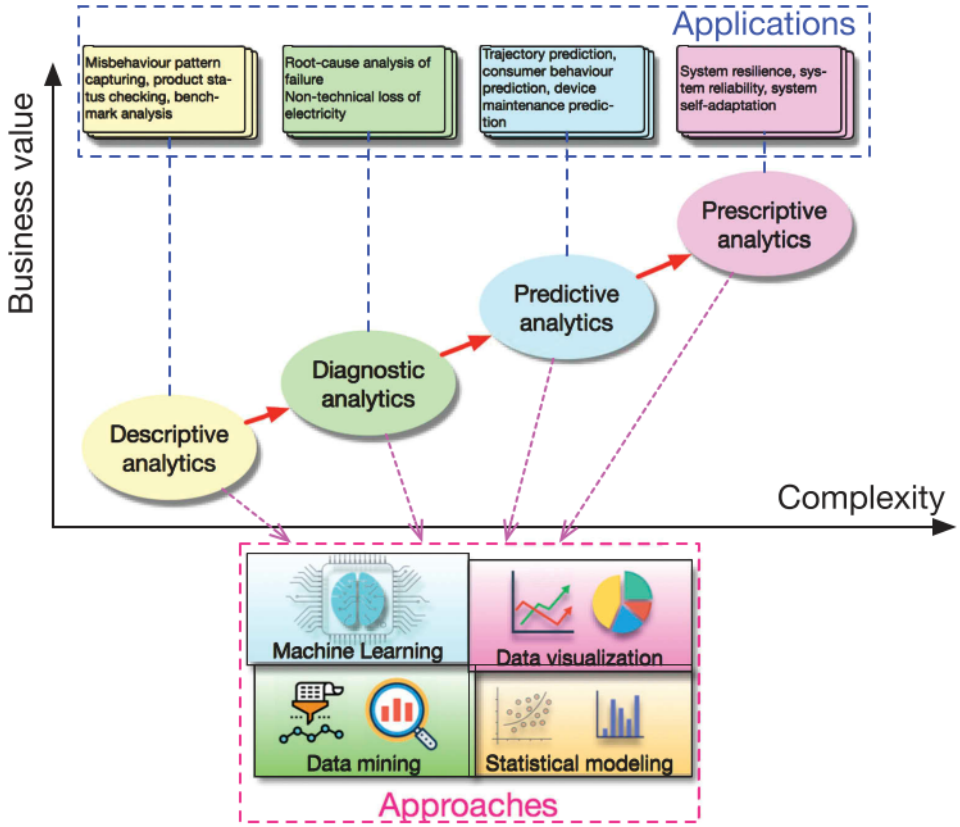


Figure 1.2: *Analytics roadmap (from [21])*

Many applications and references of predictive analytics can be found in [21, 95]. For instance, from descriptive to prescriptive use cases, it is possible to find respectively the following examples:

- Zuo et al. [110] use RFID data to analyze consumer purchase through an application of a Bayesian network
- Azadeh et al. [2] perform an analysis of faults in a mechanical component with a particular attention to data quality and noise
- Ren et al. [88] propose an automated solution for surface inspection using Deep Learning techniques

[3] Figure 1.2 and Figure 1.3 included with permission from the journal

	Questions	Approaches	Applications
Descriptive	What happened?	• Association rule mining • Clustering, sequential pattern mining • Querying, statistic reporting • Data visualisation	Misbehaviour pattern capturing Product status checking Benchmark analysis
Diagnostic	Why it happened?	• Reasoning • Bayesian analysis	Fault diagnosis Root-cause analysis of failure Anomaly detection
Predictive	What might happen in the future?	• Classification, regression • Machine learning (supervised/unsupervised) • Deep learning	Trajectory prediction Consumer behaviour prediction Device maintenance prediction
Prescriptive	What should be done?	• Simulation • Optimisation • Reinforcement learning (e.g., Q-Learning) • Decision making: e.g., Analytic Hierarchy Process (AHP), The Technique for Order of Preference by Similarity to Ideal Solution (TOPSIS)	System resilience System reliability System optimization

Figure 1.3: *Summary of Analytics applications (adapted from [21])*

- Drakaki et al. [28] define a recommender system for the optimal schedule for manufacturing using reinforcement learning

This dissertation merges four scientific products: a book chapter (Chapter 1), a paper published on an internationally recognized statistical journal (Chapter 2), a post-proceeding and an additional paper to be submitted in the next weeks (Chapter 3). Although the theoretical approaches differ, both proposals share common elements. The goal is to develop methodologies capable of solving typical problems in the industrial world such as predictive maintenance and monitoring of operations performed by machines within a process.

In the first work (Chapter 2), a three-steps approach is proposed in order to monitor when a process of integrated circuits goes out of control. The overall goal is to detect this drift and consequently analyze the highlighted defect structures. Specifically, the occurrence of defects on a planar surface is assumed to be modelled as a spatial stochastic process, and the presence of a defect pattern is defined when the occurrence of defects deviates from a completely random situation. So, the first step is used to monitor drift's occurrence in the production process, while the second and third steps finalize the identification of defect clusters with relative approximation of their shape and structure. This solution is intended to be implemented inline with the process so that it could be stopped when a structured quality issue is occurring. Additionally, it offers also tools to visualize the type of defects with an indication of an area of risk associated to the cluster's locations.

The second work has been extensively developed into two separate papers (see Summary), merged in Chapter 3. The first paper is a discussion of a preliminary theoretical definition of the proposed approach. The second

one presents the complete work including a more detailed description of the methodology and an evaluation of the proposal in comparison to alternatives. The proposed method is general-purpose so that it can be applied for several contexts and not only related to industry. The focus is on inference and representation of a conditional independence structure within a set of variables. The novelty introduced by this model is its capability of simultaneously handling numerical and categorical variables using a joint distribution in a Bayesian framework. This model also shows flexibility in its definition offering two alternative parametrizations connected by defined relations. The choice of the distribution representation can be conveniently done accordingly to the task to be performed (e.g., parameters estimate in unconstrained models, structure learning for graph based models). The implementation of the proposed method shows promising results both in a simulation environment and within real data applications, outperforming other alternative methods.

CHAPTER 2

Identification of spatial defects in semiconductor manufacturing [4]

2.1 Introduction

Spatial statistics has long played a fundamental role in environmental, agriculture and ecological studies as well as in economics, epidemiology and other human sciences. Less usual are applications in industry and technology where the spatial dimension is often crucial to assess the quality of production processes.

In this chapter, we investigate the spatial structure of defects occurring in integrated circuit fabrication. Integrated circuits are built through several different physical and chemical steps that are performed on thin silicon slices of a few inches in diameter, called wafers. Thousands of chips can be obtained from each single wafer and the precision of the manufactured product is essential [8]. Even though extremely accurate equipment in near dust-free rooms are typically employed, the presence of defects is almost unavoidable. Defects occurring on the wafer surface are the main cause of yield loss in the semiconductor industry. To quickly detect an excess of defects and their spatial structure is often crucial to the entire fabrication process.

Defect displacement over the wafer surface can be unstructured or structured. Unstructured defects tend to occur all over the wafer surface, mostly

[4] Article published in "Applied Stochastic Models in Business and Industry": https://doi.org/10.1002/asmb.2615

due to random causes, such as thermal variation or the presence of particles in the room during some specific process steps. On the contrary, structured defects tend to occur in clusters and can usually be ascribed to specific causes of the production process, i.e. human mistakes, drifts or faults in machinery performance. In this case, defects tend to show small scale patterns and their size, location and shape can convey a lot of information on their generative mechanism, which is fundamental to root-cause analysis. A prompt identification of these structures permits the reduction of waste and the need to rework production items, hence improving the yield.

Visual inspection of defects by human experts is time consuming, costly and prone to errors. Automated procedures and computer vision methods have been introduced over the last thirty years to inspect defects and identify their structure over the wafer surface.

In light of these considerations, this work proposes a methodology to detect and display spatial defect patterns effectively. The proposed approach is flexible, not relying on strong parametric assumptions, computationally convenient and easily implemented, possibly in-line, during the fabrication process.

More specifically, a control chart for the presence of spatially structured defects is first introduced to promptly grasp potential patterns on the wafer area. Then, a clustering procedure is proposed to identify the areas of the wafer prone to high defect concentration and to display their shape.

Statistical process control (SPC) charts [80] are widely used for monitoring the stability of different processes over time. First introduced in the manufacturing industries, they are now employed in other fields such as finance [29] and healthcare [38]. Control charts detect a process distributional shift when the charting statistic is beyond control limits. Developments in metrology provide the opportunity to improve process monitoring by obtaining many measurements on each sampled unit. A greater number of measurements can increase the sensitivity of control charts to detect flaws in local regions and extend the statistical process control to spatial-data monitoring. Scott et. al. [45], for instance, used variogram parameters to define spatial control charts to monitor the mean over time whereas Garthoff and Otto [37] focused on the mean and covariance charts for spatial autoregressive models to signal possible spatial changes via conventional EWMA and CUSUM charts.

The monitoring of defects in semiconductor manufacturing is an important issue that has been discussed in the technical literature for at least 3 decades. The book by May ad Spanos [77] reviews many statistical techniques adopted in this field and, in particular, considers control charts for defects and defect density. Conventional defect control charts assume that wafer defects follow a Poisson distribution [80]. However, the spatial displacement of defects tends to show higher heterogeneity due to an increasing complexity as the size of wafer increases. In the presence of clustering,

the Poisson assumption may be incorrect and standard control carts may point to sample measurements that fall outside the control limits. Revised defect control charts based on the Neyman Type-A distribution were suggested [35] to reduce the number of false alarms of conventional charts based on the Poisson assumption. Although this type of charts adjusts for clustering when monitoring the overall number of defects occurring in a wafer, they did not detect specific spatial patterns since the aggregate counts at the wafer level are considered.

As mentioned above, the spatial distribution of the local defects clusters frequently has assignable causes, hence indicating possible interventions. For instance linear or elongated clusters could often be the result of a scratch that occurred during material handling and shipping; the edge ring is typically ascribed to etching malfunctioning or rapid thermal annealing process; an excess vibration in equipment can possibly liberate particles and cause defects in nearby subregions; zonal patterns often arise from the thin film deposition; patterns located around the wafer centre or along the wafer edge may be caused by non-uniformities of the thin film deposition or an uneven temperature distribution in the rapid thermal annealing process, whereas unstructured-located defects are likely due to some contamination of the clean rooms or equipment. For this reason several techniques have been developed to detect the presence of spatial clustering of defects over time. Reviews are provided by Cunningham and MacKinnon [20] and Zhou et al. [107] amongst others.

Spatial statistics techniques [19] are often employed to detect non-random patterns. Many papers have tackled defect spatial clustering at chip level by comparing how many functional chips are around a defective chip and vice versa using appropriate measures of spatial dependency such as joint counts statistics [34,47,55]. Another possible approach is to count the number of defects that occur in every chip of the wafer and model the spatial correlations in 2D spatial counts. Shang et al. [94] have recently suggested modelling defect counts by intrinsic Gaussian Markov random fields in a hierarchical Bayesian framework, using Markov Chain Monte Carlo to estimate model parameters and, based on the estimates, developing monitoring schemes utilizing the multivariate exponentially weighted moving average procedure to detect shifts in the trend and spatial correlation. Monitoring structured defect occurrences on the wafer area by chip-aggregate statistics is, however, not the only possible way to address the problem. Nowadays equipment perform in-line inspection during a manufacturing process producing a wafer map that includes the number of defects, their size, and their location on the inspected wafer. Hence, defect monitoring can operate at the location rather than chip level. Lee-Ing et al. [66] suggested a clustering index that is calculated at the location level to develop a multivariate Hotelling T^2 chart for jointly monitoring the local clustering as the number of defects.

Given the nature of the defect dataset considered in the case study that motivated this work, the control chart proposed in this work follows the latter approach.

However, practitioners would be interested, not only in a decisional rule in determining wafer quality, but also in assessing how strong the defect signal is, in order to plan subsequent actions appropriately. For this reason, control charts based on p-values, hereinafter *pv-chart*, have been suggested by several authors (see [44] amongst others). Li et al. [69] thoroughly described this approach using the cumulative sum (CUSUM) charts [84] when mean shifts in Phase II SPC are of interest.

In pv-charts, the in-control distribution of the charting statistic (i.e. under the null hypothesis that the system is working correctly) is preliminary computed and the p-value of an appropriate significance test is obtained at any given time point. If the p-value is smaller than a pre-specified significance level, the chart points out a process distributional shift. As Li et al. [69] observed, pv-charts have some benefits if compared to conventional control charts. First, whereas the control limits are different for different types of conventional charts (e.g. one-sided or two-sided charts), a pv-chart always has a vertical axis in the range $[0, 1]$ and a unique control limit corresponding to the significance level. Secondly, it provides a measure of how the data suggest a potential out-of-control state even if it is not detected. Finally, when monitoring structured defect patterns using intra event spacing, as described below, and differently from many other situations in SPC, reference values for the charting statistics are not immediate to work out. In addition, for the problem considered in this chapter, benchmark datasets or long time series for the monitored statistics are not available to define the thresholds of a control chart. Monitoring the p-value naturally provides a reference value by the significance level of the associated test. Hence, in these circumstances, resorting to p-value charts is deemed an appropriate and convenient way to proceed.

Analogous to the majority of the papers mentioned in this section, we assume that the defect process is governed by a spatial point process on the wafer area and the chart is based on testing whether the observed point pattern conforms to a completely spatial random process. Complete spatial randomness [26] is a property that describes a spatial point process where each point is placed in space independently of any other point. Wafers with clustered defects are not completely spatially random and an automated test for the presence of complete spatial randomness is useful to assess a spatially structured defects pattern [20]. Hence, a p-value control chart for spatially structured defects can be naturally based on a statistical test for the complete randomness of the configuration of the event locations where the p-value is calculated using Monte Carlo simulations. Our approach does not require the aggregation of defects at the chip, die or wafer level nor the implementation of any data preprocessing (i.e. data transformation and

outlier identification) or model estimation, a step that characterises all the charting procedures mentioned above.

As already said, spatial clustering of defects can be due to a variety of reasons and the shape and location of the cluster may be informative on the potential malfunctioning of the production process. For this reason, a large number of studies on the automatic retrieval of spatial features of defect clusters in semiconductor manufacturing have been published based on a large variety of approaches. Model-based clustering algorithms via Bayesian inference was proposed by Hwang et al. [51] whereas Yuan et al. [104] suggested a multistep defect analysis where the model-based approach based on a mixture distribution is implemented in the final step to reduce computational time. Adjacency-clustering based on the Markov Random Field model has been suggested to identify the defect patterns and accurately predict the yield [49]. Traditional clustering algorithms for spatial data such as DBSCAN [32] have been modified and adopted to identify defect patterns [57]. Many studies tackled the problem of cluster identification as a classification issue categorising patterns in typical structures possibly after de-noising the data [83, 100]. Convolutional neural network methods were also proposed to this end (see, amongst others, Whang and Chen [101] and Jin et al [56] and references therein).

A large majority of the studies in the field, including almost all the papers mentioned above, investigates defect clustering moving from the wafer map i.e. a graphical representation of a silicon wafer at which all the good and defective dice are represented, hence the clustering is investigated at the chip level. The clustering algorithm suggested in this work follows a different route and investigates the defect patterns at the location level. In particular, the considered procedure is appropriate to identify non-convex possibly elongated-shaped clusters, such as those produced by scratches.

Scratches across the wafer area are recurrent and particularly detrimental to the yield. A scratch produces a sequence of defects that tend to be aligned along the scratch or to be slightly displaced, hence producing clusters of points usually difficult to estimate and display.

The procedure proposed in this chapter consists of two main steps. First, an agglomerative clustering algorithm based on minimum spanning tree (MST) is used to identify the main clusters. Then, the scratch shape is estimated using alpha-shapes, hence it is largely non parametric in nature. The flow chart reported in Figure 2.1 summarises the sequence of the proposed methodologies.

The chapter is organized as follows. In the next section the case study that motivated this work in the first place is introduced. In Section 2.3, a pv-chart for spatially structured defect patterns is proposed. In Section 2.4, the clustering detection algorithm is discussed along with its graphical representation. In Section 2.5, the procedure is applied to a dataset coming from the considered wafer fabrication process. Discussion of the main

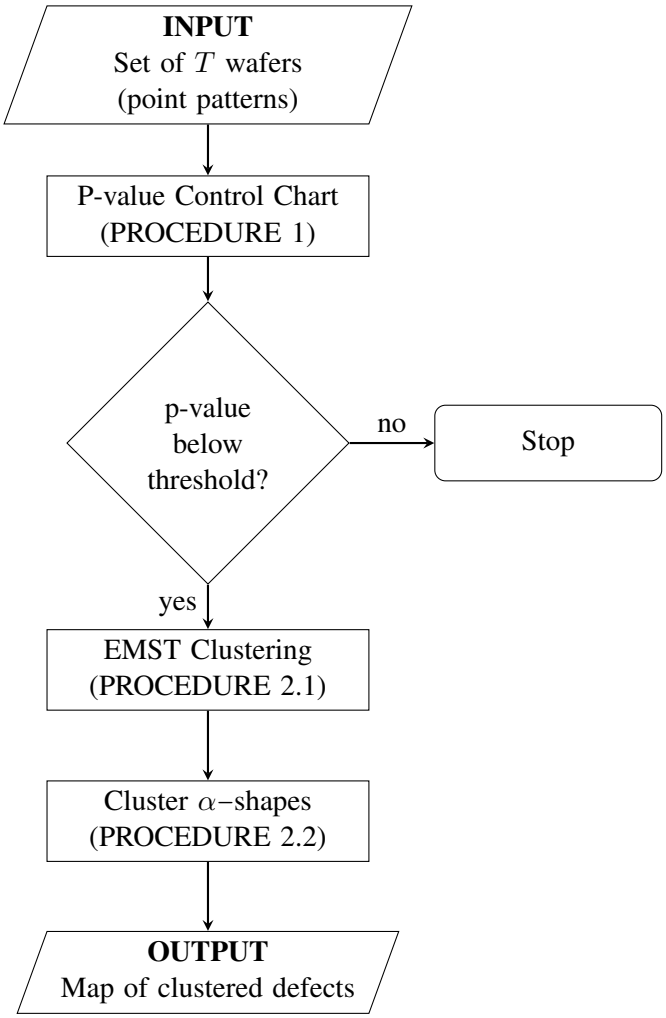


Figure 2.1: *Flow chart of the proposed methodologies. The procedures mentioned in the chart are described in detail in Section 2.3 and Section 2.4.*

results and conclusions in Section 2.6 end the chapter.

2.2 Defect data

One of the major sources of defects during chip fabrication is from dielectric Chemical Mechanical Polishing/Planarization (CMP). CMP is a process that removes materials to smooth surfaces. In CMP, a wafer is placed on a carrier and pressed into contact with a slurry film on a polishing pad. The planarization and polishing of the wafer is achieved through a mechanical abrasion and chemical etching [9]. One of the critical steps where CMP is used is when oxide is polished away until silicon nitride is fully exposed rendering the shape of the circuit desired by engineers. In the CMP process,

scratches formation originates [93] when the mechanical CMP action process is attributed to the polishing pad interactions with a slurry. A defect or a polishing scratch appears when for example particles of the residual oxide is not fully removed or when, because of the spinning of the wafer (i.e. high-speed rotation of the wafer), a particle run over the surface imprinting a trajectory [62]. Not all defects cause failure of a die; it mostly depends on both where they are located and on the size of the particle. To this aim, scanning electron microscope (SEM) images are used. After wafers are inspected using the inspection system, a wafer map that includes the number of defects, their sizes, and their locations on the inspected wafer, is produced.

In the dataset considered in this work, defect coordinates refer to the centre of a wafer having a diameter of approximately 8 inches. Defects were extremely small in dimension and were virtually identified by their punctual location. Data were collected from two different machines. 64 wafers from the first machine were sequentially inspected with the number of defects ranging from 11 to 91, the average number of defects per wafer being 37.6, whereas 6 wafers were inspected from the second machine with the number of defects ranging from 2 to 31 and the average defect number being 12.75. The map associated with the first four wafers inspected from the two machines are shown in Figure 2.2, where potential structured defect patterns seems to be present.

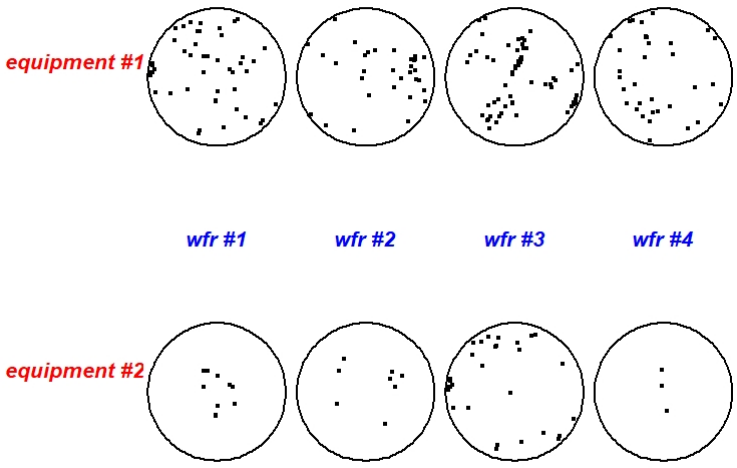


Figure 2.2: *Defect wafer maps obtained from the first and the second machine, first and second row of the panel respectively.*

2.3 The p-value charts for spatially structured defect patterns

Hereafter, a defect is represented by a point $\mathbf{x} = (x_1, x_2)$ occurring in a circular planar surface W representing a wafer. The spatial dynamics of defects on the wafer area is assumed to be governed by a spatial point process [26] occurring over W.

A spatial point process is a collection of random points such that for any open set B in W, the counting measure $N(B)$, representing the number of points in B, is finite. A point pattern $\mathbf{X} = (\mathbf{x}_1, \mathbf{x}_2, \ldots)$ is a finite collection of points in the area of interest and it is typically interpreted as a sample from (or a realisation of) the point process. In the actual problem considered here, a set of defects observed in W produces a point pattern. The lack of interaction among the points of a point pattern is called complete spatial randomness (CSR) property and it is typical of the Homogeneous Poisson Process (HPP) [26]. Let $N(B)$ be the random number of events that occur in a generic bounded region $B \subset W$, a spatial point process is Homogeneous Poisson or CSR if: i) $N(B)$ follows a Poisson distribution with expected value $\lambda|B|$ where $|B|$ is the size of B and $\lambda > 0$ is known as the intensity rate of the process, ii) for any choice of k disjoint bounded regions of W, the random variables $N(B_1), \ldots, N(B_k)$ are independent [52]. Assuming $N(B) = n$, then the conditional property of CSR states that these n points are independent and uniformly distributed in B.

