

**FINDING CAUSALITY IN SOCIOTECHNICAL SYSTEMS A
BAYESIAN NETWORK STRUCTURE LEARNING APPROACH TO
PROBLEM UNDERSTANDING**

A Dissertation
Presented to
The Academic Faculty

by

William Davies Martin

In Partial Fulfillment
of the Requirements for the Degree
Masters of Science of Mechanical Engineering in the
Woodruff School of Mechanical Engineering

Georgia Institute of Technology
May 2018

COPYRIGHT © 2017 BY WILLIAM MARTIN

**FINDING CAUSALITY IN SOCIOTECHNICAL SYSTEMS A
BAYESIAN NETWORK STRUCTURE LEARNING APPROACH TO
PROBLEM UNDERSTANDING**

Approved by:

Dr. Cassandra Telenko, Advisor
School of Mechanical Engineering
Georgia Institute of Technology

Dr. Katherine Fu
School of Mechanical Engineering
Georgia Institute of Technology

Dr. Yan Wang
School of Mechanical Engineering
Georgia Institute of Technology

Date Approved: December 5, 2017

COPYRIGHT © 2017 BY WILLIAM MARTIN

In loving memory of Vivian Ziwei Guan

“I want to live life shamelessly, mischievously, and infectiously awake”



My hope is that this work embodies her desire to create a more equitable world for all people through design.

ACKNOWLEDGEMENTS

I would like to acknowledge the support of my friends, my pets, and Family, without whom this would not have been possible.

TABLE OF CONTENTS

ACKNOWLEDGEMENTS	iv
LIST OF TABLES	viii
LIST OF FIGURES	ix
LIST OF SYMBOLS AND ABBREVIATIONS	xi
SUMMARY	xiii
Chapter 1. Introduction	1
1.1 Definition of a Sociotechnical System	2
1.2 Sociotechnical systems and sustainability	2
1.3 Potential for Machine Learning	4
1.4 Bike Share Case Study	4
1.5 Chapter Summary	5
Chapter 2. Approaches to Learning Causality from Data	7
2.1 Sociotechnical data	7
2.1.1 Small data size	8
2.1.2 Data Amalgamated from multiple sources	9
2.1.3 Data Sparseness	10
2.1.4 Latent Variables	10
2.2 Networks built from data	11
2.2.1 Feature Selection	11
2.2.2 Neural Networks	12
2.2.3 Bayesian Networks	14
2.3 Chapter Summary	16
Chapter 3. Bayesian Network Structure Learning	17
3.1 Bayesian Network Concepts	17
3.1.1 Bayes' Theorem	17
3.1.2 D-separation and d-connection	21
3.1.3 Markov equivalent structures	24
3.2 Assumptions underlying BN Structure Learning	25
3.2.1 Causal Sufficiency Assumption	26
3.2.2 Markov Assumption	28
3.2.3 Faithfulness Assumption	29
3.3 Structure learning	32
3.3.1 Expert built networks	33
3.3.2 Search and score	34
3.3.3 Conditional Independence	41
3.3.4 Hybrid approaches	43
3.3.5 Exhaustive searches	45

3.3.6	Averaging BNs	46
3.4	Parameter Learning	46
3.4.1	Maximum Likelihood Estimate	46
3.4.2	Bayesian approach	47
3.5	Predictive Inference	47
3.5.1	Junction Tree Algorithm	48
3.6	Validation of network	49
3.6.1	Predictive Inference Accuracy	49
3.6.2	Causal match with Expert networks	49
3.7	Chapter Summary	50
Chapter 4. Datasets and Methodology		51
4.1	Data	52
4.1.1	ALARM Data	53
4.1.2	On Board Diagnostic (OBD) Data	55
4.1.3	Divvy Data	58
4.1.4	Datasets and assumptions	63
4.2	Methodology	64
4.2.1	Pre-processing/Characterizing data	64
4.2.2	Learning Bayesian Network Structure	71
4.2.3	Learning BN parameters	72
4.2.4	Evaluating Predictive Inference Accuracy	72
4.2.5	Evaluating structure against expert model	73
4.3	Chapter Summary	75
Chapter 5. Results and Analysis		76
5.1	Results	76
5.1.1	ALARM Dataset	76
5.1.2	OBD Dataset	81
5.1.3	Divvy Dataset	86
5.2	Alternative Network Comparisons for Observational Datasets	91
5.2.1	Time execution of Algorithms	91
5.2.2	Density of Learned Structures	92
5.2.3	OBD Data	94
5.2.4	Divvy Data	99
5.3	Chapter summary	103
Chapter 6. Discussion		105
6.1	Data Assumptions and Algorithms	105
6.1.1	Other contributing factors	107
6.2	Bike share Case Study	108
6.2.1	Learned relationships	109
6.2.2	Qualitative Verification of Learned Relationships	115
6.3	Chapter Summary	116
Chapter 7. Conclusions and Future Work		118
7.1	Future work	120

7.1.1 Latent Variable Detection	121
7.1.2 Data Discretization	123
7.2 Limitations	123
APPENDIX A. OBD GOVERNING EQUATIONS	126
REFERENCES	127

LIST OF TABLES

Table 1: Algorithms Types, Strengths, and Weaknesses.....	41
Table 2: Variables in the OBD Dataset.....	57
Table 3: Variables in the Bike Share Dataset.....	61
Table 4: Assumptions and datasets.....	64
Table 5: Sample size Breakdown ALARM.....	68
Table 6: Sample size Breakdown OBD.....	70
Table 7: Sample size Breakdown Divvy.....	71
Table 8: Algorithm Default Parameters.....	72
Table 9: Standard Deviation for Results of OBD.....	86
Table 10: Standard Deviation of Results for Divvy.....	91
Table 11: Assumptions and Evidence.....	106

LIST OF FIGURES

Figure 1: Markov Blanket [30]	12
Figure 2: Sample Neural Network [33]	13
Figure 3: Bayesian Network [35]	15
Figure 4: Four Types of Paths [40]	22
Figure 5: Descendent of Common Effect [40]	23
Figure 6: Markov Equivalence [41]	25
Figure 7: Example of Causal Sufficiency for Smoking Symptoms [40]	27
Figure 8: DAG Faithfulness Assumption Figure [40]	30
Figure 9: Process Diagram	52
Figure 10: ALARM Network [66]	54
Figure 11: Expert OBD Network	56
Figure 12: OBD Variables	58
Figure 13: Expert Network for Divvy	60
Figure 14: Three tiers of Variables	63
Figure 15: Instrumental Variables	65
Figure 16: Runtime of Algorithm for ALARM	77
Figure 17: Density of Algorithms for ALARM	78
Figure 18: MSHD of Algorithms for ALARM	79
Figure 19: SHD of Algorithms for ALARM	79
Figure 20: Accuracy of Algorithms for ALARM	81
Figure 21: Runtime of Algorithms for OBD	82
Figure 22: Density of Algorithms for OBD	83
Figure 23: MSHD of Algorithms for OBD	84
Figure 24: SHD of Algorithms for OBD	84
Figure 25: Accuracy of Algorithms for OBD	85
Figure 26: Runtime of Algorithms for Divvy	87
Figure 27: Density of Algorithms for Divvy	88
Figure 28: MSHD for Divvy	89
Figure 29: SHD for Divvy	89
Figure 30: Accuracy of Algorithms for Divvy	90
Figure 31: OBD Unconnected Network	95
Figure 32: OBD Correlation Network with threshold 0.3 (arcs represent +/- correlation prior to discretization)	95
Figure 33: OBD Expert Network	96
Figure 34: OBD Average BN with a Threshold of 60%	98
Figure 35: Divvy Unconnected Network	100
Figure 36: Divvy Correlation Network with a threshold of 0.3 (arcs represent +/- correlation prior to discretization)	100
Figure 37: Divvy Expert Network	101
Figure 38: Divvy Average BN	102
Figure 39: Divvy Correlation Network with a threshold of 0.3 (arcs represent +/- correlation prior to discretization)	109

Figure 40: Average Temperature per Gender Ratio	111
Figure 41: Average Temperature for each number of Riders	112
Figure 42: Average BN Structure Divvy.....	112
Figure 43: % Workday vs Gender Ratio	114
Figure 44: % Workday vs Number of Riders.....	115
Figure 45: Research Direction	118
Figure 46: Clique and clique with latent variable	122

LIST OF SYMBOLS AND ABBREVIATIONS

\mathcal{G}	Symbol representing a Bayesian Network structure
Θ	Symbol representing the parameters in a Bayesian Network
\mathcal{D}	Data generated by a known/unknown Bayesian Network
Ω	Symbol representing the entire state space for the variables
n	Number of unique variables in Bayesian Network
i	Variable used for enumerating
j	Variable used for enumerating
k	Variable used for enumerating
\cap	Set operator indicating that two events coincide
\subseteq	One variable is a subset of another and is included in other
\cup	Set operator indication that variable belongs in at least one of two sets
\in	One variable is an element of another
X_n	Symbol representing each of n variables in a network
r_i	The number of discrete states a variable X_i can take on
\prod_{X_i}	Configurations of the parent variables for X_i
q_i	The number of possible parent configurations of a set \prod_{X_i} for X_i

w_{ij}	Represents a single configuration of $\prod x_i$ where j is $(1 < j < q_i)$
Γ	The gamma function
N	The number of samples in a training dataset used for learning
N'	Hyper prior for N
N_{ij}	The number of instances where the variable has a particular parental configuration
N'_{ij}	Hyper prior for N_{ij}
N_{ijk}	Number of instances where the variable takes on a particular value with particular parental configuration
N'_{ijk}	Hyper prior for N_{ijk}
LL	Log likelihood score
BIC	Bayesian Information Criterion
MDL	Minimum description length
AIC	Akaike Information Criterion
\emptyset	An unconnected Bayesian network or empty parent set
$X \perp\!\!\!\perp Y S$	X is conditionally independent of Y given set S
Pa^B	Parental variables for a network B
\forall_i	For all symbol meaning operation applies to all
C_i^n	Candidate Parental variables for variable i on iteration n

SUMMARY

Understanding the causal relations governing sociotechnical systems allows designers to better predict system dynamics, identify the root cause of issues, and subsequently design more sustainable systems. Visual representations can help develop this understanding and may be generated by a variety of approaches. This research explores the use of Bayesian Network (BN) structure learning algorithms to generate Directed Acyclic Graphs (DAGs) to represent system relationships.

BNs are a particularly promising approach to visual representation because they convey information about dependencies and independencies in a system. BNs are often generated manually by experts, but many data driven BN learning algorithms have been developed that may aid non-experts in making decisions. These data driven approaches are beneficial when experts are not available, or designers want to avoid biases that experts might have. Nevertheless, most BN learning algorithms are designed for data that satisfies the Causal Sufficiency Assumption, Markov Assumption, and Faithfulness Assumption; all of which are not usually fully satisfied in real world sociotechnical data.

This research aims to evaluate alternative methods for automating discovery of relationships between variables in observational datasets. The evaluation considers the assumptions made in BN learning methods about the observational data. This is accomplished by combining the methodologies of several independent researchers to characterize the assumptions valid in different datasets [1][2][3]. Several kinds of BN structure learning algorithms and post processing techniques are then used to learn causal

Commented [MWD1]: This instance I am just saying causal relationships, which is what we are trying to understand

Commented [TC2R1]: ok

Commented [TC3]: we didn't develop methodologies, just tested existing ones on observed data

Commented [TC4]: We didn't develop a guide.

Commented [WMSR4]: True just combined others work

Commented [TC6]: Add citation

Commented [WM7R6]: Will do sorry I did not know if citations went in summaries

networks from the datasets. The performances of the learned networks are then compared by the network's match with (an expert) network and ability to perform predictive inference.