Hypothesising that the CSR condition holds true for the defects occurring on the wafer area amounts to the assumption that no spatial structures are present and defects are somehow a physiological result of the fabrication process. Instead, structured defects can usually be ascribed to specific faults of the production process and a prompt identification can convey relevant information on the mechanism that originates the defects.

Testing the CSR condition requires a suitable test statistic that summarises the "distance between the data and the CSR null hypothesis". In particular, a common test based on the spacings or shortest distance distribution in a point pattern is considered below. More specifically, let R be the random variable representing the distance of a typical event of a point pattern to its nearest event in the sample, $G(r)$ the cumulative distribution function of R expected under the CSR condition and $\widehat{G}(r) = \frac{1}{n}\sum_{i=1}^{n} I(r_i < r)$ the empirical cumulative distribution function (ecdf) where r_i is the value of R for the i-th event in the sample. The functional $T(G, \widehat{G}) = \int_{\mathcal{R}}(G(r) - \widehat{G}(r))^2 dr$, where $\mathcal{R}$ is the support of R, is a measure of the divergence between $\widehat{G}(r)$ and $G(r)$. For practical calculations this functional is approximated by

$$D = D(G, \widehat{G}) = \sum_{j=1}^{m}[G(r_j) - \widehat{G}(r_j)]^2 \tag{2.1}$$

where the sum is taken over the grid of m values ranging from 0 to M, M

being an appropriate upper limit of the sample distances. Hence, a large value of D observed on the data does not support the CSR hypothesis.

The p-value can be obtained by approximating the distribution of D under the null hypothesis via Monte Carlo simulations. This is obtained by simulating a large number B of point patterns from the HPP. For the b-th simulated point pattern the ecdf $\widehat{G}_b^*(r)$ is computed at the m distances and D_b^* is also calculated. Finally, the p-value is obtained by $p = \frac{1+\sum_{b=1}^{B} I(D_b^* > D_{obs})}{B+1}$ where $D_{obs} = D(G, \widehat{G}_{obs})$ and $\widehat{G}_{obs}$ is the ecdf calculated on the observed point pattern.

To simulate a sample from a CSR process is computationally simple. A simple algorithm just mirrors the definition of the CSR property given above. First, a random draw is taken from a Poisson distribution to simulate the number n of defects occurring on the wafer area. Then, their coordinates are generated from a bivariate uniform distribution on the planar support. The intensity rate adopted when simulating a point pattern is the empirical intensity rate, which is the number of the observed defects divided by the area of the wafer, i.e. the maximum likelihood estimate of λ in the case of homogeneous Poisson processes.

A p-value control chart for spatially structured defects can be based on the statistical test for the CSR property described above. Similar to the CUSUM chart idea, the defects occurring up to a given time t are cumulated and the ecdf of the observed nearest neighbour distances is calculated on the cumulated point patterns. The test statistics D_t is calculated and the p-value, p_t, is worked out as described above.

The overall procedure to produce the chart is summarised in Procedure 1.

In order to construct the chart, the p_t values are reported on the vertical axis and the time t at which the p-value is calculated is reported on the horizontal axis. A horizontal line is also added to the chart at the value of statistical significance to be considered.

Since p-values are calculated using the distribution of test statistics which is actually estimated via simulations under the CSR hypothesis, they are affected by the variability induced by the simulation procedure. In order to control for this variability, we suggest the calculation of a confidence interval of the p-value by bootstrapping the simulated values D_b^*, $b = 1, \ldots, B$. The bootstrap percentile confidence interval can also be displayed on the chart by a pair of segments corresponding to the upper and lower limit of the interval. The process is considered out of control when the confidence interval of the p-value is entirely below the significance level α adopted for the statistical test. Some examples of this chart are reported in Section 2.5.

Procedure 1 p-value chart calculation

Input: $\mathbf{X}_1, \ldots, \mathbf{X}_T, W$ ▷ Sequence of point patterns and process window
Output: Calculate a vector of p-values to be plotted in the chart

procedure SUBPROC($\mathbf{X}$, W) ▷ p-value simulation
calculate the ecdf $\widehat{G}(r)$ using the observed point pattern $\mathbf{X}$
calculate $D = D(G, \widehat{G})$ in equation (2.1)
for b=1,. . . ,B **do**
simulate a point pattern $\mathbf{X}_b^*$ from a HPP
calculate $\widehat{G}_b^*(r)$ using $\mathbf{X}_b^*$ at m fixed values
calculate $D_b^* = D(G, \widehat{G}_b^*)$ in equation (2.1) and store its value
end for
calculate the p-value p
return p
end procedure
end procedure

Main procedure

set $\mathbf{N}_0 = \varnothing$ and $t = 1$
while $t \leq T$ **do** ▷ T: last inspection time
calculate $\mathbf{N}_t = \mathbf{N}_{t-1} \bigcup \mathbf{X}_t$
apply SUBPROC to $\mathbf{N}_t$ to calculate p_t
store p_t in a T-vector $\mathbf{v}$
replicate SUBPROC for B bootstrap replicates of $\mathbf{N}_t$
store percentile confidence intervals in a two-columns matrix $\mathbf{M}$
t=t+1
end while
return $\mathbf{v}$ and $\mathbf{M}$

2.3.1 A simulation study to explore the pv-chart performance

When exploring defects in manufacture artefacts, a prompt identification of the defect structure is a crucial point to reduce waste and item reworking, speeding up the fabrication process and reducing costs. Hereafter, a concise simulation study, which explores the performance of the approach suggested in the previous section, is reported.

As mentioned in the introduction, any type of structure in spatial defect patterns is somewhat suspicious and a potential symptom of a process malfunctioning. However, defects often tend to cluster in space and defect clustering is typically the major concern. For this reason, clustering structures have been considered in the simulation study presented below.

To simulate a clustered process, a clustering operation between point patterns is performed [52]: every point $\mathbf{x}$ of a given point process, named *the parent*, is replaced by a cluster $N_{\mathbf{x}}$ of points, named *the daughters* and their set-theoretic union is the realization of the clustered point process.

More specifically, to simulate defects, a clustered Neyman-Scott process has been adopted. Neyman-Scott processes originate from independent clustering operations applied to a stationary Poisson process. The parent points form an HPP with a given intensity and the daughter points are random in number and scattered independently and with identical distribution around their parent. Parents are only auxiliary constructs since they are not observable in reality and do not contribute to the final point pattern, which consists exclusively of the daughter points. As far as the distribution of the daughters, the clustered process considered in this section has a Thomas specification: the locations of the daughters are symmetrically distributed around their parent according to a Normal distribution with a scale parameter σ.

Since scratches across the wafer are of particular concern here, in this simulation study clusters are generated to have elongated non-convex shapes, which are typical in the presence of scratches. Assuming that the scratch shape can be adequately approximated piecewise linearly, parents are generated according to an HPP defined on a network of lines [3], which is kept fixed across the simulated wafers. The linear network that represents the support of the parent process is displayed in Figure 2.3.

The simulation design is structured as follows.

i) generate a point pattern using a Neyman-Scott-Thomas linear network clustered process on a circular window of radius 1;

ii) construct the pv-chart described in the previous section and save p-value and the out-of-control time if an out-of-control event actually occurs;

iii) repeat steps i) and ii) many times in different scenarios

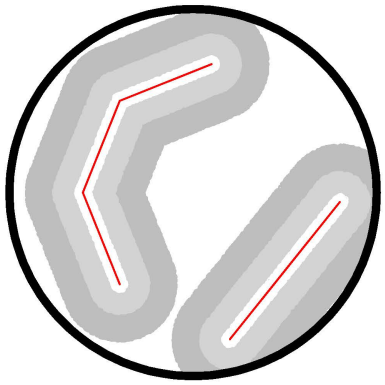


Figure 2.3: *Piecewise linear network. Shaded areas are the subregions of the wafer which are less than* 3σ *apart from the linear network and that represent the area more prone to defects for different values of* σ*:* $\sigma = 0.01$ *white area,* $\sigma = 0.05$ *light grey area and* $\sigma = 0.1$ *grey area.*

A maximum of 20 consecutive wafer inspections is considered in the simulation, assuming that an effective gauge would detect the spatial structure of defects in a reasonably small number of wafer inspections. For simplicity, we assume that one wafer is processed at a given time point and that the time points are equally spaced. The p-values are calculated using 250 Monte Carlo replicates for each simulated wafer. In order to investigate the pv-chart performance, step i) and ii) are repeated 1000 times. Data simulation mirrors the definition of clustered point processes given above. In step i), an HPP on the linear network is assumed to simulate parent locations. For each parent $\mathbf{x}$, the number of daughters $n_{\mathbf{x}}$ is simulated by a random draw from a Poisson distribution. Then, their coordinates of daughters are generated by drawing a random sample of size $n_{\mathbf{x}}$ from a bivariate Gaussian distribution centred at the parent location with a prefixed standard deviation

σ in both parent coordinates. To simulate the data, we set the expected number of the parent HPP on the network as big as 3 and the expected number of daughter points for each parent point also equal to 3. This guarantees a number of defects in the final simulated point pattern equal to roughly the average number of defects that we observed in the dataset used in the case study that will be presented later on in this chapter.

In a clustered pattern, more points occur at short distances than would be expected if the pattern were CSR and hence, the nearest-neighbour distance distribution function for a clustered pattern takes larger values than for homogeneous Poisson patterns, resulting in a bigger value of the test statistics (2.1) and a smaller p-value. The average out-of-control time and the average p-value are computed under different scenarios and reported in table 2.1. More specifically, the procedure has been repeated for different values of σ, namely $\sigma = 0.01$, $\sigma = 0.05$ and $\sigma = 0.1$. Figure 2.3 also shows the subregions of the wafer that are less than 3σ apart from the linear network for the different values of σ listed above. This area represents the region of the wafer which is "likely" to be involved by the defect clustering. When $\sigma = 0.01$, this prone-to-defect area is quite narrow (about 5% of the entire wafer area) and spatial clustering is expected to be strong. When $\sigma = 0.1$, the prone-to-defect area is extremely large, about 65% of the entire wafer area. In this case, clustering is extremely weak and hardly distinguishable from unstructured defects. For $\sigma = 0.05$, we still have a quite weak clustering scenario, although less extreme than the previous one, with about 30% of the entire wafer surface influenced by clustering.

A second scenario has been considered where a number of defects, not structured in space, have been added to the clustered sample in each iteration of the simulation study. More specifically, the noise component is simulated from a CSR process with an intensity rate equal to one third of the intensity rate observed in the simulated sample of clustered defects. This guarantees, on average, 25% of the overall simulated defects made by spatially unstructured events in each iteration.

σ^2 value	average time	average p-value	% of p-values ≤ 0.05
Without noise			
0.10	11.2	0.194	0.437
0.05	6.0	0.028	0.947
0.01	1.4	0.015	0.980
With noise			
0.10	11.0	0.356	0.206
0.05	11.0	0.116	0.635
0.01	2.0	0.019	0.986

Table 2.1: *Average time before out-of-control occurrence and average p-value obtained in 1000 simulations. Two scenarios are considered with and without (unstructured) spatial noise.*

Table 2.1 shows the results of the simulation exercise. It clearly appears that the suggested procedure is extremely efficient in detecting the clustering when clustering is strong or even moderate. In the case of a strong clustering (σ = 0.01), the average time before the pv-chart points out an out-of-control status is roughly 2 (i.e. on average, only two wafers are processed before the presence of clustering is detected) and a significant p-value (≤ 0.05) is found before the twentieth wafer in roughly 98% of the cases. Also, in the case of a moderate clustering, the chart performs well. In these two scenarios, i.e. strong and moderate clustering, the procedure is found to work reasonably well, also when CSR events are added to the simulated point patterns.

As one could expect, the procedure has a less brilliant performance in the case of a very mild clustering (σ = 0.1). In these circumstances, the pv-chart is not able to detect the clustering within the first twenty wafers in roughly 75% of the entire bunch of the performed simulations. The average p-value is quite high as well as the average time when the out-of-control status is detected. Indeed, this is an extreme situation and very difficult to detect with an extremely large area of the wafer, about 65% of the entire wafer surface, prone to defects (see Figure 2.3). In these circumstances, spatial clustering, although present, can hardly be distinguished from unstructured defectivity and, in practical applications, it is presumably not attributable to any local malfunctioning of the fabrication processes.

2.4 Graphical tools for spatial non-convex cluster detection

Once an out-of-control status is detected, further analyses are necessary to investigate the spatial structure of the defects over the wafer area. Hereafter, a non-parametric method is presented with this aim. The procedure consists of two main steps. First, an agglomerative clustering algorithm based on a minimum spanning tree is proposed to identify the main clusters that are present on the wafer. In the second step, the cluster shape is estimated using the alpha-shape of each detected cluster of defects. The procedure is sketched in Procedure 2[5] and is explained in detail in the next two sub-sections. All the algorithms presented in this chapter have been implemented in the R software [87].

2.4.1 Detection of defect clusters: the minimum spanning tree algorithm

In graph theory, a graph is an ordered pair of sets (V, E), where V is a set of nodes or vertices and E is a set of edges which are binary subsets of V, i.e. each edge is identified by a pair of nodes. Two nodes connected by an

[5] In Procedure 2, $|V|$ denotes the usual Lebesgue measure of a segment or a set V i.e. its length or area respectively.

edge are called adjacent. The edge can be directed (also called an arrow) or undirected (a line). The edge connecting two nodes α and β in V is a directed edge pointing to β if $(\alpha, \beta) \in E$ and $(\beta, \alpha) \notin E$. A sequence of adjacent nodes identifies a path. A graph is connected if any pair of nodes is connected by a path. A cycle is a path in which all the vertexes are distinct, except for the first and the last. A graph is called acyclic if it contains no cycles. A tree is a non-empty, connected, acyclic and undirected graph. A tree is called edge-weighted if a number is assigned to each edge, i.e. a weight function $w(\alpha, \beta)$ is defined for each edge $(\alpha, \beta) \in E$. The weight of the tree is the sum of all the weights associated to its constituent edges. The *Minumum Spanning Tree* is a particular tree such that the total edge weight is the minimum possible amongst all the possible spanning trees of (V, E). In particular, if the weight function is the distance between α and β, the tree is called a distance tree. If the adopted distance function is Euclidean, the tree is also called a Euclidean Minimum Spanning Tree (EMST from now on).

From the requirement that the sum of the edge weights must be minimum, the node pairs defining the edges represent points that tend to be close together; hence the EMST can be a basis for investigating clustering in a set V of defects.

This approach constructs the EMST on the set V for which efficient algorithms are known and implemented in many statistical and mathematical packages. In particular, the R library *spatgraphs* has been used in the case study presented in section 2.5.

Each edge in the EMST grown on V is the smallest edge connecting two partitions A and $V - A$ of V. Thus, points in different clusters of V should intuitively be connected by the longer edges of the EMST than points in the same cluster. Hence, the most relevant clusters of points can be obtained by cutting the longest edges in the EMST. Removing the longest edges of the EMST produces more homogeneous clusters. The elimination of the longest edge results in two-group clustering. Removal of the next longest edge results in three-group clustering and so on. This operation is usually called separating. The main idea of separating is that unusually long edges have to be removed from the EMST. The components of the EMST that remain connected at the end of the procedure identify the clusters. In this form, the EMST cluster detection contains all the information of a single linkage cluster analysis in the sense that the clusters formed at any distance d can be obtained from the EMST by deleting all segments of length greater than d and the dendrogram can also be derived from the EMST [42]. Given this, a clustering procedure based on EMST is prone to the same criticism as single linkage cluster analysis. In particular, it potentially produces, and it often does produce, elongated clusters so that a pair of points belonging to different clusters may be closer together than different members of the same cluster. Being prone to *chain* groups of objects in elongated struc-

tures, using isolated objects to link distinct groups, EMST may provide an advantage in defect clustering studies where scratches, which tend to expand by filamentary patterns of punctual defects through the wafer area, are often the main cause of defect patterns.

Cluster identification requires, however, a rule to remove some edges from the tree, sometimes called inconsistent edges. One of the seminal papers on clustering detection based on minimum spanning tree [105] suggests cutting those edges whose weights are significantly larger than the nearby edge weights, for instance k standard deviations larger than the average edge weights on each size. Zhou et al. [108] suggest to partition a set of points into a group of clusters by maximally reducing the overall standard deviation of the edges constructed from this set.

In our case, a reasonable strategy is to remove from the tree all the edges that are too long when compared to what one could expect under the CSR condition, hence representing potential bridges between two clusters. More specifically, each edge's length is compared to a scenario of an EMST computed on a point pattern originated by a CSR process. Then, an edge is cut if its length is greater than the percentile of order 0.9 of the expected average length $\bar{L}$ distribution of the edges of that EMST. Hence, the distribution of the average edge length of an EMST grown on a CSR point pattern has been simulated by repeatedly sampling an HPP. For each point pattern obtained in the simulation, the EMST is constructed and its average edge length is also calculated. Then, the length of each edge of the EMST obtained on the observed point pattern is compared to the 90% percentile of the simulated distribution. If the observed length is greater than this percentile, the edge is removed from the tree.

The exogenous selection of the cut-off to identify inconsistent edges is a somewhat critical point of many agglomerative or disjunctive hierarchical methods for cluster detection. The proposed procedure bypasses this problem, resorting to the distribution of the average edge expected assuming that the CSR property holds true to identify when two points occur too far apart.

2.4.2 Scratch cartography: α-shapes

As mentioned above, the identification of the scratch shape is one of the goals of an exploratory tool for spatial defects. The use of α-shapes to this end is described below.

Alpha-shapes were introduced by Edelsbrunner et al. [30] to provide a convenient method for delineating shapes in the plane by straight-line graphs. α-shapes are graphs, possibly disconnected, that generalize the convex hull of a set of points by weakening the convexity constraint, hence allowing for grasping elongated and concave shapes of a cluster. To build the graph, a circle of specified radius is used to link two joined points of the convex hull of V. The two points are connected by an edge if there are

no other points inside the circle or removed otherwise.

More specifically the α-shape [75] of a set of points $V = \{v_1, \ldots, v_n\}$, denoted hereafter by $\mathcal{A}_\alpha(V)$, is the graph with vertices $\{v_1, \ldots, v_n\}$ and edge set

$$E = \{v_i, v_j : \exists D_\alpha \; with \; v_i, v_j \in \partial D_\alpha \; and \; V \cap D_\alpha = \varnothing\}$$

where D_α is a disk of radius α, $\alpha > 0$, and ∂D denotes the boundary of D. For $\alpha < 0$ the disc is required to be of radius $-\alpha$ and $V \cap D_\alpha = V$. For $\alpha = 0$ the graph is the empty graph on V, i.e. the graph with no edges connecting the points in V.

The region identified, however, is sensitive to the value of α and typically selecting this value is crucial to grow the graph since it may produce disconnected polygons, polygons including holes with possible islands and so on. Tuning this parameter manually is not appropriate when this approach has to be implemented to depict the clusters occurring during a fabrication process and methods that automatically select a sensible value of α are definitely more convenient. Mandal and Murthy [74] suggested the following formula:

$$\alpha^* = \sqrt{\frac{l_n}{n}}$$

where n is the number of points in the set and l_n is the sum of the weights of the EMST edges connecting these points. In this work, we consider a different criterion. Since the α-shape is constructed separately for each cluster, to identify the influence area of a scratch, the value of α has to guarantee that the polygon is connected and covers a region close to the scratch in order to reliably capture the area where defects are likely to occur.

The idea of the procedure is to select α in such a way that the polygon is of the smallest possible area while still remaining connected. More specifically, a binary search is implemented in the interval $[0, d]$ where d is the diameter of the set of points, i.e. the maximum distance between a pair of points of V. Note that $\alpha = d$ produces the convex hull of V whereas $\alpha = 0$ the empty graph of V. The procedure repeatedly splits the search interval of the possible values of α trying to minimise the area of the α-shape under the constraint that the polygon is connected. More specifically, starting from the interval $[0, d]$, at each step the procedure constructs the α-shape with the parameter α corresponding to the midpoint, m, of the current search interval, say $[a, b]$, and checks whether the α-shape is connected. If so, the search interval is replaced by $[a, m]$, whereas it is replaced by $[a + \frac{b-m}{2}, b]$ otherwise. The procedure keeps going until the relative change in the area of the α-shape obtained in two successive steps is not below a fixed threshold ϵ, i.e. the algorithm stops at iteration $t - 1$ if $\frac{|\mathcal{A}_{\alpha_t}| - |\mathcal{A}_{\alpha_{t-1}}|}{|\mathcal{A}_{\alpha_{t-1}}|} < \epsilon$ where

$\mathcal{A}_{\alpha_t}$ is the alpha-shape of the observed point pattern of defects at iteration t.

2.5 Assessing defect structures in wafer manufacturing: pv-chart and cluster identification

In this section, the methodology discussed above is applied retrospectively to a dataset of defects that occurred in the microchip fabrication process. A set of wafers, processed by two machines in successive time points, have been considered. The coordinates of the defects that occurred during the production process were detected by a laser scan of their surface. The defect coordinates are obtained with respect to the centre of the wafer.