Commented [MW08]: Talking about the true causal structure, but not saying BN is a causal structure

Comparing datasets with different valid assumptions showed that learning algorithms decreased in their ability to perform predictive inference and recover the expert specified structure as the number of valid assumptions decreased. All the algorithms performed similarly, and the post-processing approach did not improve results when all of the assumptions were valid in a simulated dataset. The results indicated that utilizing algorithms partially based on explicit tests of independence and using an Average BN post-processing approach gave better performing networks for observational datasets that only partially satisfied the three assumptions. Finally, when none of the assumptions were proven valid, the algorithms and post-processing techniques all returned equally poor performing networks. Future work would assess more datasets and integrate human guidance with the algorithms to better define the guidelines for utilizing BN structure learning on the wide variety of datasets encountered in sociotechnical systems.

CHAPTER 1. INTRODUCTION

The goal of this research is to examine the capabilities of Bayesian network learning methods for automating the characterization of sociotechnical datasets post-processing. Sociotechnical systems involve the complex interaction of technology, people making decisions, and the environment[4]. Learning the causal structure of these systems could help designers better distinguish root causes, system dynamics, and symptoms of inefficiencies during the problem formulation stage of design, allowing for more sustainable systems level design [5]. Sustainability is an emergent characteristic of systems that cannot be predicted simply from observing a single element of a sociotechnical system. Instead, sustainability requires understanding the current behavior of sociotechnical systems and identifying the present needs of that system as well as long-term effects of the needs of future generations [6].

Learning informative network structure from observational data has already been accomplished in the fields of epidemiology, bioinformatics, operations management, and system maintenance to help researchers understand how pathogens spread, genes regulate one another, systems interact, and assets should be maintained [7][8][9][10]. Similarly, using observational data to generate informative networks for sociotechnical systems can support problem understanding by subject matter experts in sustainable systems design. Using causal structure learning algorithms as part of the model-building process can help fill knowledge gaps and check biases of practitioners.

Commented [TC9]: Please revise throughout to clarify that we just examined existing methods.

Commented [WM10R9]: Will be sure that is clear throughout

Commented [MWD11]: Just saying there is SOME causal structure

Commented [MWD12]: Here I am saying that causal graphs were learned, changed as seen

Commented [MWD13]: Saying it is causal revised

Commented [TC14R13]: ok

1.1 Definition of a Sociotechnical System

A sociotechnical system is a system that involves the interaction of technology, people making decisions, and the environment [4]. A classic example of a sociotechnical system is traffic. In traffic the technology is the vehicles, people must make decisions about their driving behavior, and the environment interacts with both the vehicles and people. Other sociotechnical systems include hiring decisions, maintenance, and the reliability of industrial systems [11]. A lot of work has been done in modeling these types of systems and trying to predict when they will break down or have problems [12]. Sociotechnical systems are present in industry, government, non-profit work, and peoples' daily lives. Any time you have people technology and the environment affecting each other you have a sociotechnical system.

1.2 Sociotechnical systems and sustainability

Understanding sociotechnical systems is fundamental in sustainability and sustainable design [11]. Sustainable design is design that meets the needs of the present without compromising the ability of future generations to meet their own needs [6]. To achieve sustainable development designers have to take into account the three pillars of sustainability which are environmental, social, and economic sustainability [13]. Environmental sustainability is the concept that people are most familiar with and relates to not extracting more natural resources than the environment can replenish in a given period. Social sustainability relates to the fair and equitable treatment of people in the system, ensuring that everyone can live a productive and happy life. Economic sustainability refers to how a system interacts with the economy at large and whether the

actions being undertaken can be continued indefinitely or will have to be ceased at some point in the future due to financial reasons [13]. Because sustainability has to do with the environment, society, and the economy it is inherently a system level problem and solutions must be defined and evaluated at the systems level.

To meet the requirements of sustainable development, practitioners should utilize a proactive system level perspective rather than a perspective that simply reacts to symptoms [11]. Fiksel reinforces that many sustainability problems are in fact systems problems citing examples in the field of industrial ecology and ecological engineering [11]. This systems approach to sustainable development is similar to the approach taken for many problems in epidemiology. Consider the occurrence of bacterial gastroenteritis in a developing country; the disease can be treated using antibiotics. However, antibiotics do not prevent people from getting bacterial gastroenteritis and may even lead to the emergence of strains of antibiotic resistant bacteria that exacerbate the problem. In this case the use of antibiotics to treat the disease is not a sustainable solution. However, if the root cause of the infections is poor hygiene in the local food processing plant, addressing that issue would be a sustainable solution to this epidemiological problem. In both sustainable systems design and epidemiological problems investigators must account for variability and uncertainty in a system spanning over multiple scales of space and time to generate sustainable solutions [11]. This research hypothesizes machine learning tools currently used in epidemiology, bioinformatics, and other areas for elucidating causal chains through the generation of informative network structure should be applicable to sustainable system design problems.

1.3 Potential for Machine Learning

There is a large amount of open source data available on sociotechnical systems today due to the rise of anonymized public data sharing. There have also been major advances in learning the best approximation of causal models from data in the fields of epidemiology, bioinformatics, operations research, and maintenance planning [8][14][15][16][9][10]. All of this research has resulted in the development of a large number of BN structure learning algorithms and post-processing approaches [17][18][8][19][15][20][21]. The gap that exists in the literature is for a methodology to select a causal structure learning algorithm/post-processing approach given the characteristics of complex observational datasets like those found in sociotechnical systems. To generate such a methodology the approaches that have been independently developed to characterize datasets must be applied to a diverse range of datasets. Different algorithms and post-processing approaches must then be applied to each of those datasets to understand performance given differing characteristics. The results of this characterization and application of algorithms can serve as a guide to how other researchers may characterize their own observational datasets and select an algorithm/post-processing approach.

Commented [MWD15]: Say approximations

1.4 Bike Share Case Study

This research will use the Divvy bike share program in Chicago, Illinois as a motivating case study for testing machine learning in a complex sociotechnical dataset [22]. The Divvy bike share system is dispersed throughout the city of Chicago. Divvy records and makes publicly available the rental data for researchers and data analysts to

assess system operation problems. In 2014 the maximum number of Divvy subscribers riding bikes at any one time was 398 which was only 13.41% of the total capacity of the bike share system. On average if bikes were being used at all it was only 107 bikes or 3.61% of the total capacity of the bike share system [22]. To increase ridership of the Divvy bike share system, designers must understand the network of factors influencing ridership and why some customer segments choose not to ride. A key demographic with unmet needs seems to be women who typically only constitute 24.5% of riders. Data driven network structure learning, with environmental variables and ride data can provide insight into the true causal relations in the system and why some customer segments are not using the service.

Commented [MWD16]: Changed and added a little blurb at end of sentence

Commented [TC17R16]: ok

1.5 Chapter Summary

This chapter has introduced the concept of sociotechnical systems and how they are related to sustainability. It hypothesizes that machine learning tools used to gain insight into epidemiological, biological, and other systems via network structure discovery can be applied to sustainable systems design. The chapter also explains that the current gap in the literature is for a methodology and guide that helps researchers working with complex datasets to select an effective algorithm and post-processing approach. The Divvy bike share program is introduced as an opportunity to apply causal model machine learning approaches to a sociotechnical observational dataset. Data for sociotechnical systems, however, present many challenges for structure learning algorithms. These challenges and the dataset characteristics that result from them will be examined in Chapters 2 and 3. The contribution of the research is to assess existing methods and characterize performance for

researchers trying to use causal network structure learning algorithms on observational datasets.

CHAPTER 2. APPROACHES TO LEARNING CAUSALITY FROM DATA

Given a causal model presented in a visual manner, designers may better see the root causes of issues and system level inefficiencies allowing them to better address issues from a systems level perspective. This systems level approach is vital for developing sustainable sociotechnical systems that regulate themselves rather than relying on constant intervention to maintain equilibrium. However, developing causal models from sociotechnical systems data is difficult in practice. Developing models is difficult because sociotechnical datasets are often small, amalgamated from multiple sources, and contains hidden variables. [These characteristics of sociotechnical data typically mean that the assumptions that underlie causal structure learning are invalid and according to Pearl one cannot definitively say that a discovered connection is causal in nature [23]. When the assumptions are not satisfied the observational data may only be used to “[estimate] the parameters of the causal link once we are absolutely sure of the causal structure, but the structure itself, and especially the directionality of the links, cannot be inferred from the data” [23].]

Commented [MWD18]: Talking about theoretical causal model

2.1 Sociotechnical data

One of the challenges of working with sociotechnical systems, in an effort to make them more sustainable, is that the datasets describing these systems are typically not ideal for many kinds of analysis [24]. This challenge exists because a sociotechnical system is composed of many interacting parts and combining the data from each component part leads to datasets which may not be representative of the system. Additionally, often sociotechnical researchers are forced to use small datasets, which cannot always fully

Commented [WM19]: Added in this section from Pearl's seminal paper.

Commented [TC20R19]: Good

inform the systems behavior under any condition. The small data size issue is only exacerbated when some of the collected data is sparse, meaning that it is missing values. Finally, variables affecting the system that are not captured in the dataset, called latent variables, lead to erroneous conclusions. These issues, which define datasets characteristics, are one of the primary drivers behind this research; to create a guide as to what algorithms and post-processing approaches to use considering certain dataset characteristics.

2.1.1 Small data size

When looking at sociotechnical data researchers are often faced with challenges from small datasets or an absence of data [24]. These challenges are common because information in sociotechnical systems often involves human subjects and their information and data are protected by certain codes of conduct. In this context small datasets are defined as datasets without enough information to fully describe a system in every possible configuration.

These small datasets pose a problem for machine learning and data analysis. When given a small dataset many machine learning and data analysis approaches are prone to an issue called overfitting [25]. The overfitting of a model means that it has formed rules and conclusions that are too specific to the dataset. The machine learning algorithms make these rules because in situations where the algorithm only has one or a few examples they tend towards assuming the given example is the rule rather than an exception. With larger datasets overfitting is not as significant of an issue because the data provide a more representative sample of the system. Small datasets can make statistical inference and

modelling very difficult, this research hopes to find out which machine learning approaches perform best with such small datasets and what kind of post-processing approaches may help avoid the problem of overfitting [25][15][21].

2.1.2 Data Amalgamated from multiple sources

To accurately represent a sociotechnical system one must have data describing the people, technology, and environmental variables [26]. It is rarely the case that these variables are all being recorded simultaneously by one data collection system; more often the observations of the three types of variables are recorded in separate datasets. To make a complete sociotechnical dataset, multiple datasets must be merged. In merging these datasets researchers introduce the possibility of spatial temporal granularity errors.

Granularity of data is the frequency of samples as a function of time (temporal) or the resolution of the data as a function of location (spatial). For example, if a researcher has the GPS signature of a car every ten seconds, the researcher can only say how fast the car was going on average during those ten seconds. If another variable for the car has a sampling frequency of one hertz, combining the two datasets may create a new dataset that is not representative of the original system. That example is an issue with temporal granularity, an issue with spatial granularity might result if one only knows rainfall in one location of a city and must extrapolate that rainfall citywide. Issues that are a result of mismatched granularities pose a challenge to many of the machine learning and data analytic approaches as combined datasets may not be representative of their generating system. Such unrepresentative data violates the Markov Assumption and Faithfulness Assumptions for BNs which are necessary for learning a true causal structure. The Markov

Commented [MWD21]: This case causal is being identified as only in the case that assumptions are valid

Assumption and the Faithfulness Assumption will be covered in Section 3.2.2 and Section 3.2.3 respectively.

2.1.3 Data Sparseness

Data sparseness is when some or all the variables are missing from a particular instance in a sample dataset. This missing information is another issue that arises with sociotechnical datasets. It is often the case that researchers will have a gap in the data collection due to extenuating circumstances. Incomplete samples are often omitted from any final dataset, but when data come from multiple sources the severity of gaps may be exacerbated by the sparseness of each separate dataset. While there are machine learning and data analytic approaches that compensate for sparseness they are often not as effective as their counterparts that use complete data. These compensatory approaches include the utilization of the estimation maximization approach to fill in missing variables based on the values of the complete instances in the sample dataset [27]. While this does provide more training data for learning, it will not provide any unique data points that are not characteristic of the data points already observed.

2.1.4 Latent Variables

Any variable that is not captured in the dataset and which is a cause of one or more variables in the dataset is called a latent variable. These latent variables may have a dramatic effect on the system causing researchers to draw connections between causally unrelated variables that shared a common latent variable. The reason that one might draw a conclusion because of a latent variable is explained in greater detail when the Causal Sufficiency Assumption is discussed in Section 3.2.1. Because sociotechnical systems are

Commented [MWD22]: This is ok

Commented [TC23R22]: yes

open ended they can be affected by a wide range of externalities that act as latent variables if left unrecorded.

2.2 Networks built from data

There are a variety of machine learning approaches that can be used to develop a model or image to convey causality to designers. The goal of this research is to determine how well different algorithms perform given the limitations of sociotechnical data. These imitations can make the three data assumptions needed for causal structure learning invalid. These three assumptions are the Causal Sufficiency Assumption, Markov Assumption, and Faithfulness Assumption and will be covered in Chapter 3.

Commented [MWD24]: Added in

Multiple machine learning approaches exist that could provide some insight into the causal structure of systems. This section describes three of these, feature selection, neural networks, and BNs. BNs are the focus of this work because they are visually informative, can represent a large global structure, and serve as a multivariate regression. Future work may reconsider neural networks which have recently proven effective at learning visually intuitive causal structures from data [28].

Commented [MWD25]: reworded

2.2.1 Feature Selection

Feature selection is a group of machine learning approaches that are oriented towards local discovery; meaning that only the parents, children, and other parents of a shared child variable can be determined [29]. This set of determinable variables is referred to as the local Markov Blanket of a variable and is shown for a variable A in Figure 1.

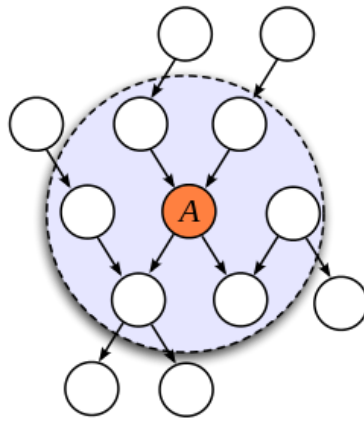


Figure 1: Markov Blanket [30]

Feature selection is a strong candidate for helping to discover and communicate causal relationships to designers; however, these approaches were never geared towards learning a large coherent structure. Without a global structure, designers may not see the full complexity of a sociotechnical system and distinguish root causes and system level inefficiencies. However, some aspects of feature selection algorithms are used in other structure learning approaches which are better suited to understanding the entire structure, namely BNs [31].

Commented [MWD26]: removed

Commented [MWD27]: removed

2.2.2 Neural Networks

Neural networks are a type of machine learning that utilizes layers of operators that act on inputs to generate outputs [32]. Neural networks create predictive models by determining weights corresponding to connections between different layers of a multi-layer network. These weights modify inputs and pass them on to the next layer of the network. If the collective sum of all the incoming messages to a node, in a layer, exceeds the

threshold set for the node it passes information along to the next layer of nodes [32]. The resulting structure of a neural network, shown in Figure 2 has input and output layers of nodes mediated by one or multiple layers of hidden nodes. The hidden nodes, through the training process, learn threshold values and connection weights that turn inputs into the correct outputs[32]. The hidden nodes are usually connected to every other node in the neighbouring layers as shown in Figure 2[32].

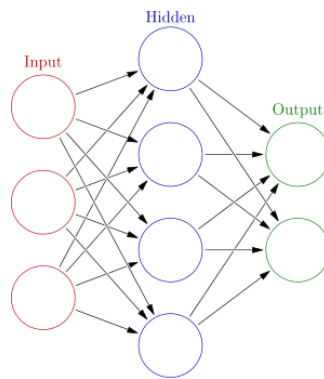


Figure 2: Sample Neural Network [33]

While this approach does learn a model of the system capable of performing predictive inference, it fails to represent the system as a causal graph. As a designer, looking at a neural network is not very informative. The nodes are all connected to one another and any insights that designers might want to gain are lost in the denseness of the network and black box nature of the hidden nodes. This issue of interpretability is only exacerbated with deep learning which is a type of neural network with many layers of hidden nodes [32]. Neural networks learn models that effectively predict outputs, but they do so in a way that is not readily interpreted by designers. However, work has been recently

Commented [MWD28]: talking about theoretical causal graph

done in utilizing deep learning to generate structures that can inform causality separate from the actual neural network structure. This work is very recent and did not influence the work of this thesis [28].

Commented [MWD29]: changed

2.2.3 Bayesian Networks

BNs utilize Bayesian statistics to factor a joint probability distribution into a set of conditional distributions given parent variables. This statement means that the entire joint probability distribution, which is the probability of every possible event as defined by the combination of the values of all the other variables, is reduced such that the probability of an event is only dependent upon a few parental variables. The fact that the distribution is conditional on only some of the other variables leads to a graph with a structure. The structure is represented in form of a Directed Acyclic Graph (DAG) with variables (nodes) connected by arrows (edges). Each node has a conditional probability table that relates it to its parent variables as shown in a sprinkler system network in Figure 3. The structure of these graphs proceeds in a linear fashion from one or multiple nodes to a terminating node without going through the same node twice, hence the name acyclic. This acyclic assumption may not be completely accurate in situations where feedback or feed forward may exist in the real system, but it still provides a model that designers can understand and easily use for insight [34].

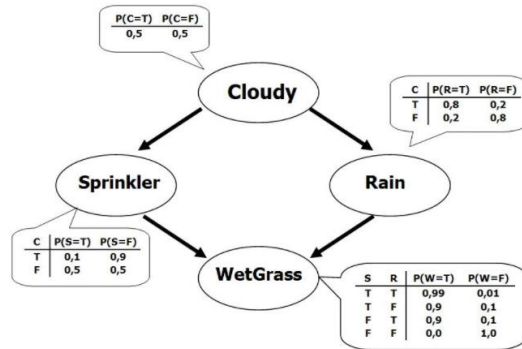


Figure 3: Bayesian Network [35]

BNs are used in this research because the visual representation of the structure can be specified by experts, suggesting that the structure is also interpretable by experts or other decision-makers. BN structure learning algorithms provide an opportunity to automate the construction of networks to provide insight where experts may be unavailable, limited, or biased. System insight from BNs has been utilized in epidemiology and in bioinformatics applications in the past two decades with success[14][15][36]. In the bioinformatics realm this BN structure learning has been successfully used to identify which genes regulate which other genes in datasets with over 800 different genes [37]. Some critics of using BNs for informing causal diagrams argue that the same insights can be obtained by generalized linear modelling [38]. However, the BN serves as a full multivariate regression model, taking into account that all variables may potentially be dependent on one another, as opposed to the multivariable model generated by generalized linear model, which does not assume that all variables may be dependent on one another [38]. BN's visually comprehensible structure, ability to learn global structures, and ability to serve as a multivariate regression are why they were selected for this research [38].

Commented [WM30]: used to have the stance on causality here I removed it and have incorporated it into the beginning now. Still not sure how to put in Dr. Wang's request. I think I have it written out but I cant get it to fit into current frameowkr.

2.3 Chapter Summary

This chapter explains that sociotechnical data is often composed of multiple merged datasets and may be small in size. These are data characteristics that undermine the Markov Assumption and Faithfulness Assumption. The sociotechnical data may also have latent variables which undermines the Causal Sufficiency Assumption. The importance of these three assumptions in structure learning from data will be presented in the next chapter and relates to the ability to recover a causal structure.

Commented [MWD31]: ok

With these limitations on sociotechnical data in mind, the chapter explores a few data driven approaches for learning causal diagrams that can inform causal connections and concludes that BNs are a promising tool. They are a promising tool because unlike other approaches they clearly illustrate causal relations in a global graph that serves as a multivariate regression.

Commented [MWD32]: qualified

Commented [MWD33]: removed

CHAPTER 3. BAYESIAN NETWORK STRUCTURE LEARNING

This chapter covers the basics of BNs and why they are beneficial during the problem formulation portion of the design process, before designers have defined customer needs or a solution approach [5]. This chapter will cover the assumptions underlying learning BNs structure from data and the algorithms that will be used to learn BNs. The process of learning parameters, validating the networks, and a comparison of the strengths of the different algorithms will also be covered. (This)chapter will serve as the framework for understanding the existing literature gap which is that there is no guide for determining a dataset's characteristics a priori and then using those characteristics to select an optimal algorithm and post-processing approach.

3.1 Bayesian Network Concepts

3.1.1 Bayes' Theorem

People typically understand probability as the long term limit of a number of random experiments, or as the strength of a belief in a single event occurring, the first being called the frequentist approach and the second being a Bayesian approach [39]. This thesis primarily takes the Bayesian approach to problems. Central to the Bayesian approach is Bayes theorem which is derived below. The conditional probability of an event E happening given that event F has happened is given by the equation (1).

$$P(E|F) = \frac{P(E \cap F)}{P(F)} \quad (1)$$

Commented [TC34]: always follow the word "this" by a noun.
This what will be a framework?

Commented [WM35R34]: This chapter

As a result of equation (1) it can be seen that equation (2) holds true with a simple rearrangement.

$$P(E \cap F) = P(E|F)P(F) = P(F|E)P(E) \quad (2)$$

Finally, Equation (2) can be rearranged to give equation (3) which is known as Bayes' Formula and is fundamental to Bayesian statistics.

$$P(E|F) = \frac{P(F|E)P(E)}{P(F)} \quad (3)$$

In this equation the term $P(F|E)$ is referred to as the posterior probability of E given F. The term $P(F|E)$ is the likelihood of E given F. The term $P(E)$ is the prior or marginal probability of the event E. Finally, the term $P(F)$ is a normalizing term that expands out as in equation (4).

$$P(F) = \sum_{H_i \in H} P(F \cap H_i) = \sum_{H_i \in H} P(F|H_i)P(H_i) \quad (4)$$

In equation (4) H is a set of pairwise disjoint events such that $H_1 \cup H_2 \cup \dots \cup H_n = \Omega$ and Ω is the entire state space. Given these definitions one can say what it means for two things to be independent. Two events E and F are **independent** if equation (5) holds true.

$$P(E|F) = P(E) \text{ and } P(F|E) = P(F) \quad (5)$$

Following from equation (5) two events are **conditionally independent** given some event

G if equation (6) holds true.

$$P(E|F \cap G) = P(E|G) \text{ and } P(F|E \cap G) = P(F|G) \quad (6)$$

As an example of independence, two normal coins A and B are flipped in sequence. In this case the outcome of coin flip A provides no indication of the outcome of coin flip B and the two are independent of one another. As an example of conditional independence, if the same coin is flipped twice in a row and there is a possibility that the coin is biased toward coming up heads more often than tails then the first flip does provide information on the second flip. That is because the first flip informs as to whether the coin is biased or fair. However, when the coin is known to be fair or biased the outcome of the first flip becomes irrelevant to our predictions of the second flip. Without knowing the coin, a prediction for the second flip is dependent on the observation of the first flip, but they are conditionally independent given knowledge about the fairness of the coin. This work will use the notation $E \perp\!\!\!\perp F|G$ to indicate that event E is conditionally independent of event F given event G under a probability distribution P [31].