Figure 2.4 shows the CSR pv-charts obtained by adopting the procedure described in Section 2.3. Wafers were scanned one after the other and the defects that occurred at each time point were layered on the top of those previously scanned by the equipment. At each time point the test statistics in equation (2.1) were calculated on the set of cumulated defects and the p-value was computed using 10,000 simulated replicates drawn assuming that the CSR property holds true. The p-value was bootstrapped 1,000 times in order to obtain a 95% percentile confidence interval to account for the variability induced by the simulation process. The confidence intervals were displayed on the chart by a vertical bar drawn around the p-value obtained at each time point. A horizontal line has been also reported in the

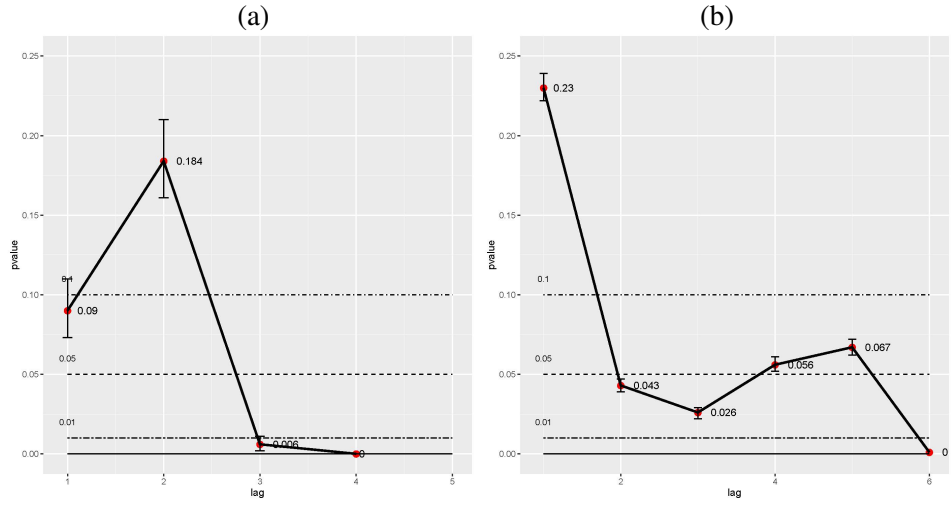


Figure 2.4: *P-value chart for the first (a) and the second (b) machine.*

graph at each of the typical significance levels adopted in common statistical tests, namely 0.1, 0.05 and 0.01. Assuming that 0.01 is the significance level actually considered, Figure 2.4-(b) suggested stopping the fabrication process and possibly intervening on the equipment after six wafers were

Procedure 2 Cluster identification and representation

Input: $\mathbf{X}$, W ▷ A point pattern and the process window
Output: α-shapes representing the clusters

procedure SUBPROC1($\mathbf{X}$, W) ▷ Simulation of the distribution of $\bar{L}$
$n = N(\mathbf{X})$
for $b : 1 \dots B$ **do**
simulate a point pattern $\mathbf{X}_b^*$ from a HPP with intensity rate $n/|W|$
calculate the EMST from $\mathbf{X}_b^*$
calculate $\bar{L}_b^*$ and store it in a B-vector $\mathbf{v}$
return $\mathbf{v}$
end for
end procedure
end procedure

procedure SUBPROC2(V,ϵ) ▷ Binary search of smallest connected α-shape
$\mathcal{A}_{\alpha_0} = ConvexHull(V)$ ▷ V: set of nodes
$a = 0$ and $b = d$ ▷ d: diameter of V
$s = 1$
while $\frac{|\mathcal{A}_{\alpha_s}| - |\mathcal{A}_{\alpha_{s-1}}|}{|\mathcal{A}_{\alpha_{s-1}}|} < \epsilon$ **do** ▷ $\mathcal{A}_\alpha$: α-shape of V
$m = \frac{a+b}{2}$
calculate $\mathcal{A}_m$
if $\mathcal{A}_m$ is connected **then**
set $b = m$, $\mathcal{A}_{\alpha_s} = \mathcal{A}_m$ and $s = s + 1$
else if $\mathcal{A}_m$ is not connected **then**
set $a = a + \frac{b-m}{2}$
end if
end while
return the last (smaller) connected α-shape found
end procedure
end procedure

Main procedure

Step 1 - agglomerative clustering algorithm based on EMST

calculate T, the EMST of $\mathbf{X}$
simulate the distribution of $\bar{L}$ usig SUBPROC1
for $e \in E$ **do** ▷ E: the set of edges of T
if $l \geq \bar{L}_{0.9}$ **then** ▷ $l = |e|$
add e in Λ ▷ Λ: list of edges
end if
end for
remove all edges in Λ from T
store all subtrees of T in τ ▷ τ: list of trees

Step 2 - cluster shape via alpha-shapes

for $t \in \tau$ **do**
apply SUBPROC2 to V_t ▷ V_t: set of nodes of t
store/plot $\mathcal{A}_{\alpha_t}$
end for

processed when the p-value and the entire bootstrap confidence interval lies completely below this value. Had one considered a significance level as large as 0.05 instead, the process could have been stopped even earlier, after the third wafer had been processed. The chart also sheds light on the status of the system even though the significance level had not actually been reached. In fact, by inspecting the results obtained by the pv-chart in the first five wafers, it clearly appears that the defect structure of the system is steadily remaining relevant since the p-value (and its confidence interval) tends to be remarkably low. A somewhat less defined pattern is shown by the second dataset that is displayed in Figure 2.4-(a). Here, the p-value is quite high at the second time point and then falls at the third point. In this case, it may be safer to monitor the system a few time points ahead to check whether the p-value stabilizes afterwards as it actually occurred for the dataset considered here.

In both cases considered above, the CSR property is not compatible with the data at hand. Hence, the procedure described in Section 2.4 has been applied in order to identify and represent spatial structures. First, the EMST was grown on the defects cumulated up to the wafer when the out-of-control status was detected and then the tree was separated using the Monte Carlo procedure described in Section 2.4.1. The results obtained for the two machines are reported in Figure 2.5 where clusters are superimposed in colour on the EMST grown on the defects detected by the two pieces of equipment. In both cases, the distribution of the average length of the MST edges has been estimated using 5,000 simulations.

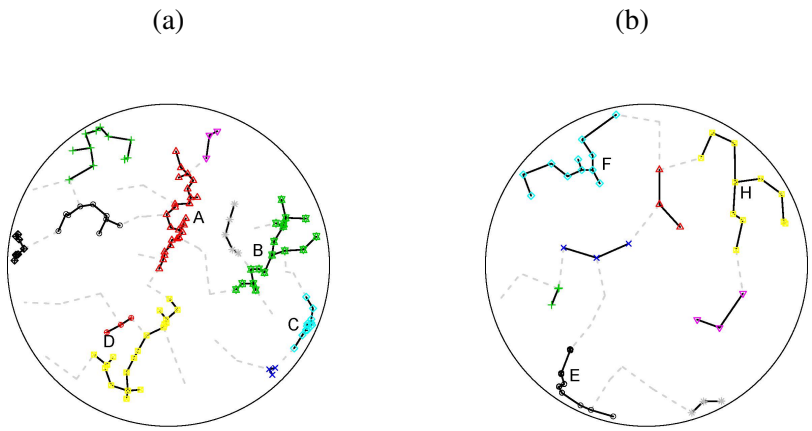


Figure 2.5: *Clusters of defects identified by the EMST algorithm for the first (a) and the second (b) machine. Points belonging to different clusters are depicted using different symbols and connected by continuous edges of the EMST. Dotted edges are those removed during the separating procedure.*

The skeleton of the sub-trees obtained by the procedure represents, however, a somewhat crude shape of the scratches that occurred on the wafer area during the fabrication process.

In addition, a more convenient representation of defect clustering should identify a region of the wafer area prone to defect. This kind of representation seems to be more appropriate than the skeleton to represent the area of influence of a scratch in subsequent wafers where a certain displacement of defect locations around the scratch is expected. Hence, α-shapes, described in Section 2.4, are used hereafter to estimate the shape of the clusters.

More specifically, α-shapes are constructed using the procedure described in the previous section for each cluster that includes more than ten defects, labelled "A" to "G" in Figure 2.5, assuming that clusters consisting of too few defects can be potentially spurious. The result of the procedure is depicted in the maps reported in Figure 2.6. The maps represent the regions of the wafer that are expected to be more prone to defects in the two production lines considered here.

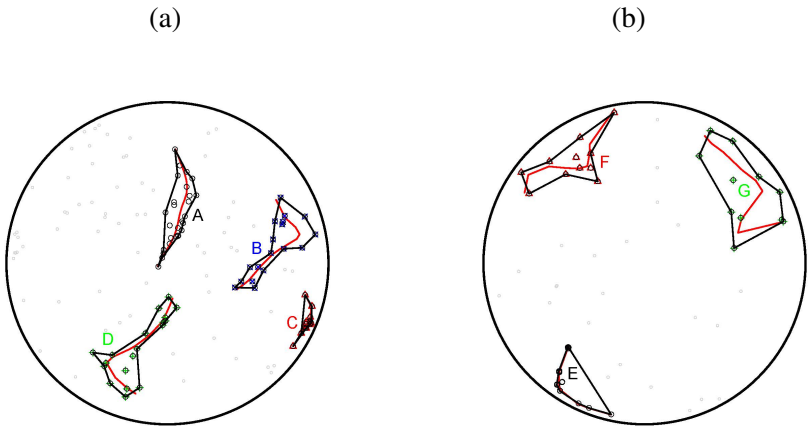


Figure 2.6: *α-shapes representing the cluster of defects found relevant in the first (a) and the second (b) machine. Lines represent the principal curves [48] estimated for each cluster. Points not included in any of the major clusters are represented in light grey.*

2.6 Discussion and Conclusions

This work proposes a procedure, composed of an ensemble of different methods, to detect defect patterns in wafer fabrication process control in the microelectronics industry. The procedure is based on two successive phases. First, assuming that defects are governed by a spatial point process, a p-value control chart is constructed to detect any possible spatial structure

of defect patterns using a test for the CSR property at each time point. The second phase aims to identify the shape and the location of defects. Particular attention is paid to the identification of clusters having complex, possibly non-convex, patterns typically difficult to detect and display. This phase is based on a minimum spanning tree algorithm coupled with alpha-shapes that are found to grasp such complex patterns effectively.

The entire procedure is non-parametric in nature since neither the EMST nor the α-shape algorithm requires any specific assumption on the mechanism that generates the data. Similarly, the Monte Carlo test adopted to implement the p-value control chart does not require any specific assumption on this mechanism and, in principle, can be adapted to different specifications of the latent process that generates the events (non necessarily the CSR one).

Taking into account the punctual nature of the data at hand, and differently to most of the literature in the field, we did not inspect defects at the chip level but felt it more appropriate to investigate the spatial locations of each single event. To the best of our knowledge, both parts of this procedure are new and have never been discussed in previous works. Although neither the EMST nor the alpha-shapes employed in the second phase are new algorithms, their use in this field is nonetheless original as are the procedures suggested to automatically tune the parameters necessary to prune the inconsistent edges off the EMST and to select the alpha-shape size.

We remark that the pv-chart considered in this work monitors the presence of a systematic structure in defect locations. Differently, the usual control charts used to monitor the defect process aim to identify "an excess" of defects on the wafer surface, possibly adjusting for the presence of clustering. These charts typically require a target value that has to be learnt from phase I studies or benchmark data whereas the procedure proposed here defines the target in terms of a stochastic property (i.e. the CSR condition) and uses the desired significance level of the test as reference value.

We also note that other methods of cluster detection and representation fit in the procedure proposed in this chapter. To exemplify this point, the DBSCAN algorithm [32] has been applied to our datasets instead of the EMST procedure described above. This algorithm produces quite similar results. Using the Rand index to compare the two clustering methods, we found that 85% of the defects are grouped similarly. However, as could be expected, DBSCAN tended to form more spherical clusters than the proposed procedure, hence proving to be less sensitive to elongated structures. Furthermore, the DBSCAN algorithm requires the tuning of an exogenous threshold to determine the neighbours. This value is usually determined by inspecting the k-nearest neighbour distances via a plot as we also did in this comparison. This approach, however, is suitable in off-line analysis whereas an automatic procedure, as the one suggested in this work, is more

convenient in in-line monitoring.

As mentioned in previous sections, non-convex elongated shaped clusters of defects are typically due to scratches occurring during the fabrication process that are often particularly detrimental to the yield. We recognise that some practitioners may feel unsatisfied with a set estimate of a scratch and would prefer a curvilinear estimation instead. As mentioned above, the skeleton of the sub-trees obtained in the second step of the clustering procedure suggested in this study provides a curvilinear estimate, although crude, of a scratch. Principal curves [48] can be used to get a smoother shape of these structures. Principal curves are a generalization of principal components that do not require an assumption of linearity. They are defined as one-dimensional curves that do not intersect or pass through the middle of a p dimensional data set, providing a non-linear summary of the data. Hastie and Stuetzle [48] also provided an algorithm to estimate the principal curves for a p-dimensional set of points of size n. The idea to employ principal curves to represent clusters having curvilinear patterns dates back to [97]. This approach has been more recently reconsidered in semiconductor defect data analysis by [51].

The use of principal curves also fits in the procedure proposed in this chapter. To exemplify this point, principal curves have been estimated separately for each cluster obtained by the algorithm proposed in Section 2.4 and superimposed on the α-shapes in Figure 2.6. It appears that the scratch shape obtained by principal curves and α-shapes looks similar. However, although principal curves may suggestively depict the scratch shape, α-shapes seems a more appropriate way to evaluate the influence area of a scratch, i.e. the region of the wafer where defects are expected to occur when processing a sequence of wafers.

The advantage of the procedure discussed in this chapter is twofold.

First, it is reasonably easy to implement and computationally efficient.

To implement the chart and derive the defect cartography only requires 262 seconds for the first piece of equipment considered in Section 2.5 (284 defects detected in the six wafers analysed) and 95 seconds for the second piece (52 defects detected in the four wafers analysed) using a CPU Intel(R) CORE (TM) i7 - 6700HQ at 2.60 Ghz with 16Gb of RAM. This makes it possible to plan in-line implementations of the tool along the fabrication process that permits prompt intervention if necessary. We notice that enhancements of MST-based clustering algorithms to improve the computational time have been suggested in recent papers (see [58] amongst others) and could easily be included in the procedure described in this work in the case of production processes characterised by a larger number of defects.

It also represents a powerful exploratory tool to identify effectively spatial structures in defect patterns that can be followed by more in-depth, possibly model based, off-line analysis of the clustering.

The procedure, in fact, has proven to be effective in detecting clusters of

defects. Having used prospectively retrospective off-line data, we were in the position to cross-validate our methodology using defect data collected on successive wafers processed by the two pieces of equipment considered in the previous section.

More specifically, for the first piece, we considered the next five wafers processed after the wafer in which the out-of-control status was detected and we calculated the percentage of defects that, in those successive wafers, lay in the four alpha-shapes depicted in Figure 2.6-(a). The percentage of defects occurring inside the four polygons ranged from 19.15% to 22%, i.e. roughly 20% of the defects in new wafers of the same production line is concentrated in 7% of the entire wafer area, 7% being the area of the wafer covered by the four alpha-shapes. This value is not far from 31%, which is the percentage of the defects falling within the four alpha-shapes in the first six wafers where the procedure was trained.

For the second piece of equipment, we only had two wafers processed after the out-of-control detection. The number of defects that occurred within the three alpha-shapes in Figure 2.6-(b) in these two wafers is 91.3% and 86.4%, respectively. These percentages are even higher than 63%, which is the percentage of defects falling within the three alpha-shapes in the first four wafers where the procedure was trained. The proportion of the wafer area covered by the three polygons is about 10.4%; hence defects are found extremely concentrated for the second machine.

Secondly, as mentioned above, the pv-chart described in Section 2.3 has proven to be extremely fast in detecting spatial structures, requiring the processing of only a few wafers. This gives a great advantage in modern microelectronics fabrication processes where items tend to be highly specialised, characterised by a short production life cycle and often produced in small lots. In these circumstances, a quick detection of defect patterns is often fundamental to avoid expensive yield loss.

Finally, different from other types of control charts, the pv-chart proposed here does not require gold standard data to be implemented since benchmark data are easily obtained via crude Monte Carlo simulations. This is another advantage for small lot production that typically does not permit the collection of charting statistics over time and, hence, to have benchmark data to set the chart.

CHAPTER 3

Bayesian inference of graph-based dependencies from mixed-type data [6]

3.1 Introduction

Understanding dependence relations between variables is an important task in several scientific domains, such as social sciences and biology. When the physical law describing the relationship between two quantities is unknown, this can be inferred from measurements collected under various conditions. In addition, when the system of interest entertains several variables, these can be organized in a graph which encodes a collection of dependence relations. The objective is therefore to estimate the graph structure, a process known as *structure learning*. Graphical *models* [63] provide a powerful framework to represent conditional dependence structures in multivariate distributions. This class of models adopt a graph-based representation to express the joint distribution of variables. The graphical structure imposes a set of pairwise conditional independencies between variables which in the case of parametric families corresponds to constraints on the parameters of the distribution. Moreover, the set of all conditional independencies encoded by a graph determines the so-called graph *Markov*

[6] A preliminary version of this work has been accepted for publication on *"Statistical Models and Methods for Data Science - 13th Scientific Meeting of the Classification and Data Analysis Group of the Italian Statistical Society"*, Springer, Editors: L. Grilli, M. Lupparelli, E. Rocco, C. Rampichini and M. Vichi.
The full work is under submission.

property.

Graphical models have been extensively studied in the Gaussian and categorical (multinomial) settings separately. Several methodologies for structure learning have been proposed under the two frameworks and following both a frequentist and Bayesian perspective. Among the former, the PC algorithm [59] and the graphical lasso [78] are the most popular methods, developed respectively for estimation of directed and undirected graphs. In the Bayesian framework, instead, Bhadra & Mallick [5] propose a Bayesian hierarchical graphical model for Gaussian variables on decomposable graphs. Moreover, Castelletti et al. [10] and Castelletti & Peluso [11] develop methodologies for structure learning of Dyrected Acyclic Graphs (DAGs) respectively for Gaussian and categorical variables whose results are applicable also to undirected decomposable graphs.

In many applied problems however, it is common to collect continuous and categorical measurements simultaneously. One example is medicine, where patients' characteristics are expressed as categorical variables, while clinical examinations are continuous measurements. A further field of application is in environmental engineering [96]: hydrocyclones provide a method for removing solids from liquids, as well as separating two liquids according to their densities. Unknown dependencies between continuous input flow rates, discrete and continuous material features, continuous environmental conditions, and categorical efficiency measures are common features of this setting. Another instance is nanotechnology, which develops functional structures designed at the atomic or molecular scale, related to optoelectronics, luminescent materials, lasing materials and biomedical imaging [4, 50, 72].

Contributions to graphical modeling for *mixed data*, namely both continuous and categorical, are relatively recent and quite narrow especially in the Bayesian framework.

Lee & Hastie [65] propose a frequentist methodology which jointly models the two set of variables by means of their conditional distributions; these corresponds to multiclass logistic and Gaussian linear regression models. They join a pairwise Markov random field model for categorical variables with a Gaussian graphical model for continuous variables and implement an undirected graph estimator that maximizes a sparsity-constrained likelihood. Similarly, Fellinghauer et al. [33] propose a method based on fitting separate l_1-regularized regressions for each variable.

Moreover, Chen et al. 2015 [13] consider a pairwise graphical model where the conditional distribution of each node is in the exponential family. They propose a neighbourhood selection approach to recover the structure of a mixed graphical model, by maximizing node-by-node penalized conditional likelihoods.

Similarly, Yang et al. [102, 103] introduce a novel class of graphical models based on generalized linear models, again assuming node-wise con-

ditional distributions in the exponential family. Much more recently, Zhuang et al. [109] propose an exponential family-based framework for graphical models using maximum likelihood estimate and sampling based approximation technique to infer Undirected Graphs (UG) for continuous and discrete data.

Following a decision-tree framework, Edwards et al. [31] propose a procedure to estimate a graph structure (either undirected or directed) based on an AIC and BIC criteria. Their method extends the algorithm of Chow & Liu [15], originally introduced for tree-based graphs. Specifically, they first select a minimal AIC or BIC graph having a forest structure by using penalized mutual information quantities. Next, they analyze separately each connected component using the method of Chow & Liu [15]. The resulting algorithm scales efficiently in high-dimensional settings and at the same time allows for dimension reduction.

Moving to a Bayesian framework, Bhadra et al. [6] propose a methodology for model selection of undirected graphs motivated by applications to cancer genomics. Here in particular, both categorical and continuous variables, such as the absence/presence of a mutation and protein expressions respectively, are typically available. Their approach is based on a Gaussian scale mixture representation of the marginal distributions of the two types of variables and can manage both continuous data that are not normal as well as discrete categorical data simultaneously.

Moreover, Zareifard et al. [106] propose a methodology for structure learning of directed graphs given continuous and categorical data. Their model assumes that categorical measurements are obtained by discretization of latent continuous variables and that the joint distribution of latent and observed Gaussian variables is multivariate normal. Accordingly, posterior inference on DAGs is conducted by implementing a Gibbs sampling algorithm based on data augmentation.

More recently, Cheng et al. [14] consider a conditional Gaussian (CG) model to analyze multivariate mixed data. This is based on the original work by Lauritzen & Wermuth [64] who define the general form of a conditional Gaussian model for variables whose distribution conditionally on a given level of categorical data is multivariate normal; the Authors also characterize the connection between the parameters of the CG distribution and the conditional independencies imposed to the the variables in the model. Starting from this model representation for mixed data, Cheng et al. [14] propose a simplified version of the CG model where parameter estimation is performed by maximizing the conditional log-likelihood of each variable with a lasso-type penalty. The methodology is then applied to high-dimensional settings to recover a sparse undirected graphical structure.

In this study we propose a Bayesian methodology for structure learning of undirected graphical models. More specifically, our framework is based on a CG distribution for the combination of categorical and continu-

ous data. We set up a Bayesian model by assigning a suitable class of prior distributions for the parameters of the distribution. Additionally, the model allows two alternative parameterizations: a *moment* and a *canonical* representation; the first one enables closed-form results in terms of parameter posterior distributions as well as marginal likelihood for an *unconstrained* (complete) graphical model. The second one instead provides an effective way to express conditional independence relations in the joint distribution, and in turn to learn the underlying graphical graphical structure.

Especially for the canonical representation, posterior inference on dependence parameters is carried out by resorting to Markov Chain Monte Carlo (MCMC) methods, which in turn allows to recover a possibly sparse graph structure. Our Bayesian methodology with canonical representation is compared with the alternative state-of-art method of Cheng et al. [14] with appreciable performances in simulation studies. The rest of the chapter is organized as follows. In Section 3.2 we introduce our model under the assumption of CG distribution. In Section 3.3 we complete our Bayesian model formulation by assigning suitable priors to the model parameters which describe associations/interactions between variables. Posterior inference on parameters and graph structures is also described in Section 3.3. We evaluate our methodology through simulation studies in Section 3.4 and apply it to the analysis of two real datasets in Section 3.5. In Section 3.6 the moment parametrization is introduced followed by some theoretical results. Finally, Section 3.7 concludes with a discussion.

3.2 Conditional Gaussian model

Accordingly to the approach proposed in the following sections, a general notation on graph theory is given. Conditional independencies between variables which characterize the *Markov property* of an undirected graph (UG) can be expressed through zero-constraints on the interaction parameters. To this end, let $\mathcal{G} = (V, E)$ be an UG, where V is the set of nodes and $E \subseteq V \times V$ the set of (undirected) edges. If $(u, v) \in E$, then also $(v, u) \in E$ and we say that $\mathcal{G}$ contains the undirected edge $u - v$; in such a case u and v are called *neighbors*. Moreover, for a given subset $A \subseteq V$, we let $\mathcal{G}_A = (A, E_A)$, with $E_A = \{(u, v) \in E \,|\, u \in A, v \in A\}$, be the *sub-graph* of $\mathcal{G}$ induced by A. We say that $\mathcal{G}_A$ is *complete* if all its nodes are connected. The UG $\mathcal{G}$ encodes a set of conditional independencies between variables which determine its Markov property. In particular, the absence of an edge $u - v$ in $\mathcal{G}$, that is $(u, v) \notin E$ implies $X_u \perp\!\!\!\perp X_v \,|\, \mathcal{X} \setminus \{X_u, \mathcal{X}_v\}$.

3.2.1 Conditional Gaussian distribution

Consider a collection of random variables $\mathcal{X} = (X_1, \ldots, X_{|V|})^\top$ indexed by the finite set V. We allow $\mathcal{X}$ to contain both categorical and continuous variables that are indexed respectively by $\Delta \cup \Gamma = V$. In what follows,

we will also refer to the collection of categorical and Gaussian variables as $Z_1, \dots, Z_p$ and $Y_1, \dots, Y_q$ respectively, so that $|\Gamma| = p$, $|\Delta| = q$. For such a combination of random quantities, Lauritzen and Wermuth [64] define a general class of probability distributions having the form

$$f(\boldsymbol{x}) = f(\boldsymbol{s}, \boldsymbol{y}) = \exp\left\{g(\boldsymbol{s}) + \boldsymbol{h}(\boldsymbol{s})^\top \boldsymbol{y} - \frac{1}{2}\boldsymbol{y}^\top \boldsymbol{K}(\boldsymbol{s})\boldsymbol{y}\right\}, \tag{3.1}$$

where $\boldsymbol{s}$ and $\mathbf{y}$ correspond to the multi-dimensional levels assumed by the categorical and continuous variables respectively, and $\boldsymbol{K}(\boldsymbol{s})$ is a $q \times q$ symmetric positive definite (s.p.d.) matrix. A probability distribution of the form (3.1) has a Conditional Gaussian (CG) distribution if and only if $\mathcal{X}_\Gamma \,|\, \mathcal{X}_\Delta = \boldsymbol{s} \sim \mathcal{N}_{|\Gamma|}(\boldsymbol{K}(\boldsymbol{s})^{-1}\boldsymbol{h}(\boldsymbol{s}), \boldsymbol{K}(\boldsymbol{s})^{-1})$, where $\mathcal{X}_A := (X_i)_{i \in A}$ for any $A \subseteq V$. Therefore, the distribution of the Gaussian variables conditionally on the configuration $\boldsymbol{s}$ of the categorical is multivariate Normal, and the marginal distribution of the categorical variables is such that

$$\Pr(\mathcal{X}_\Delta = \boldsymbol{s}) \equiv \theta(\boldsymbol{s}) = (2\pi)^{-\frac{|\Gamma|}{2}} |\boldsymbol{K}(\boldsymbol{s})|^{-\frac{1}{2}} \exp\left\{g(\boldsymbol{s}) + \frac{1}{2}\boldsymbol{h}(\boldsymbol{s})^\top \boldsymbol{K}(\boldsymbol{s})^{-1}\boldsymbol{h}(\boldsymbol{s})\right\}, \tag{3.2}$$

for each level $\boldsymbol{s}$ assumed by $\mathcal{X}_\Delta$. In addition, if $\boldsymbol{K}(\boldsymbol{s}) = \boldsymbol{K}$ for each configuration $\boldsymbol{s}$, a CG distribution is called *homogeneous* (HCG). Representation (3.1), which is based on the triplet $(g, \boldsymbol{h}, \boldsymbol{K})$ is named *canonical*. An alternative parameterization is given in terms of *moment-characteristics* parameters, see Section 3.6 for details.

Conditional independencies between variables in a CG distribution rely on the notion of *interaction*. Specifically, we can first express the canonical parameters $(g, \boldsymbol{h}, \boldsymbol{K})$ through the following expansions

$$g(\boldsymbol{s}) = \sum_{d : d \subseteq \Delta} \lambda_d(\boldsymbol{s}), \qquad \boldsymbol{h}(\boldsymbol{s}) = \sum_{d : d \subseteq \Delta} \boldsymbol{\eta}_d(\boldsymbol{s}), \qquad \boldsymbol{K}(\boldsymbol{s}) = \sum_{d : d \subseteq \Delta} \boldsymbol{\Phi}_d(\boldsymbol{s}), \tag{3.3}$$

where the new collection of parameters $\lambda_d(\boldsymbol{s})$, $\boldsymbol{\eta}_d(\boldsymbol{s})$ and $\boldsymbol{\Phi}_d(\boldsymbol{s})$ are called *interaction terms* and d represents any subset (including the null set) of Δ, the index set of the categorical variables. Each term of the expansions represents a type of interaction between variables. In particular,

- $\lambda_\varnothing$ is the *log normalizing constant*; λ_d $(d \neq \varnothing)$ are *pure discrete interactions* among variables in $d \subseteq \Delta$. If $|d| = 1$ they correspond to the *main effects* of the categorical variables;
- $\boldsymbol{\eta}_\varnothing$'s coordinates are the *main effects* of the continuous variables; $\boldsymbol{\eta}_d$ $(d \neq \varnothing)$ are *mixed linear interactions* between a continuous variable and variables in $d \subseteq \Delta$;
- $\Phi_\varnothing$'s elements are *pure quadratic interactions*; Φ_d $(d \neq \varnothing)$ are *mixed quadratic interaction matrices* between variables in $d \subseteq \Delta$ and pairs of continuous variables.

Following this representation, the distribution is homogeneous if and only if it has an interaction representation with no mixed quadratic interactions, and thus $\boldsymbol{K}(\boldsymbol{s}) = \boldsymbol{\Phi}_{\varnothing}$.

Interaction terms allow for a more direct characterization of conditional independencies between variables; accordingly, the *Markov property* of a given UG can be expressed through zero-constraints on such parameters. Let $\mathcal{G}$ be an UG; a CG distribution is said to be *nearest-neighbour Gibbs* with respect to a graph $\mathcal{G}$ (or $\mathcal{G}$-Gibbsian) if it has a representation with interaction terms satisfying

$$\begin{aligned} \lambda_d(s) &\equiv 0 \quad \text{unless } d \text{ is complete in } \mathcal{G}, \\ \boldsymbol{\eta}_d(s)_\gamma &\equiv 0 \quad \text{unless } d \cup \{\gamma\} \text{ is complete in } \mathcal{G}, \\ \boldsymbol{\Phi}_d^{\gamma\delta}(s) &\equiv 0 \quad \text{unless } d \cup \{\gamma, \delta\} \text{ is complete in } \mathcal{G}. \end{aligned} \tag{3.4}$$

where $\boldsymbol{\eta}_d(\boldsymbol{s})_\gamma$ denotes the γ-th element of $\boldsymbol{\eta}_d(\boldsymbol{s})$, corresponding to the continuous variable Y_γ (similarly for $\boldsymbol{\eta}_d(\boldsymbol{s})_\delta$) and $\boldsymbol{\Phi}_d^{\gamma\delta}$ is the (γ, δ)-element of $\boldsymbol{\Phi}_d(\boldsymbol{s})$.

Notice that a Gibbsian probability has an expansion with interactions terms involving variables that are neighbours only. Moreover, it can be proved that a CG-distribution is $\mathcal{G}$-Markovian if and only if it is $\mathcal{G}$-Gibbsian; see Proposition 3.1 in [64]. As a consequence, the joint density factorizes into a product of local densities that only depends on variables that are mutual neighbours. In addition, it can be shown that the so-obtained factorization splits up into separate factorizations of the constant, linear and quadratic terms; see Appendix B in [64].

Given the interpretation of all interaction terms, the model could be simplified in order to include in its specification only the parameters relevant to characterize the conditional independencies properties of the graph. We define a more simplified version of the model distribution in comparison with the assumptions made by Cheng et al. [14]. The idea is to avoid over-parametrization and information redundancy in representing all the possible pair-wise interactions in the graph. In what follows, we consider a simplified and homogeneous model by imposing the following conditions on the order of interactions:

- $|d| \leq 2$ for $\lambda_d(\boldsymbol{s})$,
- $|d| \leq 1$ for $\boldsymbol{\eta}_d(\boldsymbol{s})$,
- $d = \varnothing$ for $\boldsymbol{\Phi}_d(\boldsymbol{s}) = \boldsymbol{\Phi}_d$.

The simplified model omits all interaction terms between the categorical variables of order higher than two and it defines the canonical mean vector of the Gaussian variables as a linear function of the categorical variables instead of an "arbitrary" dependence function. The so-obtained distribution is also HCG since the precision (inverse-covariance) matrix $\boldsymbol{K} = \boldsymbol{\Phi}_{\varnothing}$ does

not depend anymore on the observed level $\boldsymbol{s}$ of the categorical variables. Finally, pair-wise conditional independencies between variables can be read off according to the following relationships:

$$\begin{aligned} Z_j \perp Z_k \,|\, \mathcal{X} \setminus \{Z_j, Z_k\} &\iff \lambda_{jk} = 0, \\ Z_j \perp Y_\gamma \,|\, \mathcal{X} \setminus \{Z_j, Y_\gamma\} &\iff \boldsymbol{\eta}_{j\gamma} = 0, \\ Y_\gamma \perp Y_\delta \,|\, \mathcal{X} \setminus \{Y_\gamma, Y_\delta\} &\iff \boldsymbol{\Phi}_{\varnothing}^{\gamma\delta} = 0. \end{aligned} \tag{3.5}$$

where Z_j, Z_k and Y_γ, Y_δ represent categorical and continuous variables respectively.

3.2.2 Likelihood function

Consider now n i.i.d. observations from the simplified and homogeneous specification of model (3.1) $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n$, with $\boldsymbol{x}_i = (z_{i,1}, \ldots, z_{i,p}, y_{i,1}, \ldots, y_{i,q})^\top$ for $i = 1, \ldots, n$ and the $(n, p+q)$ data matrix $\mathbf{X}$ obtained as row-binding of the individual observations. Let also $\boldsymbol{z}_i = (z_{i,1}, \ldots, z_{i,p})^\top$ and $\boldsymbol{y}_i = (y_{i,1}, \ldots, y_{i,q})^\top$. Categorical measurements in the dataset, corresponding to variables $Z_1, \ldots, Z_p$ can be equivalently represented through a contingency table of counts $\boldsymbol{N}$. In particular, for each configuration $\boldsymbol{s}$ of the categorical variable, we let $n(\boldsymbol{s}) = \sum_{i=1}^n \mathbb{1}\{\boldsymbol{z}_i = \boldsymbol{s}\}$ be the corresponding observed frequency in $\boldsymbol{N}$.

If we let $\mathcal{Z}_j$ be the set of levels assumed by Z_j, for $j = 1, \ldots, p$, then $(Z_1, \ldots, Z_p)$ takes value in the product space

$$\mathcal{Z} = \times_{j=1}^p \mathcal{Z}_j$$

which coincides with the cells of the contingency table $\boldsymbol{N}$.

Specifically, the relevant quantities to consider are,

$$\begin{aligned} n(j) &= \sum_{i=1}^n z_{ij} = \sum_{i=1}^n \mathbb{1}\{z_{ij} = 1\}, \\ n(k, j) &= \sum_{i=1}^n z_{ij} z_{ik} = \sum_{i=1}^n \mathbb{1}\{z_{ij} = 1, z_{ik} = 1\}, \end{aligned} \tag{3.6}$$

corresponding to the (marginal) number of one values assumed by single and pairs of variables respectively.

In what follows, we also assume for simplicity all categorical variables being binary.

The likelihood function can be written as

$$
\begin{aligned}
f(\boldsymbol{X}\,|\,\boldsymbol{\theta}) &= \prod_{i=1}^{n} f(\boldsymbol{x}_i\,|\,\boldsymbol{\theta}) \\
&:= \prod_{i=1}^{n} \exp\left\{ \sum_{|d|\leq 2} \lambda_d(\boldsymbol{s}) + \frac{1}{2}\left(\sum_{|d|\leq 1} \boldsymbol{\eta}_d(\boldsymbol{s}) \right)^T \boldsymbol{\Phi}_{\varnothing} \sum_{|d|\leq 1} \boldsymbol{\eta}_d(\boldsymbol{s}) \right\} \\
&= \prod_{i=1}^{n} \exp\left\{ \left[\lambda_0 + \sum_{j=1}^{p} \lambda_j z_{ij} + \sum_{j<k} \lambda_{jk} z_{ij} z_{ik} \right] + \boldsymbol{y}_i^{\top} \left[\boldsymbol{\eta}_0 + \sum_{j=1}^{p} \boldsymbol{\eta}_j z_{ij} \right] - \frac{1}{2} \boldsymbol{y}_i^{\top} \boldsymbol{\Phi}_{\varnothing} \boldsymbol{y}_i \right\} \\
&= \exp\left\{ n\lambda_0 + \sum_{j=1}^{p} \lambda_j n(j) + \sum_{k<j} \lambda_{kj} n(k,j) + n\boldsymbol{\eta}_{\mathbf{0}}^{\top} \bar{\mathbf{y}} + \sum_{j=1}^{p} \boldsymbol{\eta}_{\boldsymbol{j}}^{\top} \boldsymbol{t}(j) - \frac{1}{2} \mathrm{tr}(\boldsymbol{R}\boldsymbol{\Phi}_{\varnothing}) \right\},
\end{aligned}
\tag{3.7}
$$

where $f(\boldsymbol{x}_i\,|\,\boldsymbol{\theta})$ is given in Equation (3.1), now specialized to the simplified homogeneous model, and with the dependence on $\boldsymbol{\theta}$ emphasized. Furthermore, $\bar{\mathbf{y}}$ is vector-sample mean of $Y_1, \ldots, Y_q$, $\boldsymbol{t}(j)$ represents the interaction vector between the two types of variables and $\boldsymbol{R}$ is the interaction matrix of continuous variables; more specifically:

$$
\begin{aligned}
\bar{\mathbf{y}} &= \frac{1}{n} \sum_{i=1}^{n} \mathbf{y}_i \in \mathbb{R}^q, \\
\boldsymbol{t}(j) &= \left(\sum_{i=1}^{n} y_{il} z_{ij} \right)_{l=1,\ldots,q} \in \mathbb{R}^q, \\
\boldsymbol{R} &= \sum_{i=1}^{n} \mathbf{y}_i \mathbf{y}_i^{\top} \in \mathbb{R}^{q\times q}
\end{aligned}
\tag{3.8}
$$

Starting from this model specification, it is possible to obtain a closed-form expression for the log-normalizing constant λ_0, simply by integrating over the domain of the continuous variables $Y_1, \ldots, Y_q$ and summing over the domain of the categorical variables, $\mathcal{Z}$.

In the case of HCG distribution we obtain

$$
\begin{aligned}
\exp(\lambda_0)^{-1} &= \sum_{\boldsymbol{s}\in\mathcal{Z}} \Pr(\mathcal{X}_{\Delta} = \boldsymbol{s}) \\
&= (2\pi)^{-\frac{q}{2}} \det(\boldsymbol{\Phi}_{\varnothing})^{\frac{1}{2}} \sum_{\boldsymbol{s}\in\mathcal{Z}} \exp\left\{ \sum_{|d|\leq 2} \lambda_d(\boldsymbol{s}) + \frac{1}{2}\left(\sum_{|d|\leq 1} \boldsymbol{\eta}_d(\boldsymbol{s}) \right)^T \boldsymbol{\Phi}_{\varnothing} \sum_{|d|\leq 1} \boldsymbol{\eta}_d(\boldsymbol{s}) \right\},
\end{aligned}
\tag{3.9}
$$

where $\Pr(\mathcal{X}_{\Delta} = \boldsymbol{s})$ is the marginal distribution of $\mathcal{X}_{\Delta}$ obtained after integration over $\mathbf{y}$.

Therefore, the value of λ_0 can be obtained as

$$
\lambda_0 = -\log\left(\sum_{\boldsymbol{s}\in\mathcal{Z}} \Pr(\mathcal{X}_{\Delta} = \boldsymbol{s}) \right). \tag{3.10}
$$

3.3 Bayesian inference

We approximate the posterior distribution of model parameters by resorting to the MCMC strategy provided by the R package `MCMCPack` [76] which is based on a random walk Metropolis-Hastings (MH) algorithm; see [39, 89]. Since we are interested in estimating also the graphical structure underlying the data, we consider the canonical model representation. Accordingly, starting from the likelihood function in (3.7), we assign the following independent prior distributions

$$\lambda_j, \lambda_{jk} \sim \mathcal{N}(0,1), \quad \boldsymbol{\eta}_0, \boldsymbol{\eta}_j \sim \mathcal{N}_p(\mathbf{0}, \mathbf{I}), \quad \boldsymbol{\Phi}_{\varnothing} \sim \mathcal{W}_q(a_\Omega, \boldsymbol{U}) \tag{3.11}$$

where $a_\Omega = q$ and $\boldsymbol{U} = \boldsymbol{I}_q$.

The Metropolis Hastings (MH) algorithm applies to parameter posterior distributions from which direct sampling is not possible. Accordingly, the output is a sequence of random samples approximately drawn from the posterior distribution. At each step the algorithm proceeds by drawing candidate values from a proposal distribution (Gaussian in our case) and the proposed parameter is accepted with a probability given by the MH ratio. To initialize the algorithm, we first optimize the posterior distribution with the BFGS (quasi-Netwon) method from R function `optim`. This algorithm belongs to the class of quasi-Netwon methods because in each step the Hessian matrix is approximated and updated using information from previous iteration. This assumption avoids to perform expensive computation at each iteration. We input the so-obtained optimal values as initial point for the Metropolis-Hastings and the rescaled Hessian matrix of the loss function as the covariance matrix of the Gaussian proposal distribution.

This type of representation allows for posterior inference on model parameters and implicitly to perform structure learning of the underlying UG according to (3.4). Through the implementation of a suitable MCMC scheme, it is possible to approximate the posterior distribution of each interaction term-parameter, whose posterior mean provides a point estimate of the corresponding term, unless the corresponding edge is set to zero through structure learning procedure. Moreover, structure learning is performed building a credible set on the posterior distribution of each interaction term. For one-dimensional parameters θ, a credible interval of level α is the symmetric interval $I_\alpha(\theta)$ which contains a proportion $1 - \alpha$ of the probability mass of the posterior distribution. Given that the presence of an edge corresponds to a non-zero value of the corresponding parameter, the adjacency matrix $\boldsymbol{A}$ of graph $\mathcal{G}$ can be estimated by excluding edges whenever the associated credible interval includes the zero value. More in details, through the MCMC we recover $\hat{I}_\alpha(\lambda_{jk})$, $\hat{I}_\alpha(\Phi_{\varnothing,jk})$ and $\hat{I}_\alpha(\eta_{jk})$ given by the empirical quantiles. Then for $j \in \Delta$ and $k \in \Delta$, we have $\hat{A}_{jk} = 1 - \mathbf{1}\left(\hat{I}_\alpha(\lambda_{jk}) \ni 0\right)$ for those edges expressing dependencies between discrete variable; for $j \in \Gamma$ and $k \in \Gamma$, we have $\hat{A}_{jk} = 1 - \mathbf{1}\left(\hat{I}_\alpha(\Phi_{\varnothing,jk}) \ni 0\right)$ for edges between

continuous variables, and otherwise $\hat{A}_{jk} = 1 - \mathbf{1}\left(\hat{I}_\alpha(\eta_{jk}) \ni 0\right)$ for mixed dependencies; compare also with Equation (3.4).

Let the 0-superscript denote a true unknown quantity, for instance $\boldsymbol{\theta}^0$ is the true unknown parameter vector $\boldsymbol{\theta}$, λ_{jk}^0 is the true unknown value of λ_{jk}. Also, call $\omega_s(\boldsymbol{\theta})$ the exponential function argument in λ_0, so that $\lambda_0 = q\log(2\pi)/2 - \log|\boldsymbol{\Phi}_\varnothing|/2 - \log\sum_{\boldsymbol{s}\in\mathcal{Z}}\exp\{\omega_s(\boldsymbol{\theta})\}$, where, for any configuration $s = (s_1 s_2 \cdots s_p)$ and with $s_0 := 1$, we have defined

$$\omega_s(\boldsymbol{\theta}) := \sum_{j=1}^{p}\lambda_j s_j + \sum_{k<j}\lambda_{jk}s_j s_k + \frac{1}{2}\left(\sum_{j=0}^{p}\boldsymbol{\eta}_{\boldsymbol{j}}^{\top}s_j\right)\boldsymbol{\Phi}_\varnothing\left(\sum_{j=0}^{p}\boldsymbol{\eta}_{\boldsymbol{j}}s_j\right),$$

and finally $w_s(\boldsymbol{\theta}) := \exp\{\omega_s(\boldsymbol{\theta})\}/\sum_{s\in\mathcal{Z}}\exp\{\omega_s(\boldsymbol{\theta})\}$. We next show that the estimation of all parameters is asymptotically correct, and that therefore the correct dependency structure will be recovered for samples of size large enough.

Proposition 3.3.1. *Given knowledge of all other parameters and a square loss function, we have, for the Bayes estimator $\hat{\lambda}_{jk}$, $j,k \in \Delta$, that*

$$\sqrt{n}\left(\hat{\lambda}_{jk} - \lambda_{jk}^0\right) \xrightarrow{d} N\left(0, \frac{1}{\gamma_{jk}(\boldsymbol{\theta}^0)(1-\gamma_{jk}(\boldsymbol{\theta}^0))}\right),$$

where $\gamma_{jk}(\boldsymbol{\theta}) = \sum_{s\in\mathcal{Z}} w_s(\boldsymbol{\theta})s_j s_k$.

Proof. Because of $f(\boldsymbol{X}|\boldsymbol{\theta})$ belonging to the exponential family, together with a prior on λ_{jk} which is continuous over $\mathbb{R}$ and with finite expectation, the Bernstein Von-Mises theorem applies; see for instance Theorems 1.4.2 and 1.4.3 of Ghosh & Ramamoorthi [40]. Then $\hat{\lambda}_{jk} \approx N\left(\lambda_{jk}^0, 1/I_n(\lambda_{jk}^0)\right)$, where $I_n(\lambda_{jk}) = -\mathbb{E}\partial^2 \log f(\boldsymbol{X}|\boldsymbol{\theta})/\partial\lambda_{jk}^2$. Now, see that

$$I_n(\lambda_{jk}) = -n\mathbb{E}\frac{\partial^2}{\partial\lambda_{jk}^2}\lambda_\varnothing = n\left[\frac{\frac{\partial^2}{\partial\lambda_{jk}^2}\sum_{s\in\mathcal{Z}}\exp\{\omega_s(\boldsymbol{\theta})\}}{\sum_{s\in\mathcal{Z}}\exp\{\omega_s(\boldsymbol{\theta})\}} - \left(\frac{\frac{\partial}{\partial\lambda_{jk}}\sum_{s\in\mathcal{Z}}\exp\{\omega_s(\boldsymbol{\theta})\}}{\sum_{s\in\mathcal{Z}}\exp\{\omega_s(\boldsymbol{\theta})\}}\right)^2\right],$$

and since

$$\frac{\partial}{\partial\lambda_{jk}}\sum_{s\in\mathcal{Z}}\exp\{\omega_s(\boldsymbol{\theta})\} = \frac{\partial^2}{\partial\lambda_{jk}^2}\sum_{s\in\mathcal{Z}}\exp\{\omega_s(\boldsymbol{\theta})\} = \gamma_{jk}\sum_{s\in\mathcal{Z}}\exp\{\omega_s(\boldsymbol{\theta})\},$$

we obtain $I_n(\lambda_{jk}) = n\gamma_{jk}(1-\gamma_{jk})$. □

Remark 3.3.1. *No further assumptions on the existence and asymptotic distribution of the maximum likelihood estimator for $\boldsymbol{\theta}$ is required, since the model is in the exponential family. For the same reason, no assumptions are needed on the asymptotic behaviour of the likelihood ratio; see for instance Example 7.7 of van der Vaart [99].*

The above result focuses on those Bayesian estimators related to the the dependencies among discrete components, and show that they are asymptotically Gaussian, centered on the correct unknow parameter value, and with a vanishing variance. We next show an analogous result for those Bayesian estimators that capture the mixed conditional dependecies, among continuos and discrete coordinates.