A BN is a graphical representation of a joint probability distribution factored over a set of variables $X = \{X_1, X_2, \dots, X_n\}$. This BN consists of a network structure \mathcal{G} and a set of local probability distributions for each of the nodes conditional upon the parental variables. By representing the system in this fashion users can take advantage of the concepts of independence and conditional independence to reduce the complexity of the joint probability distribution. In a case without conditional independence, equation (7)

Commented [WM36]: This is a revised section as per your comments, it is much clearer now with your input

Commented [TC37R36]: good

represents the number of terms in the joint probability distribution describing the relationship of n binary variables.

$$2^{n-1} = \text{number of entries in CPT for binary variables} \quad (7)$$

This equation is exponential, and entries will increase rapidly as more variables are added. To show the importance of independencies start with the identity in equation (8).

$$P(X_1, X_2, \dots, X_n) = P(X_1 | X_2, X_3, \dots, X_n) P(X_2, \dots, X_n) \quad (8)$$

It can then be seen that if $X_1 \perp\!\!\!\perp \{X_3, \dots, X_n\} | X_2$, meaning X_1 is independent of the remaining variables given X_2 the identity transforms into that shown in equation (9).

$$P(X_1, X_2, \dots, X_n) = P(X_1 | X_2) P(X_2, \dots, X_n) \quad (9)$$

The expression with X_1 has become significantly shorter and the joint term has become smaller. This example shows the power of using conditional independence and it is these independencies that define the structure \mathcal{G} of the BN and factor the joint probability distribution. Any time such independence is found the BN graph \mathcal{G} will not have an edge between the independent variables.

The BN is described with a network structure encoding the independencies specified as \mathcal{G} and with conditional probability parameter Θ describing the outcome of each node given its parents in the graph \mathcal{G} . The BN can be completely defined with the

information in the structure \mathcal{G} and the conditional probability parameters Θ . It is the assumption of this work that the independencies in \mathcal{G} can give insight into the causal nature of the system as part of the problem formulation step of the design process, resulting in more sustainable system design.

Commented [MWD38]: qualified to be less strong

3.1.2 D-separation and d-connection

D-separation and d-connectedness are terms that describe how the different variables are related to one another in the terms of independence and conditional independence defined in section 3.1.1. The “d” in d-separation and d-connection represents dependence.

For two variables to be related to one another in a BN requires that there be a path between them in the graph structure \mathcal{G} . A path is any route that follows edges in the graph \mathcal{G} regardless of direction. A path is active if it transmits information between the start and end of the path and is inactive if it does not transmit information. For two variables X and Y to be d-separated means that knowledge about variable X does not give any information about Y. Two variables X and Y are d-separated if there is no active path between them. To be d-connected means that knowing information about one variable X gives information about the other variable Y. When two variables are d-connected there is at least one active path between them. So two variables X and Y may be connected by many paths in a graph, but if all of the paths are inactive then the two variables are d-separated. Conversely, if just one of the paths is active then the variables are d-connected.

To determine if a path is active one must determine that every vertex on the path is active. To define whether a vertex along a path is active or not look at the four situations

in Figure 4 where variable Z in the three variable vertex set X, Y, and Z is empty or unknown.

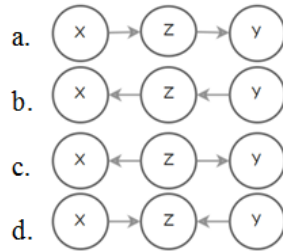


Figure 4: Four Types of Paths [40]

In the first two instances there is a directed path from X to Y and Y to X respectively. If one interprets these edges causally then in the first case X is an indirect cause of Y and in the second case Y is an indirect cause of X. In the third case X and Y are effects of a common cause Z. When Z is unknown all three of these situations give rise to association or dependence between X and Y. This means that all three of these paths are active (d-connected) and knowing information about X will give information about Y. In these three cases the vertex Z is considered active when it is empty or unknown. In the fourth case Z is a common effect of both X and Y, and there is no connection between X and Y. When Z is unknown in the fourth case the path is inactive because knowing information about X will not inform the state of Y. In this fourth case the vertex Z is inactive if it is empty or unknown. However, if Z is known then knowing something about X will give information about Y and the path and vertex Z are active. When a vertex in a path is non-active it is called a collider and blocks information from propagating across the path, when the vertex is active it is referred to as a non-collider.

The paths in a, b, and c are active when the variable Z is unknown and the path in d is active when variable Z is known. The paths in a, b, c, are inactive when the variable Z is known and the path in d is inactive when variable Z is unknown.

Commented [WM39]: added in with the most recent revisions from you ^-^

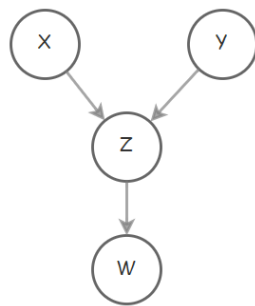


Figure 5: Descendent of Common Effect [40]

The final part of the story involves the decedents of a potential collider. To demonstrate this concept, look at the example network in Figure 5. In this network X and Y are common causes of Z and Z is a cause of W. As demonstrated before if the value of Z is known then X and Y are d-connected, and vertex Z is a non-collider. Consider then that knowing the value of W informs the value of Z and therefor even if the value of Z is unknown it can still be active if the value of W is known. These concepts of d-separation and d-connectedness are fundamental to how many BN structure learning algorithms determine structure. Additionally, in this work these connections are being used to inform a causal understanding of the system which will help designers to generate sustainable solutions.

The concepts of correlated, causally related, d-connected, and d-separated are all central to this thesis. A correlation between two variables means that there is a pairing in

the observation of those two variables. Two variables are causally related when experimentally changing the value of one of the variables (the causal variable) will affect the value of the other (the effect variable). Two variables are d-connected, given the observations, when one can make a better prediction about the value of one of the variables given information about the other. Two variables are d-separated when knowledge of an additional variable makes those two variables independent.

Commented [WM40]: added in with your most recent revisions, again much clearer now.

3.1.3 Markov equivalent structures

Markov equivalent structures can broadly be defined as two structures that represent the same set of independence assertions. More formally, two DAGs are Markov equivalent if and only if they have the same undirected skeleton and the same V-structures. If two graphs \mathcal{G}_1 and \mathcal{G}_2 have the same skeleton (edges without regard to direction) for a set of nodes and have the same set of uncoupled head to head meetings in structures like those demonstrated by variables X, Y and Z in Figure 5 and case (d) in Figure 4 then they will encode the same independence assertion. If these two conditions are met then all of the dependencies within the two Markov equivalent graphs will be the same, meaning the variables will be d-connected and d-separated in the same fashion in both graphs. This concept of Markov equivalence is important because when learning a BN from data some algorithms are incapable of distinguishing between Markov equivalent structures. For an illustration of Markov equivalent structures see Figure 6.

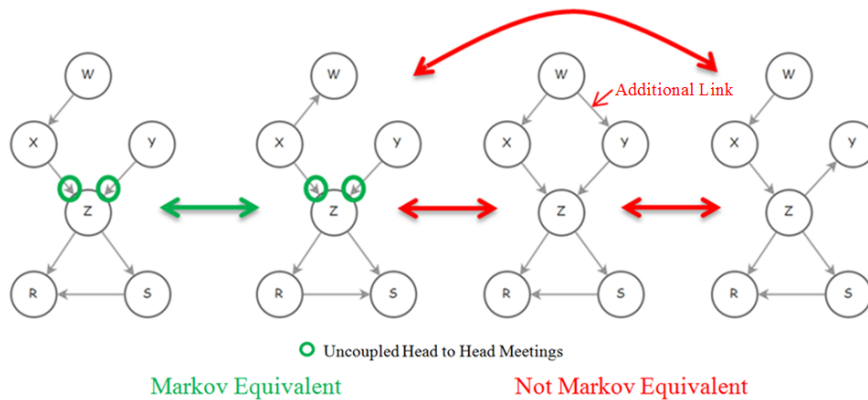


Figure 6: Markov Equivalence [41]

3.2 Assumptions underlying BN Structure Learning

With the terminology introduced in the previous section the Causal Sufficiency Assumption, the Markov Assumption, and the Faithfulness Assumption, all of which are usually made about datasets being used for BN structure learning, can be introduced. These three assumptions about the descriptive datasets must be true for many of the algorithms to be able to learn the true causal graph structure given only the dataset [3]. The Causal Sufficiency Assumption relates to the inclusion of all variables that affect two or more other variables in the current dataset. The Markov Assumption and Faithfulness Assumption relate to how well a dataset represents its underlying causal structure. Reasons why a dataset might not reflect the underlying causal structure include the exclusion of latent variables (and violation of the Causal Sufficiency Assumption) or improper discretization or combination of multiple datasets (causing violation of the Markov Assumption and Faithfulness Assumption) [42].

Commented [MWD41]: added, it is theoretically, not in practice here

3.2.1 Causal Sufficiency Assumption

The Causal Sufficiency Assumption is the most fundamental assumption and implies that all the shared causes of two variables in the BN are recorded in the network. If there is an unobserved variable that is a parent to two observed variables then the observations will show a dependency between the two variables that share the unobserved parent [40]. Without observing the parent, the data indicate a causal effect between the two child variables, even if they are conditionally independent. Because the real cause of these two variables is unobserved algorithms are unable to orient the direction of causality between these two nodes properly. In observational data it is difficult to completely satisfy the Causal Sufficiency Assumption and some have claimed it is impossible to determine the number of potential latent variables [43]. This claim is based on the fact that one can always imagine another variable, typically an endogenous or even exogenous variable at a different level of detail [42]. However, local groups in the graph can have all the necessary variables identified while other far removed parts of the graph do not. Richards showed in his work that without Causal Sufficiency independencies between variables and local regions can be still be determined [40]. Structure learning approaches that relax the Causal Sufficiency Assumption have been developed by many researchers including Pearl and Spirtes and are still able to provide insight such as the Inductive Causation (IC*) and the Fast Causal Inference (FCI) algorithms respectively [44][45]. It is difficult to determine if a network region is causally sufficient, most attempts to determine this are based on Bowden and Turkington's work with instrumental variables [1]. The instrumental variables methods have been able to provide some guidance as to the presence of latent variables for a local area [1][46]. The method used is explained in more detail in the methodology, but

Commented [MWD42]: talking theoretically

Commented [C43]: Give an example.\

WM – gave the IC* and the FCI algorithms as evidence

Commented [TC44]: good

it relies on testing out how a variety of instrumental variables react to detect the presence and relationships of latent variables. However, Judea Pearl has pointed out there are drawbacks to this instrumental variable approach one of which is that it works poorly with nonlinear relationships [46].

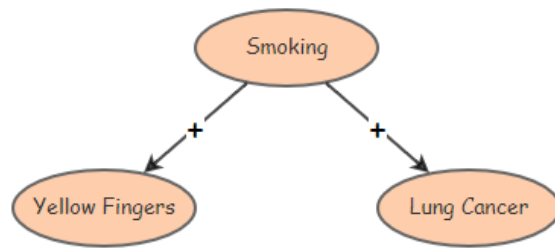


Figure 7: Example of Causal Sufficiency for Smoking Symptoms [40]

The importance of the Causal Sufficiency Assumption can be shown with a simple example in Figure 7. Say that a researcher has a true causal graph describing the variables Smoking (S), Yellow Fingers (Y), and Lung cancer (C). Given a set of observed data the researcher should be able to learn the original graph. However, if the researcher only observes variables (Y) and (C) they might conclude that the two variables are dependent upon one another. When (Y) is true (C) tends to be true as well. That might lead the researcher to believe that there is some causal connection between (Y) and (C). However, the direction of causality would be difficult to determine from data describing the two variables alone [40]. Given data for all three variables the conditional independence of the two becomes apparent and the true structure can be seen.