Proposition 3.3.2. *Under the same conditions of Proposition 3.3.1, we have, for the Bayes estimator $\hat{\boldsymbol{\eta}}_j$, $j = 0, \dots, p$, that*

$$\sqrt{n}\left(\hat{\boldsymbol{\eta}}_j - \boldsymbol{\eta}_j^0\right) \xrightarrow{d} N\left(\mathbf{0}, \frac{1}{2}(\boldsymbol{\Phi}_\varnothing^0)^{-1}\left(\gamma_j(\boldsymbol{\theta}^0)\boldsymbol{I} + 2\boldsymbol{\Phi}_\varnothing^0\left(\boldsymbol{\Psi}_j(\boldsymbol{\theta}^0) - \boldsymbol{\Lambda}_j(\boldsymbol{\theta}^0)\boldsymbol{\Lambda}_j(\boldsymbol{\theta}^0)^\top\right)\right)^{-1}\right),$$

where $\gamma_j(\boldsymbol{\theta}) = \sum_{s\in\mathcal{Z}} w_s(\boldsymbol{\theta}) s_j$, $\boldsymbol{\Lambda}_j(\boldsymbol{\theta}) = \sum_{s\in\mathcal{Z}}\sum_{k=0}^p w_s(\boldsymbol{\theta}) s_j s_k \boldsymbol{\eta}_k$, and where $\boldsymbol{\Psi}_j(\boldsymbol{\theta}) = \sum_s \sum_{k,l} w_s(\boldsymbol{\theta}) s_j s_k s_l \boldsymbol{\eta}_k \boldsymbol{\eta}_k^\top$.

Proof. We follow the same line of reasoning of Proposition 3.3.1, with $I_n(\boldsymbol{\eta}_j) = -n\frac{\partial^2}{\partial\boldsymbol{\eta}_j\partial\boldsymbol{\eta}_j^\top}\lambda_\varnothing$. Note that

$$\frac{\partial}{\partial\boldsymbol{\eta}_j}\exp\{\omega_s(\boldsymbol{\theta})\} = 2\boldsymbol{\Phi}_\varnothing \exp\{\omega_s(\boldsymbol{\theta})\}\sum_{k=0}^p \boldsymbol{\eta}_k s_k s_j,$$

where $s_0 := 1$, and that

$$\begin{aligned}\frac{\partial^2}{\partial\boldsymbol{\eta}_j\partial\boldsymbol{\eta}_j^\top}\exp\{\omega_s(\boldsymbol{\theta})\} &= 2\boldsymbol{\Phi}_\varnothing\frac{\partial}{\partial\boldsymbol{\eta}_j^\top}\left[\exp\{\omega_s(\boldsymbol{\theta})\}\sum_{k\neq j}\boldsymbol{\eta}_k s_k s_j + \exp\{\omega_s(\boldsymbol{\theta})\}\boldsymbol{\eta}_j s_j\right] \\ &= 2\boldsymbol{\Phi}_\varnothing\exp\{\omega_s(\boldsymbol{\theta})\}\left[2\boldsymbol{\Phi}_\varnothing\sum_{k,l}\boldsymbol{\eta}_k\boldsymbol{\eta}_l^\top s_k s_l s_j + s_j\boldsymbol{I}_p\right].\end{aligned}$$

Therefore we can write

$$\begin{aligned}I_n(\boldsymbol{\eta}_j) &= n\frac{\partial^2}{\partial\boldsymbol{\eta}_j\partial\boldsymbol{\eta}_j^\top}\log\sum_{\boldsymbol{s}\in\mathcal{Z}}\exp\{\omega_s(\boldsymbol{\theta})\} \\ &= 2n\boldsymbol{\Phi}_\varnothing\left[\gamma_j(\boldsymbol{\theta})\boldsymbol{I}_p + 2\boldsymbol{\Phi}_\varnothing\left(\boldsymbol{\Psi}_j(\boldsymbol{\theta}) - \boldsymbol{\Lambda}_j(\boldsymbol{\theta})\boldsymbol{\Lambda}_j(\boldsymbol{\theta})^\top\right)\right].\end{aligned}$$

□

Remark 3.3.2. *The assumption of knowledge of all other parameters is to simplify the derivations of the asymptotic variances. We can remove this assumption, and compute $\boldsymbol{I}_n(\boldsymbol{\theta})$ as a function of the whole parameter vector. Then the parameter subvector of interest, say $\boldsymbol{\eta}_j$, will have an asymptotic covariance matrix equal to $(\boldsymbol{I}_n(\boldsymbol{\theta})^{-1})_{\boldsymbol{\eta}_j}$, that is the submatrix of $\boldsymbol{I}_n(\boldsymbol{\theta})^{-1}$ with rows and columns corresponding to $\boldsymbol{\eta}_j$.*

We finally provide a similar result for dependencies among the nodes representing continuous coordinates $\mathbf{y}$. Since the related parameter $\boldsymbol{\Phi}$ is a symmetric positive matrix, the result is stated in terms of the half vectorization of $\boldsymbol{\Phi}$, that stacks in a vector the low-triangular part of $\boldsymbol{\Phi}$, main diagonal included.

Proposition 3.3.3. *Under the same conditions of Proposition 3.3.1, we have, for the Bayes estimator $\hat{\phi}$ of $\phi^0 = vech\,(\Phi^0_{\varnothing})$, the half-vectorization of $\Phi^0_{\varnothing}$, that*

$$\sqrt{n}\left(\hat{\phi} - \phi^0\right) \xrightarrow{d} N\left(\mathbf{0}, \left(\frac{1}{2}\mathrm{tr}\left(\Phi^0_{\varnothing}\right)^{-2} vech\,(\boldsymbol{I}_p)\, vech\,(\boldsymbol{I}_p)^\top + \boldsymbol{\Pi}(\boldsymbol{\theta}^0) - \boldsymbol{\Gamma}(\boldsymbol{\theta}^0)\boldsymbol{\Gamma}(\boldsymbol{\theta}^0)^\top\right)^{-1}\right),$$

where $\boldsymbol{\Gamma}(\boldsymbol{\theta}) = \sum_{s\in\mathcal{Z}} w_s(\boldsymbol{\theta})\boldsymbol{\tau}_s(\boldsymbol{\theta})$ and $\boldsymbol{\Pi}(\boldsymbol{\theta}) = \sum_{s\in\mathcal{Z}} w_s(\boldsymbol{\theta})\boldsymbol{\tau}_s(\boldsymbol{\theta})\boldsymbol{\tau}_s(\boldsymbol{\theta})^\top$, with $\boldsymbol{\tau}_s(\boldsymbol{\theta}) = vech\left[\left(\sum_{j=0}^p s_j\boldsymbol{\eta}_j\right)\left(\sum_{j=0}^p s_j\boldsymbol{\eta}_j\right)^\top\right]$.

Proof. We follow the same reasoning of Proposition 3.3.3, and we compute

$$\boldsymbol{I}_n(\phi) = \frac{\partial^2}{\partial\boldsymbol{\phi}_j\partial\boldsymbol{\phi}_j^\top}\left(n\log\sum_{\boldsymbol{s}\in\mathcal{Z}}\exp\{\omega_s(\boldsymbol{\theta})\} + \frac{n}{2}\log|\boldsymbol{\Phi}_{\varnothing}| + \frac{1}{2}\mathrm{tr}\left(\boldsymbol{R}\boldsymbol{\Phi}_{\varnothing}\right)\right),$$

where the third term is null, the second term is $\frac{n}{2}\mathrm{tr}\left(\Phi^0_{\varnothing}\right)^{-2}\mathrm{vech}\left(\boldsymbol{I}_p\right)\mathrm{vech}\left(\boldsymbol{I}_p\right)^\top$, and the first term is

$$n\left[\frac{\frac{\partial^2}{\partial\boldsymbol{\phi}_j\partial\boldsymbol{\phi}_j^\top}\sum_{\boldsymbol{s}\in\mathcal{Z}}\exp\{\omega_s(\boldsymbol{\theta})\}}{\sum_{\boldsymbol{s}\in\mathcal{Z}}\exp\{\omega_s(\boldsymbol{\theta})\}} - \frac{\frac{\partial}{\partial\boldsymbol{\phi}_j}\sum_{\boldsymbol{s}\in\mathcal{Z}}\exp\{\omega_s(\boldsymbol{\theta})\}\cdot\frac{\partial}{\partial\boldsymbol{\phi}_j^\top}\sum_{\boldsymbol{s}\in\mathcal{Z}}\exp\{\omega_s(\boldsymbol{\theta})\}}{\left(\sum_{\boldsymbol{s}\in\mathcal{Z}}\exp\{\omega_s(\boldsymbol{\theta})\}\right)^2}\right]$$
$$= n\left[\boldsymbol{\Pi}(\boldsymbol{\theta}) - \boldsymbol{\Gamma}(\boldsymbol{\theta})\boldsymbol{\Gamma}(\boldsymbol{\theta})^\top\right].$$

□

3.4 Simulation results

We compare the proposed methodology with the approach described in the work of Cheng et al. [14]. Since both models are based on the CG distribution, the comparison is convenient to test our extension to a Bayesian framework. Additionally, the Authors showed in the paper a good performance of their methodology with respect to other state-of-the art approaches. Thus, we are implicitly assessing our proposal to other alternatives.

In what follows, we name HMGM the method of Cheng et al. [14] and BGM-MD our proposal.

To compare these two approaches, we define a simulation setting as follows:

- Total number of variables: $\{5, 10, 20\}$ of which the discrete variables are respectively $\{2, 5, 10\}$
- Total number of observations: $\{500, 1000, 2000\}$

Specifically, we consider all the combination of these two dimensions, having nine possible scenarios. For each setting the total number of simulations is 40 and each time the true setting (graph and parameters) is different.

The true adjacency matrix is randomly generated using a probability of inclusion equal to $\frac{3}{2(p+q-1)}$, instead true parameters and data are generated using the R package `hmgm` provided by the authors of [14]. Additionally, as suggested by the authors, HMGM is replicated 40 times on random subsets of data (90% of total) within each simulation. From each run both the parameters and graph estimate are kept. The final parameter estimates are obtained by averaging all the partial results and the graph structure estimate is done by including edges selected at least 90% of times.

As explained in the previous section, we use an MCMC strategy to perform inference. For this simulation, we run the algorithm for 50000 iterations, with a burn-in period of 10000 and a thinning parameter equal to 10. The burn-in period allows to discard part of initial samples when the chains are not stationary, and with a thinning parameter equal to 10 we keep every 10th value, avoiding auto-correlations in the chains. Before the computation of credible intervals, it is fundamental to check the convergence of the MCMC chains. This step has been performed using the R package `coda` [86] which provides diagnostic functions to test for convergence (e.g., Geweke's test), cross-correlations and auto-correlations.

3.4.1 Simulation plots and tables

In this subsection, we report the results of the simulation study through which we compare HMGM and BGM-MD. Firstly, we evaluate both methods effectiveness in recovering the true structure of the underlying graph. The performance is assessed comparing the estimated graph to the corresponding true graph. Figure 3.1 shows the distribution over the simulation replicates of Structural Hamming Distance (SHD) which represents the number of edge insertions, deletions or flips needed to transform the estimated graph into the true one (lower values of SHD correspond to better performance). We see that BGM-MD outperforms the alternative methodology in all settings. Moreover, the difference increases accordingly to the number of variables considered and it is possible to notice that for BGM-MD sample size has a positive effect on structure learning. We report in Tables 3.1-3.3 a summary of the comparison using other performance indicators such as False Positive Rate (FPR), Misspecification Rate (MISR), specificity (SPE), sensitivity (SEN), precision (PRE) and Matthews correlation coefficient (MCC), defined as

$$FPR = \frac{FP}{TN + FP}, \quad MISR = \frac{FN + FP}{|V|\big(|V| - 1\big)},$$
$$SPE = \frac{TN}{TN + FP}, \quad SEN = \frac{TP}{TP + FN}, \quad PRE = \frac{TP}{TP + FP},$$
$$MCC = \frac{TP \cdot TN - FP \cdot FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}}$$

where TP, TN, FP, FN are the numbers of true positives, true negatives, false positives and false negatives (respectively) and $|V|$ is the total number of variables.

Tables 3.1-3.3 report the mean and standard deviation of each indicator summarizing over the simulation replicates and scenarios. For all indicators, except for FPR and MISR, higher values correspond to better performance. It appears that BGM-MD generally performs better the task of structure learning, increasing its effectiveness with larger number of observations. According to these indicators, it appears that HMGM has always a significant higher value of FPR with respect to BGM-MD. This means that our approach estimate more sparse graphs and it is more efficient in including the right edges.

Secondly, to quantify the error of parameters vector estimate we use Root Mean Square Error (RMSE) defined as

$$RMSE = \frac{1}{|\boldsymbol{\theta}^0|} \sum_{i=1}^{|\boldsymbol{\theta}^0|} (\hat{\boldsymbol{\theta}}_i - \boldsymbol{\theta}_i^0)^2$$

where $\boldsymbol{\theta}^0$ is the vector of true parameters of size $|\boldsymbol{\theta}^0|$ and $\hat{\boldsymbol{\theta}}$ is the vector of estimated parameters accordingly to the description in Section 3.3. Figure 3.2 shows RMSE distributions for the simulation settings according to number of variables and observations. It is possible to notice that generally BGM-MD performs better and the error reduces as n increases, as expected.

In these simulations we set credible sets at 90% in order to compare with edge selection approach used by Cheng et al. In our approach the choice of the level α may influence the results of structure learning and we observe that our performance increases using smaller values of α (e.g., 0.01). We expect this situation because smaller values of α correspond to wider intervals. Therefore, posterior distributions that are centered on a small value are more likely to include the zero in the credible set as the level decreases. Additional simulated scenarios at different levels of alpha are currently under investigation.

3.5 Application to real datasets

We now apply our methodology to three real datasets, and in comparison with HMGM. The idea is to show how structure learning can address different tasks in diverse applications.

For each dataset, specific MCMC parameters (number of iterations, burn-in period, thinning) are fixed. In particular, the burn-in period allows to discard some initial samples when the chains are not stationary, and with a thinning parameter equal to w we keep every w^{th} value, discarding all other values and avoiding auto-correlations in the chains. MCMC tuning is performed according to the results returned by `coda` [86] R package

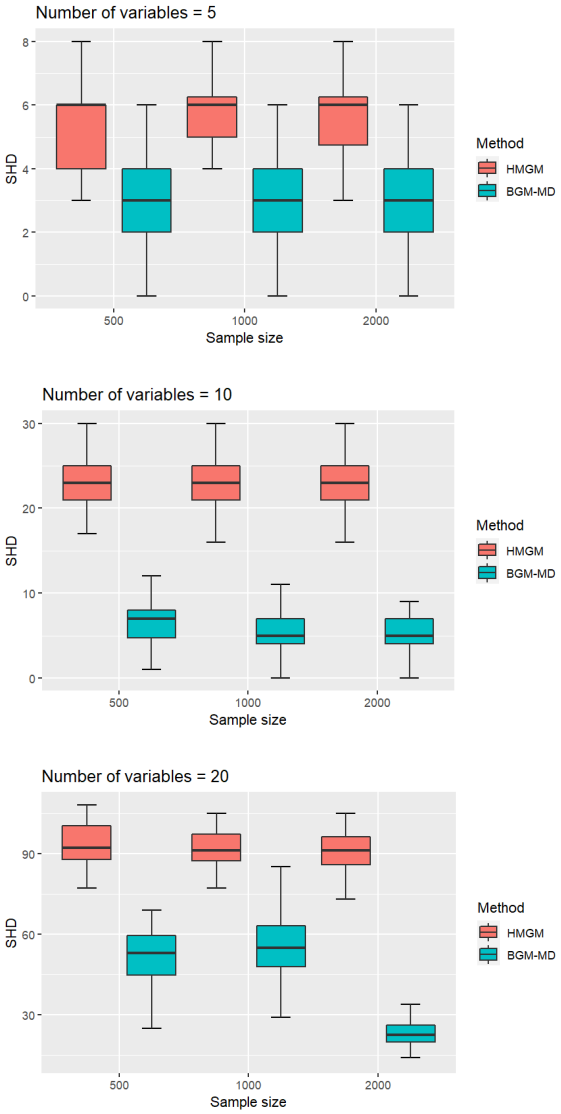


Figure 3.1: *Performance results for structure learning: Structural Hamming Distance (SHD)*

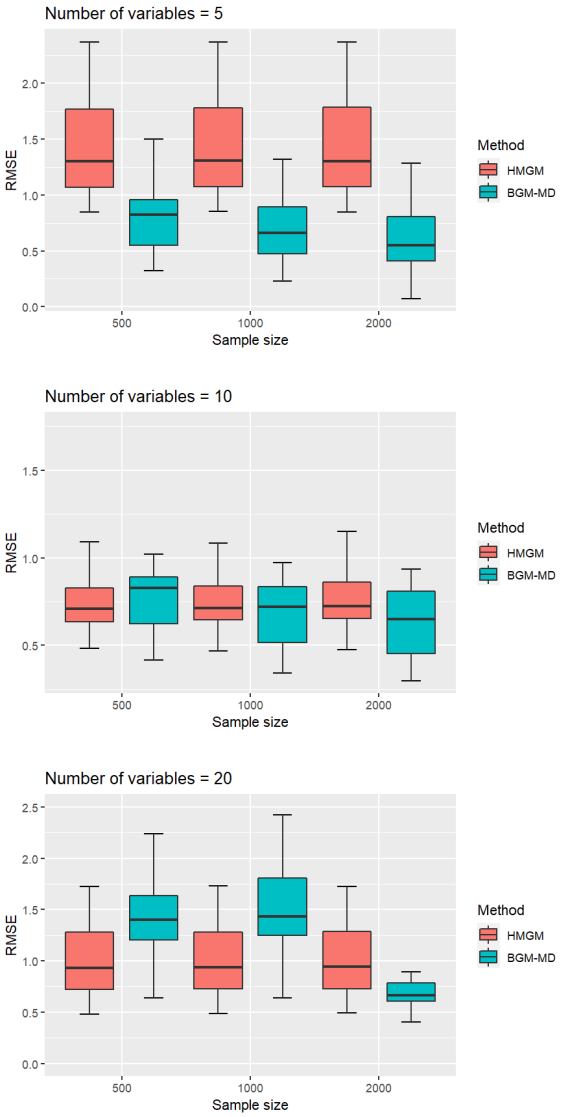


Figure 3.2: *Performance results for parameters inference: Root Mean Square Error (RMSE)*

diagnostic tests for convergence (e.g., Geweke's test), stationarity, autocorrelations and cross-correlations. For brevity, only some of these analysis for the different applications are shown, as an example. All algorithms are preliminary initialized through an optimization step, using the approach presented in the previous sections. As in the simulation studies, credible sets are at 90%, and HMGM selects edges appearing at least 90% of 40 repetitions.

3.5.1 Nanostructure dataset

In the nanotechnology industry, the focus is on functional structures designed at the atomic or molecular scale, related to optoelectronics, luminescent materials, lasing materials and biomedical imaging [4, 50, 72]. Of particular interest is the Cadmium Selenide material, found to exhibit one-dimensional morphologies (nanostructures) of nanowires, nanobelts, and nanosaws [73], often with the three morphologies closely related. We use the dataset of [22], where the Authors proposed a generalized linear model to analyse those experimental nanostructures.

From a statistical perspective, nanostructure data can be represented as categorical variables, corresponding to the dominant nanostructure type, or as binary variables, expressing absence/presence of a structure type. But, the presence of each nanostructure is a function of continuous noisy features, themselves random variables. These predictors are the three key process variables affecting the morphology of the nanostructures: temperature, pressure and the distance from the source material of the substrate where the deposition of nanostructures is collected. Process variables are typically set on specific (nominal) values, however there are unavoidable fluctuations associated with the synthesis process that allows to consider them as stochastic. In the data only nominal values are available, therefore we add some noise. In this setting, our methodology can therefore provide a rigorous framework to analyse the dependence among the three nanostructures, learn how process variables can affect nanostructure occurrence probabilities, and find the experimental setting leading to maximum probability of a given morphology.

In the data there are four levels of temperature (630, 700, 750, 800°C) and nine levels of pressure (4, 100, 200, 300, 400, 500, 600, 700, 800 mbar). For each setting, several substrates were placed near the source, to observe the deposition of nanostructures, and this distance is measured from its midpoint to the source. After each experiment, through electron microscopy, the number of appearance of different nanostructures is counted. Further details on the experimental design can be found in [73]. In the pre-processing step, a comparatively small random noise is added to temperature and pressure using the information about their mean and variance included in [22]. Then, these variables are Box-Cox transformed to assure Gaussianity; see, for example, Figure 3.3. Also, nanostructure

counts are converted into binary variables on the presence/absence of each specific structure. Since temperature has a key role in the development in nanostructures, we conduct two separate analyses: (i) we also transform the temperature into a binary variable, using a threshold of 750°C, or (ii) we separately analyze the data for each level of temperature. The choice of the threshold for the temperature is related to some considerations described in [73]. According to diagnostic test conducted on the output, settings and MCMC tuning for both applications are summarized in Table 3.4. As an example, we report in Figure 3.4 the auto-correlation plots for Case 1: we can observe that auto-correlations decrease significantly after a small number of lags.

Figures 3.5 and 3.6 show the estimated graphs in analyses (i) and (ii) respectively, whilst Figure 3.7 shows the credible set at 90% of the interaction parameters λ and η. From these boxplots, we can notice how mixed interactions change at different level of temperature. In both analyses HMGM estimates more complex structures, as expected from the simulation results where FPR is sensibly higher than the Bayesian approach. Especially, in Case 2 HMGM does not clearly evidence different relations according to the levels of temperature, as showed by BGM-MD estimates. From this graphical representation of conditional independencies, it emerges that nanostructures are intermingled and controlled mainly by the value of pressure, as evidenced in [73]. We recognize it from Figure 3.6 showing that pressure has a connection with a particular type of nanostructure depending on the temperature value. Additionally, using parameters estimates it is possible to compute all the probabilities for nanostructures configurations finding the optimal process conditions as performed by [22].

3.5.2 Hepatitis dataset

The second study refers to data collected on patients affected by acute and chronic hepatitis disease, publicly available at `https://archive.ics.uci.edu/`. The original dataset contains 155 observations and 19 continuous and binary variables, such as patient information, symptoms and standard biochemical measurements. The attribute (*Class*) indicates if the patient has survived or not the disease. According to particular attention received in previous works to some variables [12, 24, 90], we selected a subset of them which are relevant related to *Class*. Thus, the dataset for our study includes 11 variables divided into:

- Gaussian: *Age*, *Bilirubin*, *Albumine*, *Protime*
- Binary: *Class*, *Sex*, *SpleenPalpable*, *Spiders*, *Histology*, *AscitesVarices*, *FatigueMalaise*.

Continuous variables are transformed using a Box-Cox transformation to guarantee Gaussianity; see Figures 3.8 and 3.9, as an example. Within

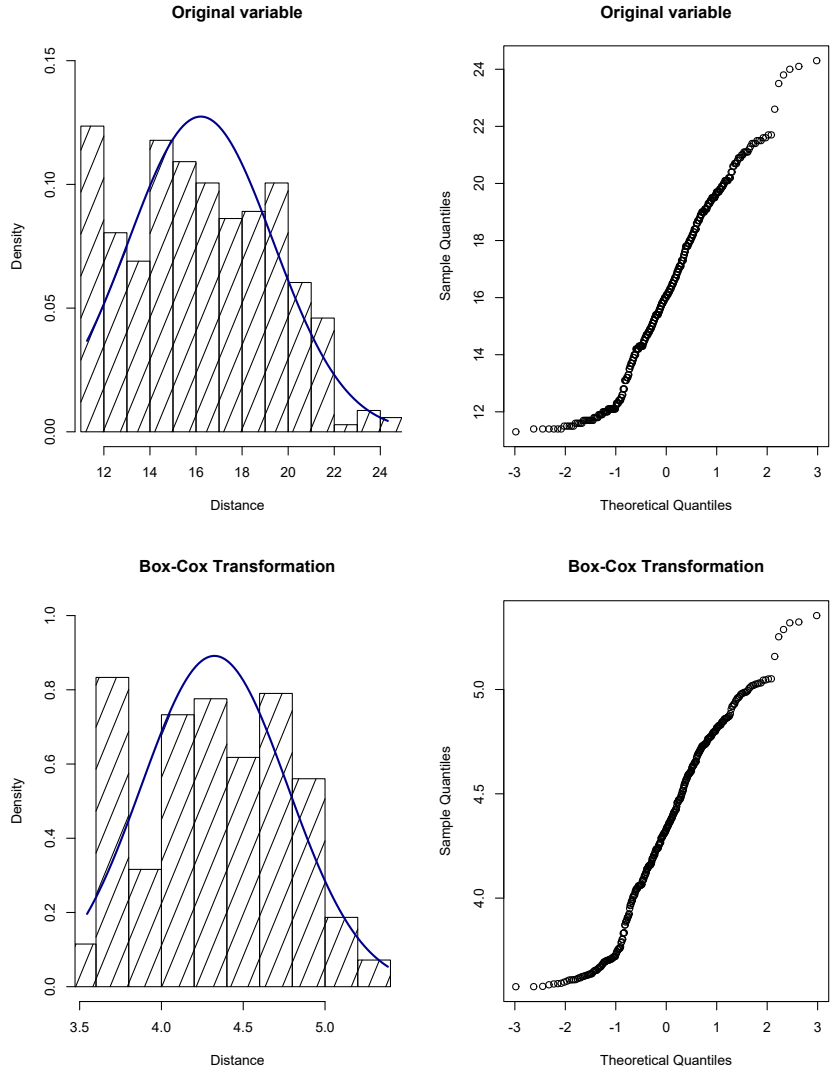


Figure 3.3: *Box-Cox transformation of* Distance *variable in nanostructure dataset*

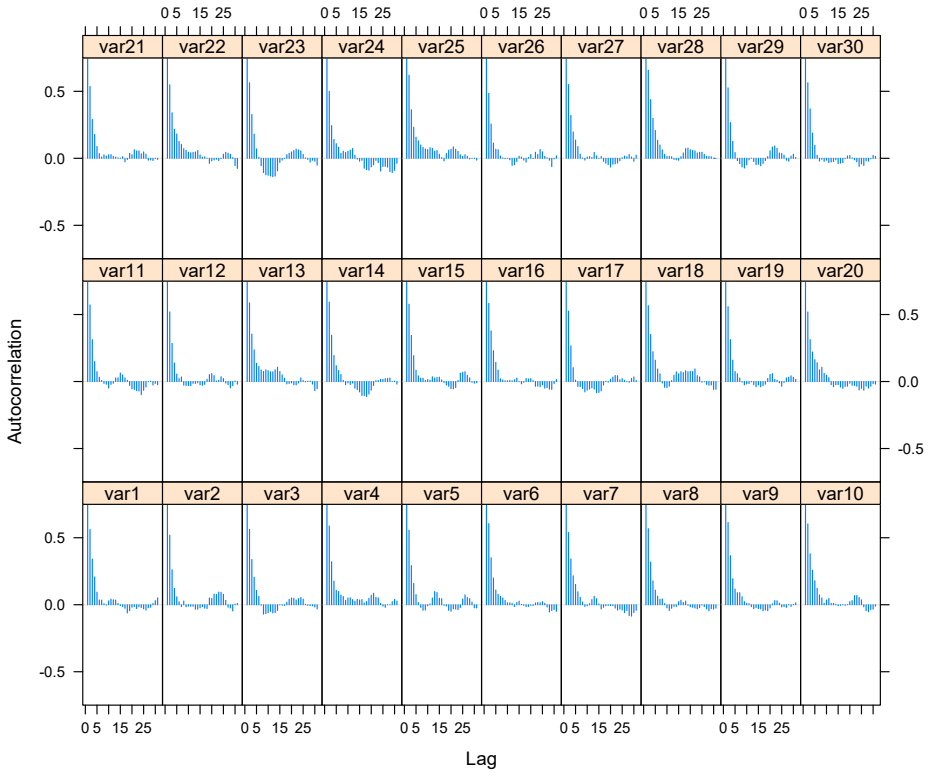


Figure 3.4: *Auto-correlation plots for all estimated parameters in Case 1 (see 3.4) of nanostructure application*

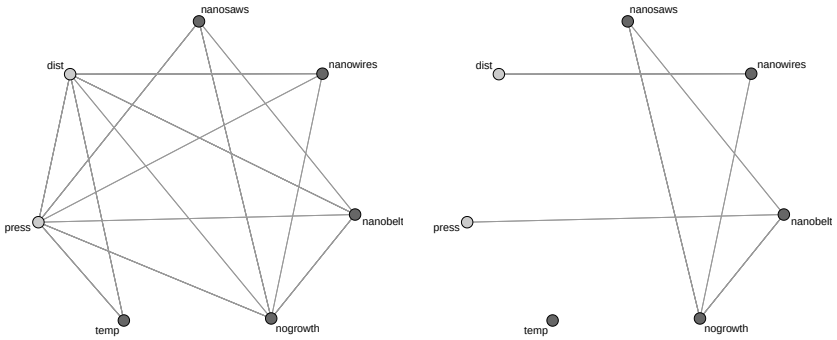


Figure 3.5: *Estimated graph structure in Case 1 (see 3.4). On the left HMGM, on the right BGM-MD*

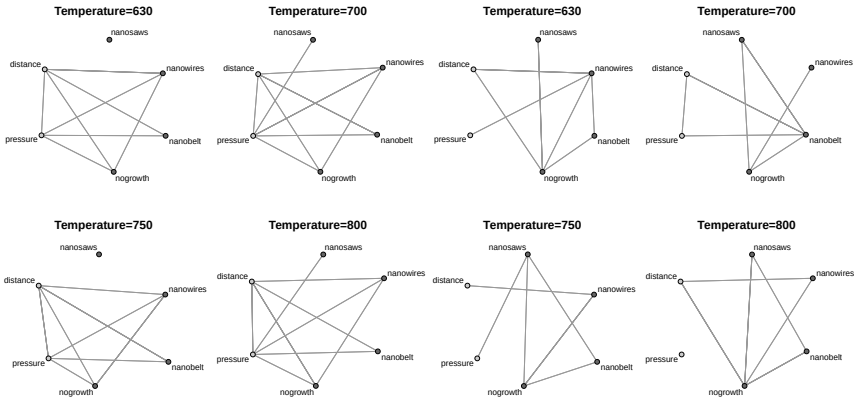


Figure 3.6: *Estimated graph structure in Case 2 (see Table 3.4). On the left HMGM, on the right BGM-MD*

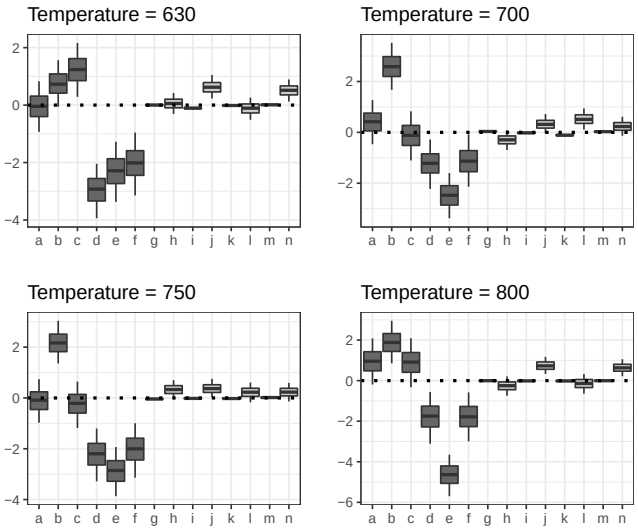


Figure 3.7: *Credible sets at 90% of estimated interaction parameters (λ and η) in Case 2 (see Table 3.4).*

the set of binary variables, *AscitesVarices* and *FatigueMalaise* were originally four separate variables *Ascites, Varices, Fatigue* and *Malaise*. We encoded them into these new paired variables assigning 1 when both symptoms are present and 0 otherwise. Given this simultaneous presence in patients of pairs of symptoms, this strategy helps to avoid multicollinearity in the variables and reduce the number of parameters eliminating redundancy in the information.

Our framework will provide insights about the dependence structure among symptoms and biochemical measures, learning which variables affect the probability of survival, for a patient with a diagnosis of acute hepatitis. In this analysis, the MCMC algorithm is set with 100000 iterations, with a burn in period of 30000 and a thinning parameter equal to 100. In Table 3.5 we report some diagnostic results, especially for the mixed interactions. The first two columns show the estimate of each parameter and the corresponding standard error. Additionally, Geweke's test for stationary distributions shows results of convergence with small values of the statistic test. Then, values of auto-correlations at different lags are reported in the last three columns. The results show a convergence of the algorithm with good mixing behavior, as displayed in the cross-correlation plot in Figure 3.10. In Figure 3.11 we report the resulting graph of the BGM-MD, compared to the outcome of HMGM method. It is possible to notice that graph estimated by HMGM has a large number of edges, especially the outcome variable *Class* is connected to all other variables. Instead, BGM-MD estimates a more sparse graph highlighting a smaller set of depedencies. From the results, it appears that the probability of survival is mainly related to the

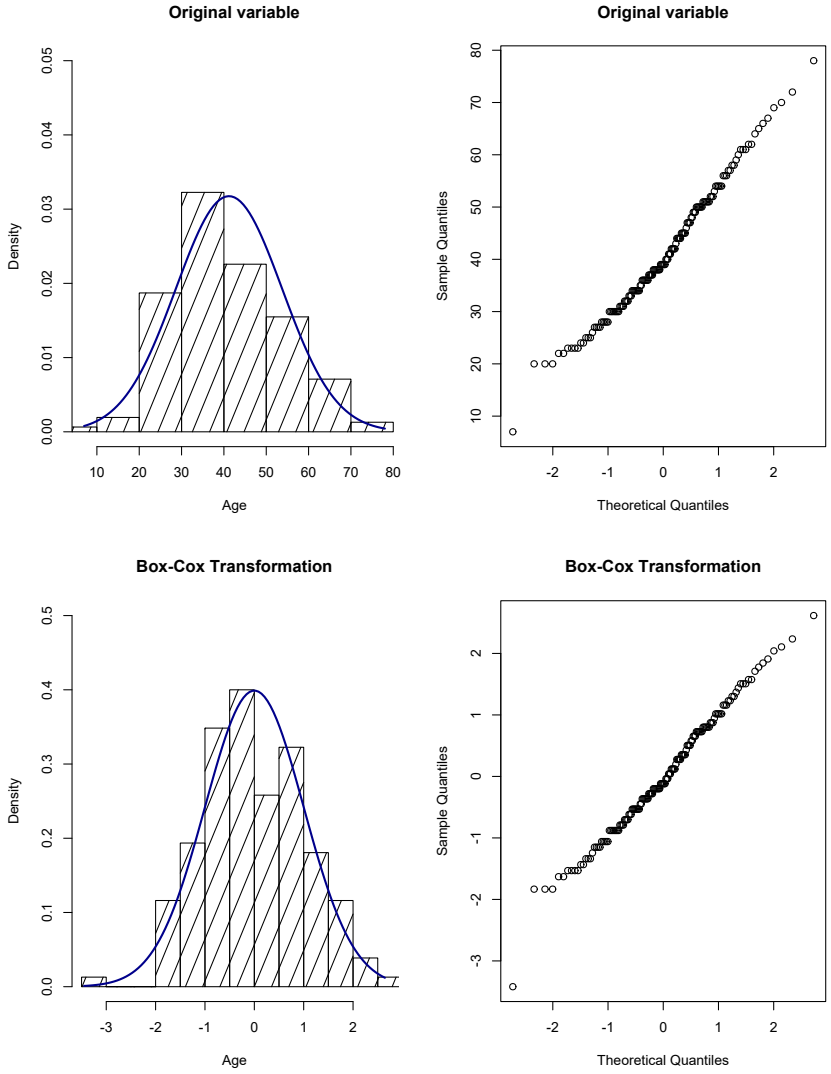


Figure 3.8: *Box-Cox transformation of* Age *variable in hepatitis dataset*

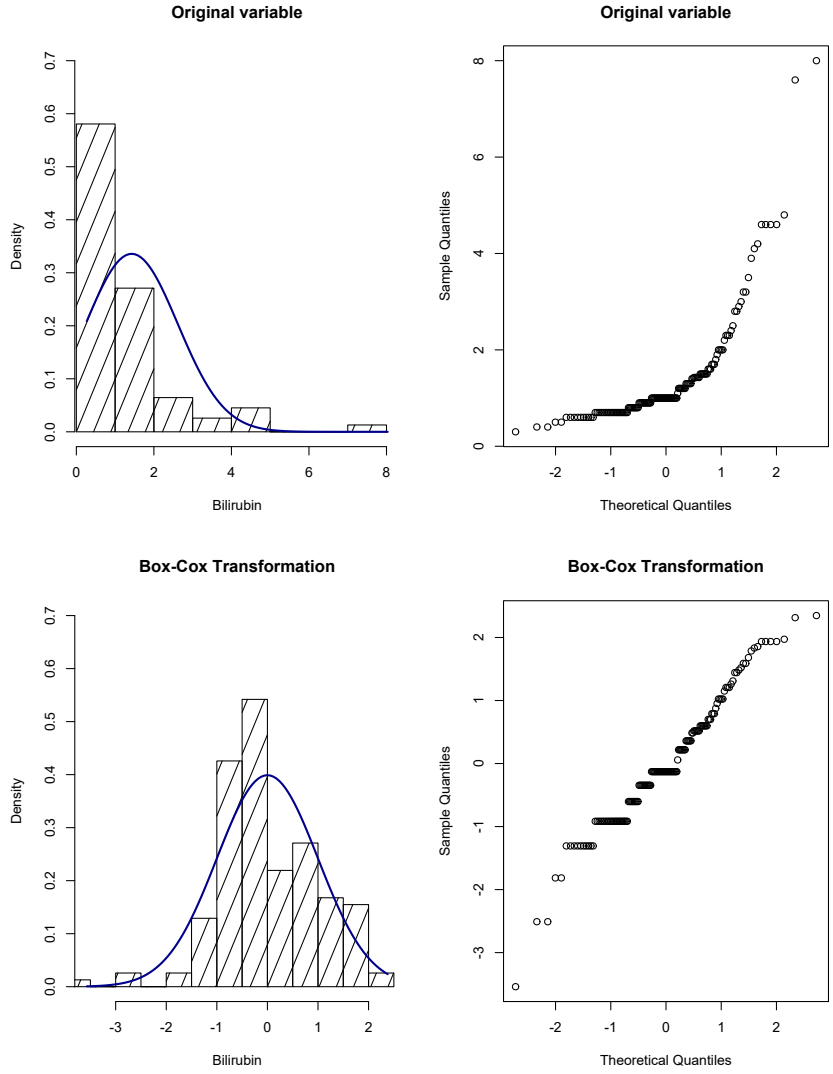


Figure 3.9: *Box-Cox transformation of* Bilirubin *variable in hepatitis dataset*

measurements of *Bilirubin* and *Albumine* together with the symptoms of *AscitesVarices* and *FatigueMalaise*. Additionally, it is possible to consider some indirect factors of influence such as *Age* and *Histology*.

3.5.3 Heart-disease patients dataset

The third analysis consists of an application of the proposed methodology to a dataset relative to heart-disease patients collected from the *Cleveland Clinic Foundation* and publicly available at `https://archive.ics.uci.edu/`. The original dataset contains $n = 303$ observations and 76 variables. We include in our study 12 variables, which also received particular attention in several previous analyses (see dataset information). To ease the implementation of our methodology, categorical variables with a number of levels larger than two were also converted into binary variables. Moreover, within the categories of possible symptoms, the majority of observations concentrate in one specific type. Thus, each of the so-obtained binary categorical variables indicates the absence/presence of a symptom (e.g., “chest pain type"), patient's characteristic (e.g., “fasting blood sugar") or disease (e.g., “diagnosis of heart disease"). Variables included in the analysis are then divided into:

- Gaussian: *age*, resting blood pressure (*trestbps*), serum cholesterol (*chol*), maximum heart rate achieved (*thalach*);
- Binary: *sex*, fasting blood sugar (*fbs*), exercise induced angina (*exang*), chest pain type (*cp*), diagnosis of heart disease (*num*), resting electrocardiograph results (*restecg*), slope of the peak exercise ST segment (*slope*), type of test (*thal*).

We run the MCMC algorithm for 50000 iterations, with a burn-in period of 10000 and a thinning parameter equal to 10.

We first report in Figure 3.12a the resulting graph estimate. As described in Section 3.3, our method for structure learning is based on the computation of a credible interval (here at the 90% level) for each of the model parameters. Figure 3.12b summarizes the credible intervals and quantiles of parameters involving variables that were estimated to be connected with "diagnosis of heart disease" (*num*). All of them do not include the zero value in their 90% credible sets, and accordingly the corresponding edges are included in the graph estimate of Figure 3.12a. Similar results, that we do not include for brevity, were obtained for the other parameters.

3.6 Moment representation of CG distribution

As mentioned in Section 3.2.1, it is possible to use an alternative parametrization for the CG distribution. This representation is more suitable for posterior inference on parameters instead of a task of structure learning. In this

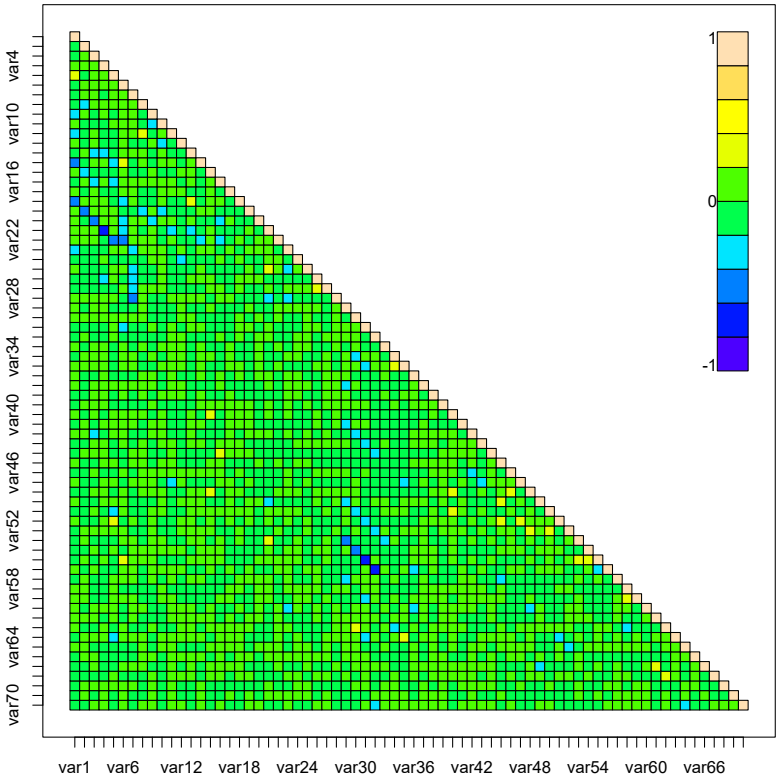


Figure 3.10: *Cross-correlation output for hepatitis dataset: each variable identifies a parameter in the model*

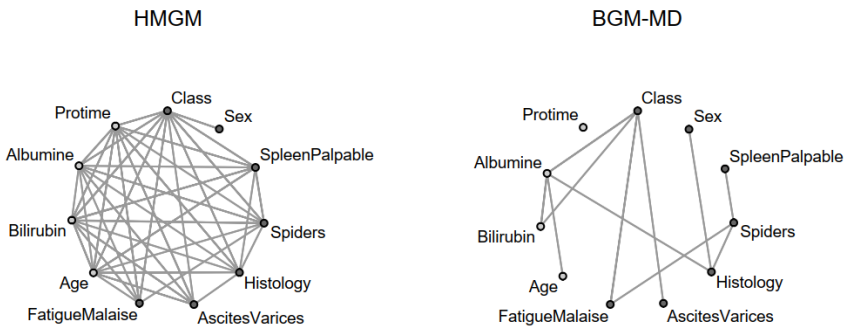


Figure 3.11: *Estimated graph structure. On the left HMGM, on the right BGM-MD (see Table 3.5.2)*

Section, this re-parametrization of the mixed model is introduced and some results of its Bayesian extension are shown.

Let $(Z_1, \dots, Z_p)$ be p categorical variables, $(Y_1, \dots, Y_q)$ q continuous variables. Let also $\mathcal{I}$ be the space of all possible configurations of the p categorical variables and $\boldsymbol{\theta} = \{\theta(s), s \in \mathcal{I}\}$ where $\theta(s) = \Pr(Z_1 = s_1, \dots Z_p = s_p)$ is the probability to observe configuration $s = (s_1, \dots, s_p)$.