Commented [MWD45]: theoretical

Commented [MWD46]: theoretical

3.2.2 Markov Assumption

The Markov Assumption for Bayesian networks states that a variable X is conditionally independent of its non-decedents given its parents in a dataset. This assumption relies on all of the appropriate parental variables being included in the dataset and thus the satisfaction of the Causal Sufficiency Assumption [42]. Data satisfy this assumption when the independence relations found by applying d-separation to the true graph \mathcal{G} will be found in the data population describing that graph \mathcal{G} [40]. Given a threshold for independence defined by the researcher (usually 95% confidence level), independence tests may only indicate the correlation that is an artefact of the data or the influence of latent variables [42]. The criterion for independence is critical for resolving a structure out of data that is no longer representative of its generating system due to data amalgamation or a lack of data. If the independence criterion is less stringent, for example a 90% confidence interval, then learning algorithms will discover more independencies in dataset. With a lower threshold the algorithm is more likely to discover the independencies present in the true graph \mathcal{G} as well as others that are not present in the true causal structure. A researcher should be conscious of this relationship and select a lower threshold when they want to be more conservative in the number of causal relationships drawn. Higher thresholds would provide more connections and could be used for identifying more hypotheses for further study. Researchers may test the results of different threshold values on the density of a network and identify a level with a manageable number of connections for verification. In an ideal scenario the algorithms will be run with multiple thresholds and performance will inform which threshold is best [42].

Commented [WM47]: changed section as per the most recent revision

As I type this I am in a smoke filled Internet Café in Huangshan china! Everyone keeps asking me what I am doing and I say 我做工作因为我是大学生! 很忙!

Commented [TC48R47]: I only know the first character.

Commented [C49]: This re-write didn't actually add to the discussion. It just added a context and sort of restated what was said. I've rewritten to explain the logic more.

WM- It is really coming together now yes this is clearer

Commented [WM50]: This is a changed section as is the one below.

As an example, a dataset for the smoking, fingers, and cancer network would not satisfy the Markov assumption if it implied a connection between yellow fingers and lung cancer. In such a case the data is not encoding the independencies found in the true generating network [32]. It is not yet quantitatively possible to know whether a dataset satisfies the Markov Assumption for causal BNs a priori. According to Dawid in his work “Beware the DAG!” the validity of the Markov Assumption cannot be proven a priori without specific domain knowledge since Markov equivalent structures are mathematically indistinguishable [47]. However, some work has been done on domain based criteria for qualitatively judging when the Markov Assumption does or does not hold for a particular expert specified BN structure [2]. This process developed by Lemmer entails trying to understand if the variables currently present in the network can have their interactions understood as a set of signals [2]. This process of decomposing interaction between variables to the signal level was used to evaluate the validity of the Markov Assumption in this work [2]. Lemmer explains that if any interaction can be modeled by real signals the system is complete enough for the Markov Assumption to be valid [2]. This evaluation requires domain specific knowledge for each particular dataset [2].

Commented [WM51]: This is changed section as is the one above

Commented [WM52]: This section was updated as per the most recent feedback from you.

3.2.3 Faithfulness Assumption

The Faithfulness Assumption says that the independence relations obtained by applying d-separation to the true graph \mathcal{G} will hold in the probability distribution of the data as interpreted by the defined independence threshold (usually 95% confidence interval). In this case the population will not imply any independencies that are not present in the true graph \mathcal{G} . While the Markov Assumption says that the population will exhibit all the independencies that would be found by applying d-separation to the true causal graph

Commented [MWD53]: theoretics!

\mathcal{G} it does not imply that all the independencies found within the population will be in the true graph \mathcal{G} . By assuming that a population of data satisfies the Faithfulness Assumption one assumes that only the independencies in the true graph will be discovered in the data. As a result, the more relaxed one is with the definition of independence the more likely they are to find independencies not present in the true graph. Additionally, for the Faithfulness Assumption to be valid the Causal Sufficiency Assumption must be valid and the Markov Assumption must be valid for a dataset [42].

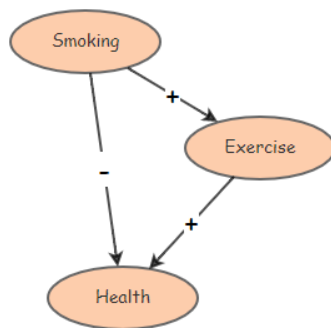


Figure 8: DAG Faithfulness Assumption Figure [40]

For example, in the network in Figure 8 there is a negative relation between health and smoking. However, there is a positive relation between smoking and exercise and there is a positive relation between exercise and health. It is possible that by chance, the effect of smoking through increased exercise and increased health will offset the decrease in health due to smoking. If these effects cancel each other out, then the data may imply independence between smoking and health. This means that a population that implies that independence is unfaithful to the true underlying graph shown in Figure 8. If the threshold

for independence is more lenient then there is a greater chance that independence will be found between the smoking variable and health variable due to chance.

There is a trade-off, as the independence criteria is relaxed the dataset is more likely to imply independencies that are truly there and have a dataset that holds true to the Markov Assumption, but at the same time this relaxation increases the likelihood that the dataset will contain chance independencies that are not present in the true graph \mathcal{G} . This means that the dataset is more likely to imply relations that do not satisfy the Faithfulness Assumption. Normally it is not too farfetched to assume that a dataset that satisfies the Markov Assumption also satisfies the Faithfulness Assumption because graphs are only unfaithful when parameters balance each other out by chance and the thresholds for independence are usually prescribed by the statistical tests accepted nominal value (95% confidence interval) [40]. Scientists tend to believe that nature is not so capricious and that the chances of these parameters balancing out are unlikely with reasonable thresholds for independence. If the definition of independence is adjusted, then there exists a larger probability of these chance cancelations to occur.

The work of Zhang and Spirtes showed that there is theoretically a way to show whether a dataset satisfies the Faithfulness Assumption for *some* causal structure by comparing the result of two algorithms [3]. If the networks learned by the two algorithms, which make different assumptions, are the same then the population is faithful to some graph. Unfortunately, the approach cannot tell what part of a causal graph is faithful or unfaithful [3]. In their work the Markov Assumption is assumed to be valid, which can be proven qualitatively, but has not yet been done quantitatively as discussed in 3.2.2 [2]. The Causal Sufficiency Assumption is also assumed to be completely valid in the theoretical

Commented [MWD54]: theoretical

work by Zhang and Spirtes, which is unlikely for any observational dataset and can only be investigated using the method of instrumental variables based on work by Bowden and Turkington [3][1]. In this work the observational datasets failed to meet the Causal Sufficiency Assumption and Markov Assumption which are prerequisites for the Faithfulness Assumption.]

Commented [WM55]: this section was added with the most recent feedback and speaks to how I assessed the validity of the faithfulness assumption.

Commented [TC56R55]: good

3.3 Structure learning

The task of learning BN's structures from data is NP hard and has to be approached using heuristics in most cases [31]. The reason for this is that the number of possible DAGs for a set of variables n is given by the recursive function in equation (10) where i is the count and $f(n)$ is the function itself [48]. The number of structures possible increases exponentially.

$$f(n) = \sum_{i=1}^n (-1)^{i+1} \binom{n}{i} 2^{i(n-i)} f(n-i) \quad (10)$$

BNs can be constructed by an algorithm, an expert, or combination of experts and algorithms. The structure learning algorithms tend to fall into one of three categories. These categories are search and score algorithms, conditional independence algorithms, and hybrid algorithms that use a mixture of the other two approached [31]. The efficacy of these algorithm classes and the specific algorithms in those classes is closely linked to the validity of the three assumptions and has been discussed by Acid, Campos, Tsamardinos, Heckerman, and many others [49][50][16][8]. Additionally, the efficacy of post-processing

techniques in light of the validity of those three assumptions has been investigated by a number of researchers using observational data [15][14][36][21]

3.3.1 Expert built networks

Historically networks describing complex systems have been built by experts in the field [51]. In BNs used for medical diagnostics the initial structure is specified by doctors. The networks parameters are sometimes supplemented using data from the real world and sometimes defined by subject matter experts [51]. Relying on experts for modeling sociotechnical systems is challenging because there are not as many experts available for new complex systems.

Equations can also facilitate expert network construction by expressing how the variables relate and what the graphical structure should look like. In this work equations were utilized to help inform the structure of one of the investigated networks. If equations do not exist then learning a network must rely on expert intuition and the findings of other researchers. This approach was also utilized in this thesis. The expert process for generating a BN structure consisted of reading previous research on variable interactions, formulating a base network from the previous research and intuition gained through first hand observation, and validating that network by testing the predictive ability of the network on testing data.

(When) a network is cyclic, it must be made acyclic to be a BN. Feedback loops in system models drastically increase the number of possible structures. For simplicity, experts constructing graphs for BNs must determine the most realistic feedforward representation of the system behavior by trimming connections that create cyclic loops.

Commented [WM57]: CASS - This response was supposed to be included when introducing Bryan's graph and to briefly note where cyclic loops were made acyclic and how that decision was made.

With the variables we have no feedback, why is it still useful. -WM - it is an instantaneous snapshot of system which still informs causality.

WM - I included it in the section about building an expert graph, as I think the more general discussion is better suited for being there. And I added a section in the explanation of Bryans graph as well.

Trimming these loops is undesirable, but effects of this feedback can still be captured by dynamic BNs which are a type of BN that represents a time series by feeding outputs from one BN into a subsequent BN. These dynamic BNs can capture feedback and evolve with the system over time[52].

Networks based off equations and the insights of research into systems provides a foundation upon which automated structure learning can help build. The data driven approaches can help to facilitate the refinement of expert models and check biases of experts to come up with the best network possible. A combined expert and data-based approach will encompass the reality found in the data with the creative insight of human experts.

3.3.2 *Search and score*

Search and score algorithms are a data driven approach that determine the best BN structure by using a scoring metric to compare alternative networks. The algorithms build and score the potential networks by adding, removing, and reversing the direction of edges between nodes in the graph, starting with an empty graph [31]. As these edges are modified the resulting scores are compared and a modification is kept if it improves the score. The searches proceed in a greedy hill climbing approach and some make use of random restarts to avoid local maxima. The scoring function used to compare the networks may be chosen from one of two general categories; Bayesian scores and information theoretic scores.

Commented [WM58]: This section is an added in section revised as per your last comments

Commented [TC59R58]: ok

3.3.2.1 Bayesian Scores

Bayesian scoring metrics utilize the conjugate prior of the multinomial distribution which is the Dirichlet distribution with hyperpriors to indicate the number of pseudocounts or confidence in a prior belief. The number of discrete states the finite random variable X_i can take is represented by r_i . The number of possible parent configurations of a set \prod_{X_i} of X_i is represented by q_i . A single configuration of \prod_{X_i} is represented by w_{ij} where $(1 < j < q_i)$. N_{ijk} is the number of instances in the data \mathcal{D} where the variable X_i takes its k -th value (where k is the value of the variable X_i) x_{ik} and the variables in \prod_{X_i} take their j -th configuration w_{ij} . N_{ij} is the number of instances in the data \mathcal{D} where the variables in \prod_{X_i} take their j -th configuration w_{ij} as in equation (11). The total number of instances in the data \mathcal{D} is N .

$$N_{ij} = \sum_{k=1}^{r_i} N_{ijk} \quad (11)$$

The Dirichlet function which is the conjugate prior of the multinomial function is used to represent the prior probability of the network B given data \mathcal{D} . In equation (12) the Γ represents the gamma function.

$$P(B, \mathcal{D}) = P(B) \times \prod_{i=1}^n \prod_{j=1}^{q_i} \left(\frac{\Gamma(N'_{ij})}{\Gamma(N_{ij} + N'_{ij})} \times \prod_{k=1}^{r_i} \frac{\Gamma(N_{ijk} + N'_{ijk})}{\Gamma(N'_{ijk})} \right) \quad (12)$$

By taking the log of equation (12) one arrives at the Bayesian Dirichlet (BD) score which is defined by equation (13).

$$BD(B, \mathcal{D}) = \log(P(B)) + \sum_{i=1}^n \sum_{j=1}^{q_i} \left(\log \left(\frac{\Gamma(N'_{ij})}{\Gamma(N_{ij} + N'_{ij})} \right) + \sum_{k=1}^{r_i} \log \left(\frac{\Gamma(N_{ijk} + N'_{ijk})}{\Gamma(N'_{ijk})} \right) \right) \quad (13)$$

Unfortunately, equation (13) is not very practical because defining a hyperprior N'_{ijk} for all i, j , and k is not realistic [53]. So the BD score is not usable in practice, but there are some versions of the BD score which are usable. The benefit of these scores is that they assign a value to a network configuration based on how well network B encodes data \mathcal{D} .

$$K2(B, \mathcal{D}) = \log(P(B)) + \sum_{i=1}^n \sum_{j=1}^{q_i} \left(\log \left(\frac{(r_i - 1)!}{(N_{ij} + r_i - 1)!} \right) + \sum_{k=1}^{r_i} \log(N_{ijk}!) \right) \quad (14)$$

Equation (14) is a special case of the BD score and is called the K2 score, it was introduced alongside the K2 algorithm by Cooper and Herskovits. The K2 algorithm utilizes this special scoring function which is a BD scoring metric in which the hyperprior indicating pseudo-counts is set to equal one [54]. In this case it is assumed no information is coming from the priors [54]. It differs from the BD score in that $\Gamma(c) = (c - 1)!$ when c is an integer resulting in the $(r_i - 1)!$ term seen.