Under the HCG assumption we can write for each $s \in \mathcal{I}$

$$Y_1(s), \dots, Y_q(s) \mid \boldsymbol{\mu}(s), \boldsymbol{\Omega} \sim \mathcal{N}_q(\boldsymbol{\mu}(s), \boldsymbol{\Omega}^{-1}). \tag{3.12}$$

In particular, the relation between the canonical representation and the moments of the Gaussian model can be expressed through the re-parameterization $\boldsymbol{\mu}(s) = \mathbf{K}^{-1}\mathbf{h}(s)$ and $\boldsymbol{\Omega} = \mathbf{K}^{-1}$.

Consider now a collection of n i.i.d. observations $\boldsymbol{z}_i = (z_{i,1}, \dots, z_{i,p})^T$, $\boldsymbol{y}_i = (y_{i,1}, \dots, y_{i,q})^T$, $i = 1, \dots, n$ from (3.2) and (3.12). Categorical data $\{\boldsymbol{z}_i, i = 1, \dots, n\}$, can be equivalently represented as a contingency table of counts $\boldsymbol{N}$ with elements $n(s) \in \boldsymbol{N}$ such that

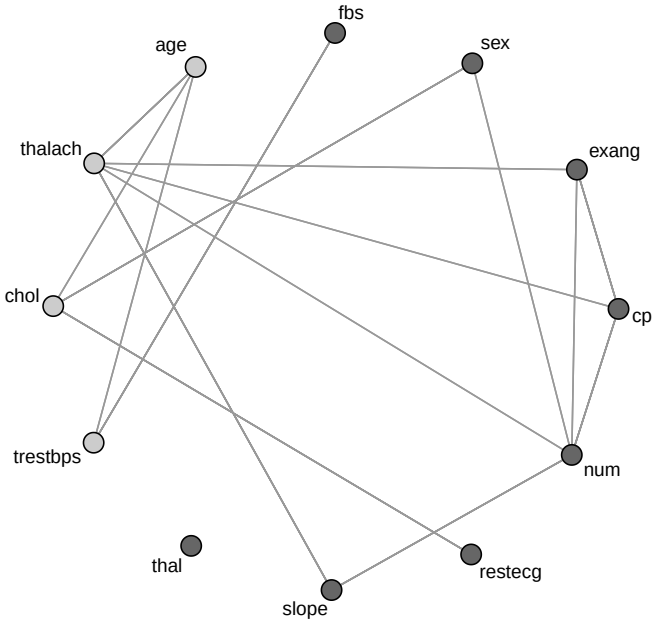
$$n(s) = \sum_{i=1}^{n} \mathbf{1}(\boldsymbol{z}_i = s),$$

where $\mathbf{1}(\cdot)$ is the indicator function and $\sum_{s\in\mathcal{I}} n(s) = n$. Following [36], the likelihood function can be written as

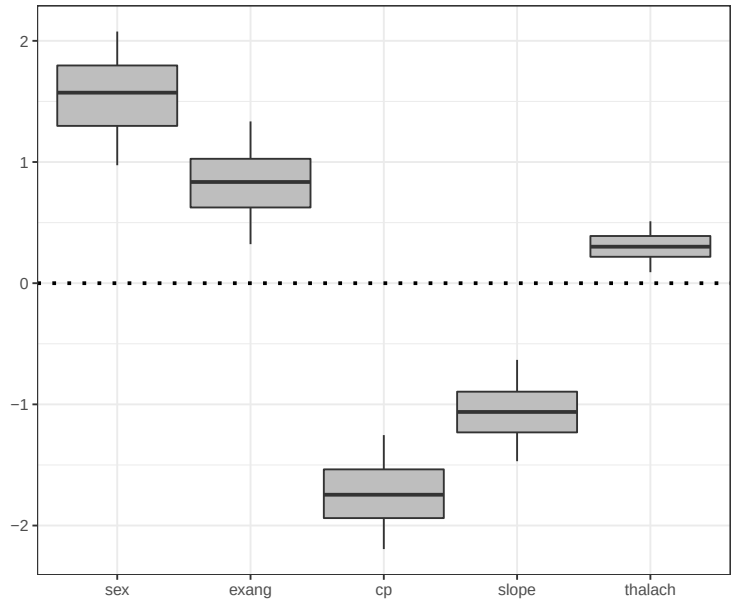
$$\begin{aligned} f(\boldsymbol{N}, \boldsymbol{y}_1, \dots, \boldsymbol{y}_n \mid \boldsymbol{\theta}, \{\boldsymbol{\mu}(s)\}_{s\in\mathcal{I}}, \boldsymbol{\Omega}) &= \prod_{s\in\mathcal{I}} \theta(s)^{n(s)} \prod_{s\in\mathcal{I}} \prod_{i\in\nu(s)} \phi(\boldsymbol{y}_i \mid \boldsymbol{\mu}(s), \boldsymbol{\Omega}^{-1}) \\ &\propto \prod_{s\in\mathcal{I}} \theta(s)^{n(s)} \prod_{s\in\mathcal{I}} \prod_{i\in\nu(s)} |\boldsymbol{\Omega}|^{\frac{1}{2}} \exp\left\{-\frac{1}{2}(\boldsymbol{y}_i - \boldsymbol{\mu}(s))^T \boldsymbol{\Omega} (\boldsymbol{y}_i - \boldsymbol{\mu}(s))\right\}, \end{aligned} \tag{3.13}$$

where $\nu(s)$ is the set of observations among $\{1, \dots, n\}$ with observed configuration s and ϕ denotes the Gaussian density. We complete our Bayesian model formulation by assigning the following prior distributions

$$\boldsymbol{\theta} \sim \text{Dirichlet}(\boldsymbol{B}), \quad \boldsymbol{\mu}(s) \mid \boldsymbol{\Omega} \sim \mathcal{N}_q(\boldsymbol{m}(s), (a_\mu \boldsymbol{\Omega})^{-1}), \quad \boldsymbol{\Omega} \sim \mathcal{W}_q(a_\Omega, \boldsymbol{U}), \tag{3.14}$$



(a) *Estimated graph with light (dark) grey dots representing continuous (categorical) variables.*



(b) *Credible intervals at 90% posterior probability and quantiles of parameters corresponding to active interaction terms with the variable num (diagnosis of heart disease).*

Figure 3.12: *Heart disease data analysis: Results.*

where, in particular, $\boldsymbol{B} = \{b(s) \in \mathcal{I}\}$ and $\mathcal{W}_q(a_\Omega, \boldsymbol{U})$ denotes a Wishart distribution having expectation $a_\Omega \boldsymbol{U}^{-1}$, with $a_\Omega > q-1$ and $\boldsymbol{U}$ a s.p.d. matrix.

It is advisable to set hyperparameters to values leading to proper prior distributions. A standard way to proceed, whenever no substantial prior information is available, is to choose hyperparameters leading to weakly informative priors. In particular, $\boldsymbol{B}$ may be set equal to a vector with all equal (e.g., unit) components (each associated to one level of the categorical variables). With regard to the Normal priors, we can fix a zero mean, while $a_\mu = 1$. Finally, the hyperparameters of the Wishart distribution can be fixed as $a_\Omega = q$, $\boldsymbol{U} = \boldsymbol{I}_q$, the (q, q) identity matrix.

Under prior parameter independence, the posterior distribution can be written after standard calculations as

$$p(\boldsymbol{\theta}, \{\boldsymbol{\mu}(s)\}_{s\in\mathcal{I}}, \boldsymbol{\Omega} \mid \boldsymbol{N}, \boldsymbol{y}_1, \ldots, \boldsymbol{y}_n) \propto \prod_{s\in\mathcal{I}} \theta(s)^{b(s)+n(s)-1}$$
$$\cdot \prod_{s\in\mathcal{I}} \left\{ |\boldsymbol{\Omega}|^{\frac{1}{2}} | \exp\left\{ -\frac{1}{2}(n(s)+a_\mu)(\boldsymbol{\mu}(s)-\bar{\boldsymbol{m}}(s))^T \boldsymbol{\Omega}(\boldsymbol{\mu}(s)-\bar{\boldsymbol{m}}(s)) \right\} \right.$$
$$\left. \cdot |\boldsymbol{\Omega}|^{\frac{a_\Omega+n-q-1}{2}} \exp\left\{ -\frac{1}{2}\mathrm{tr}[(\boldsymbol{U}+\boldsymbol{R}+\boldsymbol{R}_0)\boldsymbol{\Omega}] \right\} \right. \quad (3.15)$$

with $\boldsymbol{R} = \sum_{s\in\mathcal{I}} \mathrm{SSD}(s)$,

$$\begin{aligned} \bar{\boldsymbol{m}}(s) &= \frac{a_\mu}{a_\mu + n(s)} \boldsymbol{m}(s) + \frac{n(s)}{a_\mu + n(s)} \bar{\boldsymbol{y}}(s), \\ \boldsymbol{R}_0 &= \sum_{s\in\mathcal{I}} \frac{a_\mu n(s)}{a_\mu + n(s)} (\boldsymbol{m}(s) - \bar{\boldsymbol{y}}(s))(\boldsymbol{m}(s) - \bar{\boldsymbol{y}}(s))^T, \end{aligned}$$

where $\mathrm{SSD}(s) = \sum_{i\in\nu(s)} \boldsymbol{e}_i \boldsymbol{e}_i^T$, $\boldsymbol{e}_i = (\boldsymbol{y}_i - \bar{\boldsymbol{y}}(s))$ and $\bar{\boldsymbol{y}}(s)$ is the $(q, 1)$ vector with sample means of $(Y_1, \ldots, Y_q)$ relative to observations $i \in \nu(s)$. Note that the independence is assumed also among the configuration s for the whole set of $\boldsymbol{\mu}$ in order to have prior independence between $\boldsymbol{\mu}(s)$'s. From Equation (3.15) it follows that

$$\begin{aligned} \boldsymbol{\theta} \mid \boldsymbol{N} &\sim \text{Dirichlet}(\boldsymbol{B} + \boldsymbol{N}) \\ \boldsymbol{\mu}(s) \mid \boldsymbol{N}, \boldsymbol{Y}, \boldsymbol{\Omega} &\sim \mathcal{N}_q(\bar{\boldsymbol{m}}(s), [(a_\mu + n(s))\boldsymbol{\Omega})]^{-1}) \\ \boldsymbol{\Omega} \mid \boldsymbol{Y} &\sim \mathcal{W}_q(a_\Omega + n, \boldsymbol{U} + \boldsymbol{R} + \boldsymbol{R}_0), \end{aligned} \quad (3.16)$$

where $\boldsymbol{Y}$ denotes the (n, q) data matrix, row-binding of the $\boldsymbol{y}_i$'s; see also [23] for details on multivariate Normal models with Normal-Wishart priors and posterior calculations.

In addition, because of conjugacy, the marginal data distribution

$$m(\boldsymbol{Y}, \boldsymbol{N}) = \int f(\boldsymbol{N}, \boldsymbol{Y} \mid \boldsymbol{\theta}, \{\boldsymbol{\mu}(s)\}_{s\in\mathcal{I}}, \boldsymbol{\Omega}) p(\boldsymbol{\theta}) \prod_{s\in\mathcal{I}} p(\boldsymbol{\mu}(s)) p(\boldsymbol{\Omega})\, d\boldsymbol{\theta} \prod_{s\in\mathcal{I}} d\boldsymbol{\mu}(s)\, d\boldsymbol{\Omega}, \quad (3.17)$$

can be computed from the ratio of prior and posterior normalizing constants. Specifically, the formula in 3.17 can be written also as

$$m(\boldsymbol{Y}, \boldsymbol{N}) = m(\boldsymbol{N})m(\boldsymbol{Y}|\boldsymbol{N}) \tag{3.18}$$

because of prior independence between parameters indexing the categorical and Gaussian components of the model.

The marginal data distribution of $\boldsymbol{N}$ can be obtained from the ratio of prior and posterior normalizing constant of Dirichlet distributions in (3.14) and (3.16), respectively. Specifically:

$$m(\boldsymbol{N}) = \frac{\Gamma(\sum_{s\in\mathcal{I}} b(s))}{\Gamma(\sum_{s\in\mathcal{I}}(b(s)+n(s)))} \prod_{s\in\mathcal{I}} \frac{\Gamma(\sum_{s\in\mathcal{I}}(b(s)+n(s)))}{\Gamma(\sum_{s\in\mathcal{I}} b(s))} \tag{3.19}$$

The marginal distribution relative to the continuous data can be derived from standard results on Matrix-Normal (MN) distributions with Normal-Wishart priors. According to Definition 2.2.1 in [46], the MN distribution arises when sampling from multivariate Normal distribution. Specifically, $\mathbf{X} \sim \mathcal{MN}_{n\times p}(\boldsymbol{\Xi}, \boldsymbol{\Sigma}, \boldsymbol{\Psi})$ if and only if $\text{vec}(\mathbf{X})$ is distributed as $\mathcal{N}_{np}(\text{vec}(\Xi), \boldsymbol{\Sigma} \otimes \boldsymbol{\Psi})$

In our case, we have a collection of multivariate normal priors on the mean vectors $\boldsymbol{\mu}(s)$, $s \in \mathcal{I}$,

$$[\boldsymbol{\mu}_1(s), \ldots, \boldsymbol{\mu}_q(s)] \sim \mathcal{N}_q(\boldsymbol{m}(s), (a_\mu \boldsymbol{\Omega})^{-1}) \tag{3.20}$$

that are conditionally independent given $\boldsymbol{\Omega}$.

Let $\boldsymbol{D}$ be a diagonal matrix of dimension $|\mathcal{I}|$ with elements equal to a_μ. Then, using the Kronecker product $\boldsymbol{D}^{-1} \otimes \boldsymbol{\Omega}^{-1}$ we obtain a block-diagonal matrix whose elements correspond to the covariance matrix in (3.20):

$$\begin{bmatrix} \frac{1}{a_\mu} & & \mathbf{0} \\ & \ddots & \\ \mathbf{0} & & \frac{1}{a_\mu} \end{bmatrix} \otimes \Omega^{-1} = \begin{bmatrix} \frac{1}{a_\mu}\boldsymbol{\Omega}^{-1} & & \mathbf{0} \\ & \ddots & \\ \mathbf{0} & & \frac{1}{a_\mu}\boldsymbol{\Omega}^{-1} \end{bmatrix} \tag{3.21}$$

Therefore, we obtain a Matrix-Normal distribution as

$$\underbrace{\begin{pmatrix} \boldsymbol{\mu}_1(s_1), \ldots, \boldsymbol{\mu}_q(s_1) \\ \boldsymbol{\mu}_1(s_2), \ldots, \boldsymbol{\mu}_q(s_2) \\ \vdots \quad \vdots \\ \boldsymbol{\mu}_1(s_{|\mathcal{I}|}), \ldots, \boldsymbol{\mu}_q(s_{|\mathcal{I}|}) \end{pmatrix}}_{\boldsymbol{E}} \sim \mathcal{MN}_{|\mathcal{I}|\times q}\left(\underbrace{\begin{bmatrix} \boldsymbol{m}^T(s_1), \\ \vdots \\ \boldsymbol{m}^T(s_{|\mathcal{I}|}) \end{bmatrix}}_{\boldsymbol{M}}, \quad \boldsymbol{D}^{-1}, \quad \boldsymbol{\Omega}^{-1} \right) \tag{3.22}$$

because

$$\text{vec}(\boldsymbol{E}) \sim \mathcal{N}_{q|\mathcal{I}|}(\text{vec}(\boldsymbol{M}), \boldsymbol{D}^{-1} \otimes \boldsymbol{\Omega}^{-1}) \tag{3.23}$$

Consequently, the joint prior on the collection of mean-vector and the precision matrix $\boldsymbol{\Omega}$ is a Matrix-Normal-Wishart (MNW). Thus, following the results from [18] , the normalizing constant of the MNW distribution is

$$\kappa_{prior}(\boldsymbol{D}, \boldsymbol{U}, a_\Omega) = \frac{(2\pi)^{\frac{q|\mathcal{I}|}{2}} \cdot 2^{\frac{qa_\Omega}{2}} \cdot \Gamma_q(\frac{a_\Omega}{2})}{|\boldsymbol{D}|^{\frac{q}{2}} |\boldsymbol{U}|^{\frac{a_\Omega}{2}}} \tag{3.24}$$

Using the conjugacy results from (3.16), we can use the same results to compute the normalizing constant of the posterior MNW distribution as a function of the updated parameters

$$\kappa_{post}(\boldsymbol{D}^*, \boldsymbol{U}^*, a_\Omega + n) \tag{3.25}$$

where $\boldsymbol{D}^*$ is a diagonal matrix of elements $[a_\mu + n(s_1), \ldots, a_\mu + n(s_{|\mathcal{I}|})]$ and $\boldsymbol{U}^*$ is equal to $\boldsymbol{U} + \boldsymbol{R} + \boldsymbol{R}_0$. Finally, the marginal data distribution for the continuous part is

$$m(\boldsymbol{Y}|\boldsymbol{N}) = (2\pi)^{-\frac{nq}{2}} \cdot \frac{\kappa_{post}}{\kappa_{prior}} \tag{3.26}$$

The result in (3.16) enables posterior inference on the parameters of an *unconstrained* (complete) graphical model. Specifically, by implementing a Monte Carlo sampler it is possible to infer the parameters of the marginal distribution of discrete variables and the parameters of the conditional distribution of continuous (Gaussian) variables.

3.7 Conclusion and next steps

In this work we discussed a new Bayesian methodology which allows to jointly model conditional independencies among mixed variables and perform structure learning of related undirected graphs. The novelty of our work is the adoption of the Conditional Gaussian (CG) setting of [64] to provide an extension of existing Bayesian methodologies which deal with Gaussian and categorical data separately [10, 11], to parameter inference and structure learning of mixed data. Along the lines of these works, we will develop an MCMC algorithm for parameter estimation and structure learning for DAG for the mixed case.

Relative to the literature, we avoid the reliance on composite likelihood approximations and we provide full posterior uncertainty quantification of the parameter space, at the price of the computational cost for the evaluation of the likelihood normalizing constant, a quantity so far neglected in the literature. Our methodology shows better in-simulation performances than state-of-the-art alternative methods, particularly in the structure learning task with improvements as the sample size increases. We further provide three applications in nanotechnology and medicine, showing the versatility of the method.

As future direction, trans-dimensional reversible jump samplers [43] can handle model uncertainty and then can in principle provide a joint answer to structure learning and parameter inference, but we expect that more elaborated algorithms will be needed for an efficient exploration of a huge model space. A relevant example in this direction is the algorithm of Godsill [41], which exploits the partial analytical structures of the model: through distinct moves dedicated separately to model-related and model-unrelated parameters, we expect it to ameliorate the efficiency of the statistical analysis.

Another direction of investigation can provide, in a new context of dependency learning, a solution typically adopted in the random network literature. Random network statistical models can easily become doubly intractable problems. This means that the statistical analysis is complicated by two normalizing constants that cannot be evaluated: the one of the posterior distribution and the one of the likelihood. The first problem is automatically solved by an appropriately built MCMC algorithm. The second one is related to a likelihood normalizing constant possibly dependent on the parameters: an adaptation of the exchange algorithm of Murray et al. [82] can provide a computationally feasible method of structure learning for mixed data.

Performance results for structure learning: summary

	n=500		n=1000		n=2000	
Measure	HMGM	BGM-MD	HMGM	BGM-MD	HMGM	BGM-MD
FPR	0.59 (0.21)	**0.27** (0.22)	0.63 (0.21)	**0.28** (0.19)	0.6 (0.23)	**0.27** (0.23)
MISR	44.75 (11.98)	**30** (15.19)	47.5 (11.49)	**29** (13.92)	45.5 (12.8)	**26.5** (16.1)
SPE	41.25 (21.26)	**73.42** (21.82)	36.53 (20.83)	**71.76** (18.72)	39.68 (22.87)	**72.6** (22.57)
SEN	**70.33** (21.97)	65.6 (20.1)	69.07 (19.95)	69.01 (16.02)	69.07 (20.39)	**72.62** (16.5)
PRE	54.85 (17.05)	**73.08** (21.86)	52.46 (17.14)	**71.77** (18.44)	54.09 (18.09)	**74.39** (19.63)
MCC	12.14 (27.09)	**39.32** (31.44)	5.74 (26.68)	**40.88** (27.99)	9.05 (29.23)	**45.53** (32.92)

Table 3.1: *Scenario with 5 variables of which 2 are categorical (values in bold correspond to better performance)*

	n=500		n=1000		n=2000	
Measure	HMGM	BGM-MD	HMGM	BGM-MD	HMGM	BGM-MD
FPR	0.57 (0.05)	**0.12** (0.06)	0.56 (0.04)	**0.11** (0.05)	0.57 (0.05)	**0.12** (0.06)
MISR	51.61 (6.36)	**14.78** (6.3)	51.11 (6.37)	**12.11** (5.8)	51.28 (6.29)	**11.78** (5.79)
SPE	43.23 (5.41)	**88.4** (6.28)	43.64 (4.45)	**89.01** (5.09)	43.15 (4.81)	**88.1** (5.6)
SEN	73.45 (19.46)	74.2 (20.25)	73.84 (19.3)	**85.19** (17.08)	75.07 (20.13)	**90.22** (12.54)
PRE	22.06 (8.21)	**59.26** (16.27)	22.3 (8.25)	**63.83** (13.51)	22.39 (8.19)	**63.3** (14.65)
MCC	12.94 (16.12)	**56.86** (16.92)	13.41 (16.67)	**66.15** (14.77)	14.05 (16.99)	**68.48** (13.96)

Table 3.2: *Scenario with 10 variables of which 5 are categorical (values in bold correspond to better performance)*

	n=500		n=1000		n=2000	
Measure	HMGM	BGM-MD	HMGM	BGM-MD	HMGM	BGM-MD
FPR	0.52 (0.03)	**0.17** (0.07)	0.51 (0.03)	**0.2** (0.08)	0.5 (0.03)	**0.11** (0.05)
MISR	49.08 (4.2)	**26.36** (6.99)	48.13 (3.74)	**27.86** (8.32)	47.87 (3.87)	**12.78** (4.31)
SPE	48.45 (3.18)	**83.25** (7.11)	49.42 (2.74)	**80.34** (7.77)	49.71 (2.63)	**88.82** (5.11)
SEN	**63.48** (12.91)	25.82 (20.02)	**64.27** (12.34)	31.32 (23.25)	64.51 (12.75)	**80.42** (14.07)
PRE	19.73 (5.55)	**23.66** (19.22)	20.22 (5.4)	**24** (17.85)	20.35 (5.53)	**59.46** (12.51)
MCC	8.88 (10.82)	8.73 (21.12)	10.17 (10.09)	10.43 (23.66)	10.53 (10.55)	**61.48** (11.52)

Table 3.3: *Scenario with 20 variables of which 10 are categorical (values in bold correspond to better performance)*

Case	Variables	Iterations	Burn-in	Thinning
Case 1	*Nanostructures* (4), *Temperature*, Pressure, Distance	60000	30000	50
Case 2	*Nanostructures* (4), Temperature, Pressure, Distance	50000	10000	10

Table 3.4: *Settings for nanostructure analysis. Discrete variables are in italic.*

	Estimate	SE	Geweke's statistic	Lag 100	Lag 500	Lag 1000
var34	-0.52710	0.00027	-0.42000	0.78711	0.28345	-0.00507
var35	0.83660	0.00029	0.01000	0.78373	0.30401	0.09679
var36	0.42690	0.00028	0.01000	0.80463	0.35637	0.11954
var37	0.10660	0.00025	0.55000	0.77482	0.28175	0.08456
var38	-0.19680	0.00029	-0.51000	0.76785	0.26243	0.16439
var39	-0.09170	0.00029	0.22000	0.77813	0.33216	0.11172
var40	-0.08770	0.00025	-0.39000	0.75453	0.27309	0.06242
var41	0.12890	0.00023	3.77000	0.78767	0.29852	0.00255
var42	-0.25330	0.00026	1.50000	0.80113	0.33408	0.11072
var43	0.25180	0.00024	1.09000	0.76334	0.20294	-0.01645
var44	-0.01620	0.00024	1.16000	0.78450	0.16507	-0.09884
var45	-0.17080	0.00022	3.13000	0.79478	0.38069	0.20531
var46	-0.19390	0.00023	1.01000	0.80070	0.30059	0.04753
var47	0.11030	0.00024	-0.29000	0.80095	0.32396	0.09674
var48	0.13330	0.00023	0.79000	0.81126	0.38814	0.08443
var49	0.10480	0.00019	0.67000	0.78748	0.34008	0.22020
var50	0.15190	0.00020	1.15000	0.77963	0.27524	0.11180
var51	-0.38830	0.00020	0.45000	0.77493	0.26070	0.11587
var52	-0.26710	0.00018	-0.10000	0.76880	0.19316	0.00504
var53	0.44120	0.00035	-2.06000	0.77264	0.25270	-0.06733
var54	-0.23600	0.00033	-0.31000	0.75479	0.24324	0.04424
var55	0.61010	0.00037	-4.43000	0.77852	0.28128	0.09642
var56	0.19930	0.00040	1.40000	0.82274	0.36918	0.05563
var57	-0.00930	0.00019	-1.24000	0.77888	0.29438	0.11316
var58	-0.28820	0.00021	1.54000	0.77980	0.22593	0.00526
var59	0.30810	0.00020	1.18000	0.76474	0.29183	0.06786
var60	0.17640	0.00021	-0.85000	0.80539	0.28827	0.09216
var61	1.25230	0.00014	-1.21000	0.81042	0.34842	0.10872

Table 3.5: *Diagnostic test from* `coda` *output for hepatitis dataset. Posterior estimate for the parameters corresponding to mixed interactions are shown together with Geweke's test for stationary distributions and auto-correlations at different lags.*

Bibliography

[1] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, and X. Zheng. Tensorflow: Large-scale machine learning on heterogeneous systems. *https://www.tensorflow.org/*, 2015.