Another special case of the BD score that will be utilized in this work is the BDeu score where the “u” stands for uniform joint distribution and the “e” stands for likelihood-equivalence [54]. In this case the prior network assigns a uniform probability to each configuration of $\{X_i\} \cup \prod_{X_i}$ given the complete DAG \mathcal{G} as expressed by equation (15).

$$P(X_i = x_{ik}, \prod_{X_i} = w_{ij} | G) = \frac{1}{r_i q_i} \quad (15)$$

In the BDeu score the score only depends on one parameter, the equivalent sample size or pseudocount N' .

$$BDeu(B, \mathcal{D}) = \log(P(B)) + \sum_{i=1}^n \sum_{j=1}^{q_i} \left(\log \left(\frac{\Gamma \left(\frac{N'}{q_i} \right)}{\Gamma \left(N_{ij} + \frac{N'}{q_i} \right)} \right) + \sum_{k=1}^{r_i} \log \left(\frac{\Gamma \left(N_{ijk} + \frac{N'}{r_i q_i} \right)}{\Gamma \left(\frac{N'}{r_i q_i} \right)} \right) \right) \quad (16)$$

This parameter N' expresses the strength of the prior belief in the uniformity of the conditional distributions of the network. The value of N' is something that can be adjusted and will impact the results of the search algorithms [54]. The choice of which Bayesian score and which hyperprior to use will influence the density of the network and in practice multiple values for the hyperprior are often attempted [54]. One difficulty noted with the Bayesian scores (aside from K2) is a parameter or parameters must be chosen. The information theoretic scores do not require the user to pick a parameter.

3.3.2.2 Information Theoretic Scores

The basis for all of the information theoretic scores is the log likelihood score as it is the simplest approach and consists of minimizing the information content of a data set \mathcal{D} by a BN B [54]. The log likelihood score is shown in equation (17). The log likelihood score is prone to overfitting as there is no penalty for adding an edge between nodes in the BN B . Any time there is any evidence to suggest a relationship the log likelihood score of a network with an edge representing that relationship will score higher.

$$LL(B|\mathcal{D}) = \sum_{i=1}^n \sum_{j=1}^{q_i} \sum_{k=1}^{r_i} N_{ijk} \log \left(\frac{N_{ijk}}{N_{ij}} \right) \quad (17)$$

To avoid the issue of overfitting that comes with the log likelihood score researchers have tried limiting the number of parent variables that each node can have in the network B and have tried imposing penalty functions for adding additional edges. The second approach has given rise to the Minimum Description Length (MDL) or Bayesian Information Criterion (BIC) score for Bayesian networks which penalizes the log likelihood score as shown in equation (18).

$$BIC(B|\mathcal{D}) = MDL(B|\mathcal{D}) = LL(B|\mathcal{D}) - \frac{1}{2} \log(N) |B| \quad (18)$$

In this penalty function the number of observations represented by N and a factor $|B|$ which denotes the network complexity and is shown in equation (19) make up the penalty function.

$$|B| = \sum_{i=1}^n (r_i - 1) q_i \quad (19)$$

The penalty for the complexity is based on the number of parameters that is present in the conditional probability tables describing each node and as you have a denser network the number of parameters will increase. In the equation (19) the r_i represents the number of discrete states and the q_i represents the number of parental variables. n serves as a counter equal to the number of variables. As the number of observed samples increases the penalty

for complexity becomes less and less. This is designed to allow for more complex structures to be considered as more evidence of edges is attained.

$$AIC(B|\mathcal{D}) = LL(B|\mathcal{D}) - |B| \quad (20)$$

Another commonly used scoring function in the information theoretic realm is the AIC, seen in equation (20). It is similar to the BIC in that it penalizes the log likelihood score based on the complexity of the network as measured by the number of parameters $|B|$. However, the term that accounts for the number of samples is not included so the penalty will be less detrimental at small sample sizes and will become more detrimental than the BIC penalty at larger sample sizes. It should be noted that in both cases as the number of samples increases to infinity the BIC and AIC scores will both be the same as the log likelihood score [54]. The information theoretic scoring metrics are not dependent on a user specified variable (hyper prior in the previous case). The choice of which information theoretic scoring metric affects graph density as related to training sample size.

3.3.2.3 K2 Algorithm

The K2 algorithm is a search and score algorithm and was built on the Kabutu algorithm prior to it [31]. It operates by optimizing the score of a network over an ordering of the variables [31]. The K2 algorithm utilizes its own scoring function score which is a Bayesian scoring metric in which the hyperprior indicating a pseudocount set to equal one [54]. In this case it is assumed no information is coming from the priors. The K2 scoring function for each individual node is shown in equation (21) where the variables are the

same as described previously in section 3.3.2.1 for the K2 score and Pa^{X_i} is a set of proposed parents for variable X_i .

$$g(i, Pa^{X_i}) = \prod_{j=1}^{q_i} \frac{(r_i - 1)!}{(N_{ij} + r_i - 1)!} \prod_{k=1}^{r_i} N_{ijk}! \quad (21)$$

The algorithm uses this score to evaluate the addition of each new edge as it marches through the variable ordering. Since this algorithm is given a sequence of variables it does not have to explore all possible structures represented by equation (10) but rather only a subset of those represented by equation (22) where n is the number of variables [55].

$$\text{number of DAGs} = 2^{n(n-1)/2} \quad (22)$$

Algorithm 1: K2

```

1: procedure K2;
2: {Input: A set of  $n$  nodes, an ordering on the nodes, an upper
   Bound  $u$  on the number of parents a node may have,
   And a database  $\mathcal{D}$  containing  $m$  cases.}
3: {Output: For each node, a printout of the parents of the node}
4: for  $i := 1$  to  $n$  do
5:    $Pa^{X_i} := \emptyset$ ;
6:    $Pold := g(i, Pa^{X_i})$ ; {use equation (21)}
7:   OKToProceed := true
8:   while OKToProceed and  $|Pa^{X_i}| < u$  do
9:     let  $z$  be the node in Predecessor ( $X_i$ ) –  $Pa^{X_i}$  that maximizes
        $g(i, Pa^{X_i} \cup \{z\})$ ;
10:     $Pnew := g(i, Pa^{X_i} \cup \{z\})$ ;
11:    if  $Pnew > Pold$  then
12:       $Pold := Pnew$ ;
13:       $Pa^{X_i} := Pa^{X_i} \cup \{z\}$ 
14:    else OKToProceed := false;
15:  end {while};
16:  write ('node:',  $x_i$ , 'Parents of this node:',  $Pa^{X_i}$ )
17: end {for};
18: end {K2}; [55]
```

3.3.3 Conditional Independence

Conditional independence based algorithms work by starting with a fully connected graph and then removing edges based on conditional independence tests [31]. These conditional independence tests can be based on mutual information (MI), G2 tests, T-tests, or any other metric that a researcher defines [31]. Typically these values work with the assumption of independence with a 95% confidence interval [40].

It is believed by many in the field of BN structure learning that the conditional independence tests are better for elucidating causality in systems, because they contain explicit tests for independence [56][31][16][49]. However, the conditional independence approaches also have some weaknesses as shown in Table 1. The conditional independence approaches are weaker when utilized with small datasets and will not consider any structure that contains a dependence it has ruled out, even if that dependence improves the overall graph. There is still a debate as to whether CI or the search and score methods are more effective for understanding observational data; this thesis supports the idea that utilization of explicit tests of independence is better for determining diagrams that inform causality from observational data.

Commented [WM60]: Added section to clarify that these thresholds are more or less standardized and uses typical statistical tests.

Commented [TC61R60]: ok

Commented [MWD62]: Changed the wording

Table 1: Algorithms Types, Strengths, and Weaknesses

Sociotechnical Data Characteristic	Search and Score	Conditional Independence	Source
Small/Sparse data sets	Stronger	Weaker	[31]
Causal Relationships	Weaker	Stronger	[49]
Robustness of model (noise from multi data)	Stronger	Weaker	[42]
Detecting Latent Variables	Weaker	Stronger	[49][20]

3.3.3.1 PC Algorithm

The PC algorithm is a conditional independence algorithm named after the first names of its inventors Peter Spirtes and Clark Glymour [57]. This algorithm was a successor to the SGS algorithm and is statistically consistent with SGS except it is computationally more efficient [57]. The SGS algorithm works by first generating a complete undirected graph C on the vertex set V . SGS operates by seeing if for each pair of X and Y and each set of all other variables S , if $X \perp\!\!\!\perp Y | S$; if so, remove the edge between X and Y . The SGS algorithm then checks for colliders by looking for conditional independence and orienting edges accordingly. The remaining un-oriented edges are oriented to be consistent with the already oriented edges where possible, when it is not possible to orient the edge it is indicated as being bidirectional [18]. These bidirectional edges pose a problem in learning the parameters Θ for a given graph \mathcal{G} and will be oriented according to the same variable ordering given to the K2 algorithm to circumvent this issue. The PC algorithm improves on the methodology of the SGS algorithm by drastically decreasing the number of statistical tests needed. The PC algorithm by looking for the first instance that proves that X and Y are independent and then stops performing tests on the remaining variables in set S [18].

Algorithm 2: PC

- 1: **Data:** Takes a dataset as input
- 2: Form the complete undirected graph C on the vertex set V .
- 3: $n = 0$.
- 4: **repeat**
- 5: **repeat**
 select an ordered pair of variables X and Y that are adjacent in C such that $\text{Adjacencies}(C, X) \setminus Y$ has cardinality greater than or equal to n , and a subset S of $\text{adjacencies}(C, X) \setminus Y$ of cardinality n , and if X and Y

- are d-separated given S delete edge $X - Y$ from C and record S in $\text{Sepset}(X, Y)$ and $\text{Sepset}(Y, X)$;
- 6: **until** All ordered pairs of adjacent variables X and Y such that $\text{Adjacencies}(C, X) \setminus Y$ has cardinality greater than or equal to n and all subsets S of $\text{Adjacencies}(C, X) \setminus Y$ of cardinality n have been tested for d-separation;
- 7: $n = n + 1$;
- 8: **until** For each ordered pair of adjacent vertices X, Y , $\text{Adjacencies}(C, X) \setminus Y$ is of cardinality less than n ; for each triple of vertices X, Y, Z such that the pair X, Y and the pair Y, Z are each adjacent in C but the pair X, Z are not adjacent in C , orient $X - Y - Z$ as $X \rightarrow Y \leftarrow Z$ if and only if Y is not in $\text{Sepset}(X, Z)$
- 9: **repeat**
 If $A \rightarrow B$, B and C are adjacent, A and C are not adjacent, and there is no arrowhead at B , then orient $B - C$ as $B \rightarrow C$.
 if there is a directed path from A to B , and an edge between A and B , then orient $A - B$ as $A \rightarrow B$.
- 10: **until** no more edges can be oriented;
- 11: **Result:** Returns the pattern [58]

3.3.4 Hybrid approaches

Hybrid approaches utilize some of the concepts explored in the conditional independence based algorithms and some of the concepts from the search and score approach. Typically the conditional independence tests are used to determine local structures, similar to what is done in feature selection [29]. Once the local structures are found the independent blankets surrounding each variable are knitted together using a search and score approach that attempts to optimize the score.

3.3.4.1 MMHC

The Min Max Hill Climbing algorithm was developed by Laura Brown and Ioannis Tsamardinos, it utilizes a twostep process [17]. This process first looks for candidate parent variables for each of the nodes in the network using the MMPC algorithm, which is a

conditional independence based search [59]. The MMPC algorithm operates similar to the PC algorithm in that it determines candidate parents by looking for independence between two variables conditioned on all the other variables in the network.

After discovering the candidate parents, the partial Markov blankets discovered (consisting only of parents) for each node are assembled using a search and score approach that unifies the local structures into a single global structure with the best score. This exploration done by removing, reversing and adding edges between nodes of the network, but the add edge operation is only used when it creates a parent within the candidate parent set found by the MMPC algorithm [17].

Algorithm 3: MMHC Algorithm

```
1: procedure MMHC( $\mathcal{D}$ )
   Input: data  $\mathcal{D}$ 
   Output: a DAG on the variables in  $\mathcal{D}$ 
   % Restrict
2:   for every variable  $X \in V$  do
3:      $PCX = MMPC(X, \mathcal{D})$ 
4:   end for
   % Search
5:   Starting from an empty graph perform Greedy
   Hill-Climbing with operators add-edge, delete edge
   Reverse-edge. Only try operator add-edge  $Y \rightarrow X$  if
    $Y \in PCX$ .
6:   Return the highest scoring DAG found
7: end procedure [50]
```

3.3.4.2 SCA

The Sparse Candidate Algorithm was originally developed by Nir Friedman for use in discovering the regulatory genes in bioinformatics applications [60]. It limits the search space of possible networks by limiting the number of parents each variable can have by a parameter k . This limitation is realistic when looking at the type of regulatory gene

networks that the algorithm was developed for. In the applications the SCA was originally made for the number of variables is sometimes in the thousands and limiting the number of parents is the only way to make the problem computationally feasible.

Algorithm 4: SCA

- 1: **procedure** SCA
- 2: Input: data set \mathcal{D} , initial network B_0 , a decomposable $\text{Score}(B \mid \mathcal{D}) = \sum_i \text{Score}(X_i \mid \text{Pa}^B(X_i), \mathcal{D})$, parameter k
- 3: Output: A network B
- 4: **loop for** $n = 1, 2, \dots$ until convergence
- 5: %Restrict
- 6: Based on \mathcal{D} and B_{n-1} , select for each variable X_i a set $C_i^n (|C_i^n| \leq k)$ of candidate parents. This defines a Directed graph $H_n = (X, E)$ where $E = \{X_i \rightarrow X_j \mid \forall i, j, X_i \in C_j^n\}$
- 7: %Maximize
- 8: Find network $B_n = (\mathcal{G}_n, \Theta_n)$ maximizing $\text{Score}(B_n \mid \mathcal{D})$ among networks that satisfy $\mathcal{G}_n \subset H_n$ (i.e., $\forall X_i, \text{Pa}^{\mathcal{G}_n}(X_i) \subseteq C_i^n$)
- 9: **return** B_n
- 10: **end procedure** [8]

3.3.5 Exhaustive searches

Exhaustive search protocols exist and can be utilized when the number of variables in the network is below 33 [19]. These exhaustive searches determine the highest scoring network as determined by an information theoretic or Bayesian score. The reason that these algorithms were not utilized for this research is that the research aims to determine which algorithm is capable of learning a valid structure with a number of variables in excess of 32. Additionally, the research does not want to limit itself to only score based approaches, which one can argue the exhaustive approaches should be categorized as. One especially notable exhaustive search algorithm was developed by Tomi Silander in 2006 and relies on the decomposability of the scoring function to rapidly search the space of possible structures for the best scoring structure [19].

Commented [MWD63]: deleted

3.3.6 Averaging BNs

There exists a substantial amount of work devoted to the idea of averaging a number of learned BNs as an alternative to the selection of a single *best* structure. Much of the work done in averaging BNs is to avoid the problem of overfitting when only small datasets are available [31]. There are a number of approaches for averaging BNs including Additive Bayesian Networks, averaging over graphs learned with data subsamples, and averaging across the space of different graphs learned with different variable orderings [36][14][15][61][62][21]. It is an open question as to which averaging approach obtains the best results, especially when datasets that do not fully satisfy all of the assumptions for learning causal networks are being used [31]. This thesis investigates the performance of averaging over graphs learned with data subsamples, similar but distinct from what was done in Liu, 2007 [62].

Commented [MWD64]: theoretical

3.4 Parameter Learning

Parameter learning is where data is used to inform the conditional probability tables of the different nodes in a BN and can be learned in a number of ways. This parameter learning can only proceed once a graph structure \mathcal{G} has been learned. The two most common approaches for learning parameters Θ are maximum likelihood estimate (MLE) and Bayesian inference.

3.4.1 Maximum Likelihood Estimate

The MLE approach was used in this work, it estimates the values of the parameters Θ in the learned graphs \mathcal{G} to complete the BN. The MLE method uses the observations to find parameter values that maximize the likelihood of making the given observations and falls within a frequentist interpretation of probability. The MLE approach also necessitates that the graph \mathcal{G} be acyclic in nature. This requirement is why the results of the PC algorithm must be oriented. The MLE learns in such a manner that if a particular event is never seen to occur it is given a probability of zero. This zero estimate could prove disadvantageous for the parameter estimation of a learned graph \mathcal{G} , however to align with the methods of other researchers the MLE approach was used in this thesis [63].

3.4.2 Bayesian approach

The Bayesian approach to parameter estimation says that parameters are random variables having some known distribution. The best parameters are obtained by estimating them given the data. This falls under a Bayesian view of statistics as opposed to the frequentist approach and in this case the probability of an event that is not observed will not be set to zero. The benefits of this prior based approach to parameter estimation are noted by the authors, but to reproduce the methodology of other researchers this research will utilize the MLE approach.

3.5 Predictive Inference

Predictive inference is one of the end use of BNs and allows for predictions to be made given data about a new situation. The benefit of seeing these predictions is that one can determine how a change to a system, in this case a sociotechnical one, might affect the system. This would mean that you could run experiments in the form of simulated

inference. To use the bike share example, one might be able to see how changing the potential price of the bike share would change utilization of the system and use that as a guide to plan out expansion and predict potential revenue. Additionally, one can use the inference ability to make determinations about attributes of users that one might not have. In the Chicago bike share program there are monthly subscribers whose age and gender are known and there are daily users whose age and gender are not known. The subscriber to daily rider ratio is about 60% to 40% respectively. That means that a BN learned to describe the relation of gender and age in the subscriber system could be used to figure out the gender and age profiles of the daily riders for whom the researchers did not have data. This would be predicated on the assumption that the underlying causal network describing both types of riders is the same.

Commented [MWD65]: theoretical

3.5.1 Junction Tree Algorithm

The junction tree algorithm is a general algorithmic framework that calculates the conditional probability of a node given another set of nodes in a BN. The junction tree algorithm was used to test the inference accuracy of the learned BNs because the computational tool used, Bayes Net Toolbox, indicated that it was the best approach [64]. The junction tree algorithm operates by first moralizing the DAG to make it undirected. The algorithm then introduces the evidence that is available. The graph is then triangulated to make it chordal. With the chordal graph the junction tree can be constructed and finally the probabilities can be propagated along the junction tree. A more detailed explanation of the algorithm and underlying mathematical basis can be found in the references [65]. The junction tree algorithm will be utilized throughout this research to perform any predictive inference using a BN.

3.6 Validation of network

BNs are typically used for making inferences and for providing insight into the system through their structure. Researchers tend to benchmark learned BNs by evaluating their ability to perform accurate inference and learn a true or expert specified BN structure.

3.6.1 *Predictive Inference Accuracy*

The accuracy of inferences performed by the learned BN structures and parameters is evaluated using the holdout method. Some of the data describing each of the networks that are being investigated is set aside as the testing set. The remaining data is then used as the training set to learn a network structure and to estimate parameters for that network structure. The junction tree algorithm is then used with the complete BN to guess the value of variables in the testing data.

3.6.2 *Causal match with Expert networks*

This research is also interested in the insights that can be gained from the visual representation of the BNs and seeks to see how learned BNs compare with the true causal network or the causal network deemed most likely by experts. To do this the research needs to utilize a metric that qualifies a distance between two networks. The best way to find such a distance is to compare the number and direction of edges in two BN's structures. This distance gives an idea of the difference between the variable relations learned and true/most likely network relationship.

Commented [MWD66]: theoretical

3.7 Chapter Summary

This chapter has introduced the concept of Bayes theorem and the concept of a BN, which can be specified by an expert or learned from data. Furthermore, this chapter has explained that a multitude of algorithms exists for learning BN structure from data and that these algorithm's performance is closely tied to the validity of three assumptions about training datasets, namely the Causal Sufficiency Assumption, the Markov Assumption, and the Faithfulness Assumption. The structure learning algorithms were classified by those that use a score based approach in an additive fashion or conditional independence methods which operate in a subtractive fashion. Hybrid algorithms which utilize aspects of both approaches also exist. The chapter explains how the learned BN structures can then be populated with parameters and used for inference. Finally, the topic of validating the structure and predictive ability of the BNs learned was reviewed.

CHAPTER 4. DATASETS AND METHODOLOGY

The objective of this research is to evaluate the use of BN learning to find structures that can inform the causal connections in sociotechnical system data. BN learning is primarily limited by how well data satisfy three assumptions underpinning the learning algorithms: the Causal Sufficiency Assumption, Markov Assumption, and Faithfulness Assumption [42]. The validity of these three assumptions was determined generating a subjective estimate of each for the observational datasets using the work of Bowden/Turkington, Lemmer, and Zhang/Spirtes [1][2][3]. The research method than applied different algorithms/post-processing to three different datasets exhibited three different levels of certainty about the assumptions: a simulated ALARM data set that has a known solution and satisfies the assumptions, an on board diagnostics (OBD) data set that generally follows physical laws and has nominal random variables, and the Divvy bicycle rental data that has less predictability and few physical laws governing relationships. These three data sets allow for examination of how data characteristics change the usefulness of the BN learning algorithm and post-processing outcomes.

Usefulness of the algorithms is based on how well they can predict variables in the systems and how well the structures match an expert specified structure. The approach to creating networks and evaluating their predictability and structures has five distinct steps. These steps, shown in Figure 9, are pre-processing/characterizing data \mathcal{D} , using that data \mathcal{D} to learn a BN structure \mathcal{G} , using the data \mathcal{D} and structure \mathcal{G} to learn the BN's parameters Θ , evaluating the ability of the BN to perform inference on a set of holdout data, and comparing the BN structure to an expert's intuition.