[2] A. Azadeh, M. Saberi, A. Kazem, V. Ebrahimipour, A. Nourmohammadzadeh, and Z. Saberi. A flexible algorithm for fault diagnosis in a centrifugal pump with corrupted data and noise based on ANN and Support Vector Machine with hyper-parameters optimization. *Applied Soft Computing*, 13(3):1478–1485, 2013.

[3] A. Baddeley, E. Rubak, and R. Turner. *Spatial Point Patterns: Methodology and Applications with R*. Chapman and Hall/CRC Press, London, 2015.

[4] M.G. Bawendi, A.R. Kortan, M.L. Steigerwald, and L. E. Brus. X-ray structural characterization of larger Cadmium Selenide (CdSe) semiconductor clusters. *Journal of Chemical Physics*, 111:2564–2571, 1989.

[5] A. Bhadra and B. K. Mallick. Joint high-dimensional Bayesian variable and covariance selection with an application to eQTL analysis. *Biometrics*, 69:447–457, 2013.

[6] A. Bhadra, A. Rao, and V. Baladandayuthapani. Inferring network structure in non-Normal and mixed discrete-continuous genomic data. *Biometrika*, 74(1):185–195, 2018.

[7] L. Biewald. Experiment tracking with weights and biases. *https://www.wandb.com/*, 2020.

[8] R. Borgoni, L. Deldossi, L. Radaelli, and D. Zappa. Statistics for microelectronics. *Applied Stochastic Methods in Business and Industry*, 29:315–318, 2013.

[9] K. C. Cadien and L. Nolan. Chemical mechanical polishing method and practice. In *Handbook of Thin Film Deposition (4th Edition)*, pages 317–357. William Andrew Publishing, 2018.

[10] F. Castelletti, G. Consonni, M. Della Vedova, and S. Peluso. Learning Markov equivalence classes of directed acyclic graphs: an objective Bayes approach. *Bayesian Analysis*, 13(4):1235–1260, 2018.

[11] F. Castelletti and S. Peluso. Equivalence class selection of categorical graphical models. *Computational Statistics and Data Analysis*, 164(107304), 2021.

[12] G. Cestnik, I. Konenenko, and I. Bratko. Assistant-86: A knowledge-elicitation tool for sophisticated users. *Progress in Machine Learning*, pages 31–45, 1987.

[13] S. Chen, D. Witten, and A. Shojaie. Selection and estimation for mixed graphical models. *Biometrika*, 102(1):47–64, 2015.

[14] J. Cheng, L. Tianxi, E. Levina, and J. Zhu. High-dimensional mixed graphical models. *Journal of Computational and Graphical Statistics*, 26(2):367–378, 2017.

[15] C. Chow and C. Liu. Approximating discrete probability distributions with dependence trees. *IEEE Transactions on Information Theory*, 14:462–467, 1968.

[16] F. Ciampi. Corporate governance characteristics and default prediction modeling for small enterprises. an empirical analysis of Italian firms. *Journal of Business Research*, 68:1012–1025, 2015.

[17] F. Ciampi and N. Gordini. Small enterprise default prediction modeling through Artificial Neural Networks: An empirical analysis of Italian small enterprises. *Journal of Small Business Management*, 51:23–45, 2013.

[18] G. Consonni, L. La Rocca, and P. Peluso. Objective bayes covariate-adjusted sparse graphical model selection. *Scandinavian Journal of Statistics*, 44(3):741–764, 2017.

[19] N. Cressie. *Statistics for spatial data*. John Wiley & Sons, 1993.

[20] S. P. Cunningham and S. MacKinnon. Statistical methods for visual defect metrology. *IEEE Transactions on Semiconductor Manufacturing*, 11(1):48–53, 1998.

[21] H.-N. Dai, H. Wang, G. Xu, J. Wan, and M. Imran. Big data analytics for manufacturing internet of things: opportunities, challenges and enabling technologies. *Enterprise Information Systems*, 14(9-10):1279–1303, 2020.

[22] T. Dasgupta, C. Ma, V.R. Joseph, Z.L. Wang, and C.J. Wu. Statistical modeling and analysis for robust synthesis of nanostructures. *Journal of the American Statistical Association*, 103(482):594–603, 2008.

[23] M. Degroot. *Optimal statistical decisions*. John Wiley and Sons, 2004.

[24] P. Diaconis and B. Efron. Computer-intensive methods in statistics. *Scientific American*, 248(5):116–131, 1983.

[25] S. Dieleman. Recommending music on Spotify with deep learning. *http://benanne. github. io/2014/08/05/spotify-cnns.html*, 2014.

[26] P.J. Diggle. *Statistical analysis of spatial point patterns*. Arnold, 2003.

[27] M. Dikmen and C. M. Burns. Autonomous driving in the real world. *Proceedings of the 8th International Conference on Automotive User Interfaces and Interactive Vehicular Applications*, pages 225–228, 2016.

[28] M. Drakaki and P. Tzionas. Manufacturing scheduling using colored Petri nets and reinforcement learning. *Applied Sciences*, 7(2), 2017.

[29] R.B. Dull and D.P. Tegarden. Using control charts to monitor financial reporting of public companies. *International Journal of Accounting Information Systems*, 5(2):109–127, 2004.

[30] H. Edelsbrunner, D. Kirkpatrick, and R. Seidel. On the shape of a set of points in the plane. *IEEE Transactions on Information Theory*, 29(4):551–559, 1983.

[31] D. Edwards, G. De Abreu, and R. Labouriau. Selecting high-dimensional mixed graphical models using minimal AIC or BIC forests. *BMC Bioinformatics*, 11(8), 2010.

[32] M. Ester, H. Kriegel, J. Sander, and X. Xu. A density-based algorithm for discovering clusters in large databases with noise. *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, 1996.

[33] B. Fellinghauer, P. Bühlmann, M. Ryffel, M. Von Rhein, and J. D. Reinhardt. Stable graphical model estimation with random forests for discrete, continuous, and mixed variables. *Computational Statistics and Data Analysis*, 64:132–152, 2013.

[34] H.H. Fellows, C.M. Mastrangelo, and P.K. White Jr. An empirical comparison of spatial randomness models for yield analysis. *IEEE Transactions on Electronics Packaging Manufacturing*, 32(2):115–120, 2009.

[35] D.J. Friedman and S.L. Albin. Clustered defects in IC fabrication: Impact on process control charts. *IEEE Transactions on Semiconductor Manufacturing*, 4(1):36–42, 1991.

[36] M. Frydenberg and S. Lauritzen. Decomposition of maximum likelihood in mixed graphical interaction models. *Biometrika*, 76(3):539–555, 1989.

[37] R. Garthoff and P. Otto. Control charts for multivariate spatial autoregressive models. *AStA Advances in Statistical Analysis*, 101(1):67–94, 2017.

[38] S. Gaurav and P. DeoRaj. Control chart applications in healthcare: a literature review. *International Journal of Metrology and Quality Engineering*, 9:5, 2018.

[39] A. Gelman, J. B. Carlin, H. S. Stern, and D. B. Rubin. *Bayesian Data Analysis.* Boca Raton: Chapman & Hall/CRC, 2003.

[40] J. K. Ghosh and R. V Ramamoorthi. *Bayesian Nonparametrics*. Springer, 2003.

[41] S. J. Godsill. On the relationship between Markov chain Monte Carlo methods for model uncertainty. *Journal of Computational and Graphical Statistics*, 10:230–248, 2012.

[42] J. C. Gower and G. J. S Ross. Minimum spanning trees and single linkage cluster analysis. *Journal of the Royal Statistical Society. Series C (Applied Statistic)*, 18(1):54–64, 1969.

[43] P. J. Green. Reversible jump Markov chain Monte Carlo computation and Bayesian model determination. *Biometrika*, 82(4):711–732, 1995.

[44] O.A. Grigg and D.J. Spiegelhalter. An empirical approximation to the null unbounded steady-state distribution of the cumulative sum statistic. *Technometrics*, 54:523–529, 2008.

[45] S.D. Grimshaw, N.J. Blades, and M.P. Miles. Spatial control charts for the mean. *Journal of Quality Technology*, 45(2):130–148, 2013.

[46] A.K. Gupta and D.K. Nagar. *Matrix Variate Distributions*. Taylor & Francis, 1999.

[47] M.H. Hansen, V.N. Nair, and D.J. Friedman. Monitoring wafer map data from integrated circuit fabrication processes for spatially clustered defects. *Technometrics*, 39(3):241–253, 1997.

[48] T. Hastie and W. Stuetzle. Principal curves. *Journal of the American Statistical Association*, 84(406), 1989.

[49] D.S. Hochbaum. Adjacency-clustering for identifying defect patterns and yield prediction in integrated circuit manufacturing. *IFAC Conference on Manufacturing Modelling, Management and Control MIM*, 52(13):2086–2091, 2019.

[50] G. Hodes, A. Albu-Yaron, F. Decker, and P. Motisuke. Three-dimensional quantum-size effect in chemically deposited Cadmium Selenide films. *Physics Review B*, 36:4215–4221, 1987.

[51] J.Y. Hwang and W. Kuo. Model-based clustering for integrated circuit yield enhancement. *European Journal of Operational Research*, 178(1):143–153, 2007.

[52] J. Illian, A. Penttinen, H. Stoyan, and D. Stoyan. *Statistical Analysis and Modelling of Spatial Point Patterns*. John Wiley and Sons,Ltd, 2008.

[53] M. Innes. Flux: Elegant machine learning with Julia. *Journal of Open Source Software.*, doi:10.21105/joss.00602, 2018.

[54] M. Innes, E. Saba, K. Fischer, D. Gandhi, M. C. Rudilosso, N. M. Joy, and V. Shah. Fashionable modelling with Flux. *https://arxiv.org/abs/1811.01457*, 2018.

[55] Y.-S. Jeong, M.K. Jeong, and S.-J. Kim. Automatic identification of defect patterns in semiconductor wafer maps using spatial correlogram and dynamic time warping. *IEEE Transactions on Semiconductor Manufacturing*, 21(4):625–637, 2008.

[56] C.H. Jin, H.J. Kim, Y. Piao, M. Li, and M. Piao. Wafer map defect pattern classification based on convolutional neural network features and error-correcting output codes. *Journal of Intelligent Manufacturing*, 31(8):1861–1875, 2020.

[57] C.H. Jin, H.J. Na, M. Piao, G. Pok, and K.H. Ryu. A novel DBSCAN-based defect pattern detection and classification framework for wafer bin map. *IEEE Transactions on Semiconductor Manufacturing*, 32(3):286–292, 2019.

[58] R. Jothi, S.K. Mohanty, and A. Ojha. Fast approximate minimum spanning tree based clustering algorithm. *Neurocomputing*, 272:542–557, 2018.

[59] M. Kalisch and P. Bühlmann. Estimating high-dimensional directed acyclic graphs with the PC-algorithm. *Journal of Machine Learning Research*, 8(22):613–636, 2007.

[60] S. Khodabandehlou and M. Zivari Rahman. Comparison of supervised machine learning techniques for customer churn prediction based on analysis of customer behavior. *Journal of Systems and Information Technology*, 19:65–93, 2017.

[61] A. Kusiak. Smart manufacturing. *International Journal of Production Research*, 56(1-2):508–517, 2018.

[62] T.Y. Kwon, M. Ramachandran, and J.G. Park. Scratch formation and its mechanism in chemical mechanical planarization (CMP). *Friction*, 1(4):279–305, 2013.

[63] S. Lauritzen. *Graphical models*. Oxford Press, 1996.

[64] S. Lauritzen and N. Wermuth. Graphical models for associations between variables, some of which are qualitative and some quantitative. *The Annals of Statistics*, 17(1):31–57, 1989.

[65] J. Lee and T. Hastie. Learning the structure of mixed graphical models. *Journal of Computational and Graphical Statistics*, 24(1):230–253, 2015.

[66] T. Lee-Ing, W. Chung-Ho, and H. Chih-Li. Monitoring defects in IC fabrication using a hotelling T^2 control chart. *IEEE Transactions on Semiconductor Manufacturing*, 18(1):140–147, 2005.

[67] S. Lessmann, B. Baesens, H.-V. Seow, and L. C. Thomas. Benchmarking state-of-the-art classification algorithms for credit scoring: An upyear of research. *European Journal of Operational Research*, 247:124–136, 2015.

[68] W. Li, Y. Liang, and S. Wang. *Data Driven Smart Manufacturing Technologies and Applications*. Springer Series in Advanced Manufacturing, 2021.

[69] Z. Li, P. Qiu, S. Chatterjee, and Z. Wang. Using p-values to design statistical process control charts. *Statistical Papers*, 50:501–511, 2013.

[70] U. Lichtenthaler. Substitute or synthesis: The interplay between human and artificial intelligence. *Research-Technology Management*, 61:12–14, 2018.

[71] T. P. Lillicrap, A. Santoro, L. Marris, C. J. Akerman, and G. Hinton. Backpropagation and the brain. *Nature Reviews Neuroscience*, 21:0353–346, 2020.

[72] C. Ma, Y. Ding, D.F. Moore, X. Wang, and Z.L. Wang. Single-crystal CdSe nanosaws. *Journal of the American Chemical Society*, 126:708–709, 2004.

[73] C. Ma and Z.L. Wang. Roadmap for controlled synthesis of CdSe nanowires. *Nanobelts and Nanosaws, Advanced Materials*, 17:1–6, 2005.

[74] D.P. Mandal and C.A. Murthy. Selection of alpha for alpha-hull in R^2. *Pattern Recognition*, 30(10):1759–1767, 1997.

[75] D.J. Marchette. *Random graphs for statistical pattern recognition*. John Wiley & Sons, 2004.

[76] A.D. Martin, K.M. Quinn, and J.H. Park. Markov chain Monte Carlo in R. *Journal of Statistical Software*, 42(9):1–21, 2011.

[77] G.S. May and C.J. Spanos. *Fundamentals of semiconductor manufacturing and process control.* Wiley-Interscience, 2006.

[78] N. Meinshausen and P. Bühlmann. High-dimensional graphs and variable selection with the Lasso. *The Annals of Statistics*, 34(3):1436–1462, 2006.

[79] M. Mitchell. Why AI is harder than we think. *Proceedings of the Genetic and Evolutionary Computation Conference*, 2021.

[80] D.C. Montgomery. *Statistical quality control: a modern introduction*. Wiley, 7th edition, 2012.

[81] A. Mosavi, Y. Faghan, P. Ghamisi, P. Duan, S. F. Ardabili, E. Salwana, and S. S. Band. Comprehensive review of deep reinforcement learning methods and applications in economics. *Mathematics*, 8(10), 2020.

[82] I. Murray, Z. Ghahramani, and D. MacKay. MCMC for doubly-intractable distribution. *Proceedings of the 22nd Annual Conference on Uncertainty in Artificial Intelligence*, 2006.

[83] M.PL. Ooi, H.K. Sok, YC Kuang, S Demidenko, and C. Chan. Defect cluster recognition system for fabricated semiconductor wafers. *Engineering Applications of Artificial Intelligence*, 26(3):1029–1043, 2013.

[84] E.S. Page. Continuous inspection schemes. *Biometrika*, 42:243–254, 1954.

[85] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, and S. Chintala. Pytorch: An imperative style, high-performance deep learning library. *Advances in Neural Information Processing Systems*, 32, 2019.

[86] N. Plummer, M. amd Best, K. Cowles, and K. Vines. Coda: Convergence diagnosis and output analysis for MCMC. *R News*, 6(1):7–11, 2006.

[87] R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2017.

[88] R. Ren, T. Hung, and K. C. Tan. A generic deep-learning-based approach for automated surface inspection. *IEEE Transactions on Cybernetics*, 48(3):929–940, 2018.

[89] C.P. Robert and G. Casella. *Monte Carlo Statistical Methods*. New York: Springer, 2004.

[90] R. Rosly, M. Makhtar, M.K. Awang, M.I. Awang, and M.N. Rahman. Analyzing performance of classifiers for medical datasets. *International journal of engineering and technology*, 136(7), 2018.

[91] S. Sahoo. Big data analytics in manufacturing: a bibliometric analysis of research in the field of business management. *International Journal of Production Research*, pages 1–29, 2021.

[92] E. Santana and G. Hotz. Learning a driving simulator. *Learning a Driving Simulator*, 2016.

[93] Y.J. Seo, S.Y. Kim, and W.S. Lee. Advantages of point of use (POU) slurry filter and high spray method for reduction of CMP process defects. *Microelectronic Engineering*, 70(1):1–6, 2003.

[94] Y. Shang, T. Li, L. Song, and Z. Wang. Control charts for monitoring two-dimensional spatial count data with spatial correlations. *Computers and Industrial Engineering*, 137, 2019.

[95] M.S. Sivri and B. Oztaysi. Data analytics in manufacturing. In *Industry 4.0: Managing The Digital Transformation*, pages 155–172. Springer Series in Advanced Manufacturing, 2018.

[96] R. Sripriya, M. D. Kaulaskar, S. Chakraborty, and B. C. Meikap. Studies on the performance of a hydrocyclone and modeling for flow characterization in presence and absence of air core. *Chemical Engineering Science*, 62(22):6391–6402, 2007.

[97] D. C. Stanford and A. E. Raftery. Finding curvilinear features in spatial point patterns: principal curve clustering with noise. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(6):601–609, Jun 2000.

[98] A. Ustundag and E. Cevikcan. *Industry 4.0: Managing The Digital Transformation*. Springer Series in Advanced Manufacturing, 2018.

[99] A. W. van der Vaart. *Asymptotic Statistics*. Cambridge University Press, 2000.

[100] C.H. Wang. Recognition of semiconductor defect patterns using spatial filtering and spectral clustering. *Expert Systems with Applications*, 34(3):1914–1923, 2008.

[101] R. Wang and N. Chen. Defect pattern recognition on wafers using convolutional neural networks. *Quality and Reliability Engineering International*, 36(4):1245–1257, 2020.

[102] E. Yang, G. Allen, Z. Liu, and P. Ravikumar. Graphical models via generalized linear models. *Advances in Neural Information Processing Systems*, 25, 2012.

[103] E. Yang, Y. Baker, P. Ravikumar, G. Allen, and Z. Liu. Mixed graphical models via exponential families. *Proceedings of the Seventeenth International Conference on Artificial Intelligence and Statistics*, 33:1042–1050, 2014.

[104] T. Yuan, W. Kuo, and S.J. Bae. Detection of spatial defect patterns generated in semiconductor fabrication processes. *IEEE Transactions on semiconductor manufacturing*, 24(3):392–403, 2011.

[105] C. T. Zahn. Graph-theoretical methods for detecting and describing gestalt clusters. *IEEE Transactions on Computers*, C-20(1):68–86, 1971.

[106] H. Zareifard, V. Rezaei Tabar, and D. Plewczynski. A Gibbs sampler for learning DAG: A unification for discrete and Gaussian domains. *Journal of Statistical Computation and Simulation*, 91(14):2833–2853, 2021.

[107] Q. Zhou, L. Zeng, and S. Zhou. Statistical detection of defect patterns using Hough transform. *IEEE Transactions on Semiconductor Manufacturing*, 23(3):370–380, 2010.

[108] Y. Zhou, O. Grigorash, and T. F.Hain. Clustering with minimum spanninig tree. *International Journal on Artificial Intelligence Tools*, 20:139–177, 2011.

[109] R. Zhuang, S. Noah, and J. Lederer. Graphical models for discrete and continuous data. *ArXiv:1609.05551 [Math, Stat]*, 2019.

[110] Y. Zuo. Prediction of consumer purchase behaviour using Bayesian network: An operational improvement and new results based on RFID data. *International Journal of Knowledge Engineering and Soft Data Paradigms*, 5(2):85–105, 2016.