Commented [MWD67]: added to be less strong

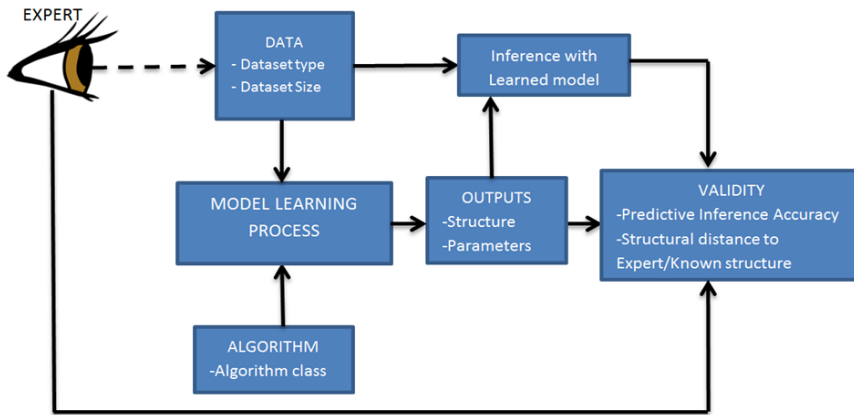


Figure 9: Process Diagram

4.1 Data

The datasets utilized in the research were selected such that each dataset satisfies a different number of the three assumptions, the Causal Sufficiency Assumption, Markov Assumption, and Faithfulness Assumption, underlying BN structure learning [42]. Subjective estimates were used to evaluate the number of assumptions that are valid for the observational datasets. This subjective estimate was based off the work of Bowden/Turkington for the Causal Sufficiency Assumption, Lemmer for the Markov Assumption, and Zhang/Spirtes for the Faithfulness Assumption. The validity of the assumptions was evaluated in the previously indicated order as each of the assumptions is a prerequisite for the next. The Causal Sufficiency Assumption was evaluated in light of the idea of instrumental variables, which can indicate the presence and relationships of latent variables [1][46][42]. By seeing which instrumental variables react to the presence of latent variables, the latent variable’s domain can be discovered [46]. If regions of the graph are shown to be causally sufficient then the method developed by Lemmer can be

used to evaluate if the system satisfies the Markov Assumption [2]. In this method the system is evaluated to see if all the interactions between variables can be modeled as signals, in which case the Markov Assumption can be appropriate. Should the dataset be Causally Sufficient and the Markov Assumption holds true methodology outlined by Zhang/Spirtes can be used to evaluate if the dataset is Faithful or not[3].

By indicating a methodology to characterize the number of valid assumptions and then evaluating the quality of networks learned with different characteristic datasets this research can serve as a guide. The quality of the BN structures is evaluated on the networks ability to perform predictive inference on a holdout dataset and the ability to recover the known or expert specified network structure.

The variable ordering is provided for each of the datasets because some of the algorithms required this ordering as an input (K2) or in a post processing step (PC) to ensure the resulting graphs are acyclic. The ordering was taken from the true structure or was built off the equations governing the system and logical reasoning about what events can causally precede other events.

4.1.1 ALARM Data

The ALARM network has 37 variables corresponding to 37 nodes in the DAG and describes alarms on in an intensive care unit (ICU) life monitoring system [51]. The ALARM data set is a standard test case for algorithm development and is shown in Figure 10. Data are generated by sampling the network and thus the datasets satisfy all three assumptions for learning a causal graph because no latent variables will be present (Causal

Sufficiency Assumption), and independencies will match those in the DAG (Markov Assumption and Faithfulness Assumption) [42].

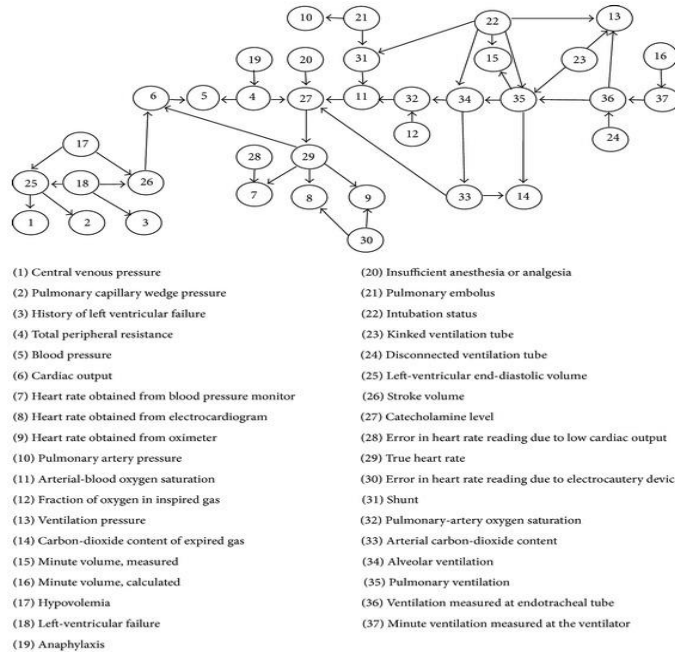


Figure 10: ALARM Network [66]

4.1.1.1 Variables

There are thirty-seven variables within the ALARM network and they can be roughly grouped into three tiers. The top most tier is reserved for eight diagnosis which include liver failure, pulmonary embolism, insufficient anaesthesia, etc. The second tier of variables includes thirteen intermediate variables that may or may not be directly observed

and are reactions to the diagnosis tier. The third tier is composed of sixteen findings variables which correspond to alarms on the ICU system and are triggered by the intermediate tier variables [51]. Due to the number of variables in the system this research will not explore each variable tier in depth. The BN's structure and the variables conditional probabilities were specified by medical practitioners and are not necessarily built from data [51]. For the ALARM network the ordering across the variables is known from the true structure.

4.1.2 On Board Diagnostic (OBD) Data

The on board diagnostic (OBD) data used in this research was collected from one vehicle moving around the Singapore metropolitan area. This system represents a dataset generated from a single mechanical system obeying governing equations and tests the algorithms' abilities to discover a structure that is in line with the known governing equations. The dataset consisted of 242,951 instances, representing 180 trips over the course of 31 days.

The Causal Sufficiency Assumption was first evaluated using a qualitative method built around instrumental variables which are discussed by Bowden/Turkington and Pearl [1][46]. The results of this evaluation showed that some portions of the graph namely the variables of Throttle, Airflow, and RPM seemed to be causally sufficient since it was difficult to imagine any instrumental variable which could change the relationship between those variables by shielding some latent variable. With a portion of the causal graph deemed to be causally sufficient as per Bowden/Turkington and Pearl, the graph was evaluated with the process outlined by Lemmer. The test from Lemmer said that the OBD

dataset did not satisfy the Markov Assumption, because the interaction of all of the variables could not be linked by direct signals [2]. The evaluation of Faithfulness of the dataset as per Zhang/Spirtes was not done because the prerequisite assumptions were not satisfied. Because the three assumptions are not valid the graph cannot be interpreted in a causal manner, but may be used to supplement and direct qualitative and experimental methods for elucidating causal connections [7].

Commented [MWD68]: added

An expert network and variable ordering were built for this mechanical system utilizing equations from ISO 15031-5. The equations utilized can be found in APPENDIX A. OBD GOVERNING EQUATIONS. The expert network derived from these equations is represented in Figure 11.

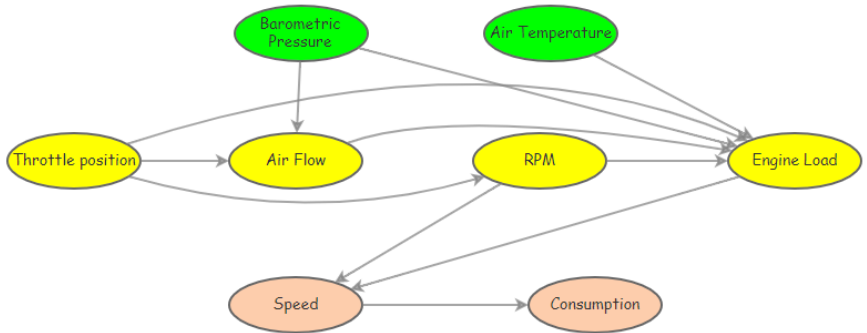


Figure 11: Expert OBD Network

4.1.2.1 Variables

The variables in the OBD data can be organized into three tiers including an environmental, engine, and trip tier. A more in-depth description of how each of the variables was calculated can be seen in Table 2. The type of variable in Table 2 is “Discrete

Tertiles Continuous variables” this phrase means that the variable was continuous, and all of the samples were grouped into Tertiles and given a discrete number based on that grouping.

Table 2: Variables in the OBD Dataset

Variable	Type of Variable	Calculation method (units)	Characteristic of
Barometric Pressure	Discrete Tertiles Continuous Variable	Instantaneous barometric pressure (pascals)	Environment
Air Temperature	Discrete Tertiles Continuous Variable	Instantaneous air temperature (degree C)	Environment
Throttle Position	Discrete Tertiles Continuous Variable	Instantaneous throttle position (% full)	Engine
Air Flow	Discrete Tertiles Continuous Variable	Instantaneous air flow (gram/second)	Engine
RPM	Discrete Tertiles Continuous Variable	Instantaneous RPM (rotations/minute)	Engine
Engine Load	Discrete Tertiles Continuous Variable	Instantaneous Engine load (% maximum)	Engine
Speed	Discrete Tertiles Continuous Variable	Instantaneous speed (km/hour)	Trip
Consumption	Discrete Tertiles Continuous Variable	Instantaneous consumption (L/100km)	Trip

The environmental tier variables describe the environment in which the trip is being made and are represented by green nodes in Figure 12. The variables in this tier are the instantaneous barometric pressure and instantaneous air temperature. The engine tier variables include the average throttle position, airflow, RPM, and engine load at each second during the trip. These engine variables are in yellow in Figure 12. The trip tier variables include the speed of the vehicle and the instantaneous fuel consumption. These trip tier variables are shown in salmon in Figure 12. The ordering of the variables follows left to right top to bottom in Figure 12. The ordering was determined from the same equations used to determine the expert network. These equations can be seen in APPENDIX A. OBD GOVERNING EQUATIONS.

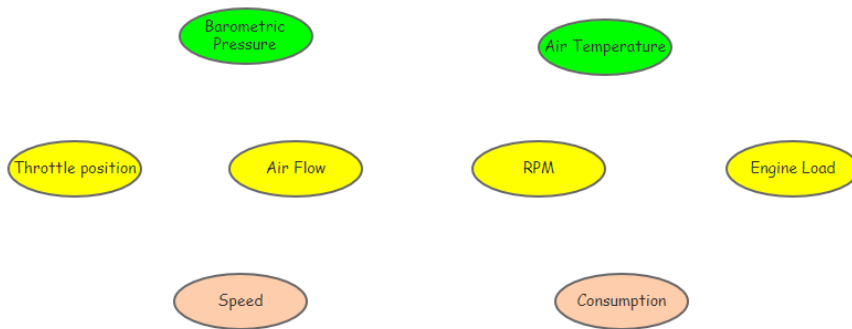


Figure 12: OBD Variables

4.1.3 Divvy Data

The data for the Divvy network has the most uncertainty and comes from two sources that have been combined to make a single dataset. The first source is the data from the Divvy bike share program in Chicago [22]. The evaluated dataset for the Divvy bike share consisted of the subscriber rides for 2014 which included 1,663,394 million individual rides. The second data source is from the National Oceanic and Atmospheric Administration (NOAA) which reports the weather conditions for Chicago using the weather station at Chicago O'Hare international airport [67]. Samples were downloaded from the NOAA and aligned with the Divvy samples dates and times. The combined dataset serves as the primary case study for this research and represents a sociotechnical system with data amalgamated from multiple sources and where many latent variables are expected to exist. The evaluation built around the work of Bowden/Turkington, Lemmer, and Zhang/Spirtes was applied and the system failed to meet the Causal Sufficiency Assumption for even local regions of the graph. That is because there were no instrumental

variables that could not be seen to mask the effect of some latent variable, thus implying a large number of latent variables. Additionally, the framework outlined by Lemmer deemed that the system data did not satisfy the Markov Assumption, because the interaction of variables could not be broken down to the signal level. The Faithfulness Assumption was known to be false considering that the previous two assumptions were invalid.

A researcher working with 2014 Divvy bike share data has been developing a statistical model for predicting ridership at individual stations in Chicago. Figure 13 represents that researcher's believed true network for the Divvy network and arrived at this network via a three step process. First, the researcher read existing research on the variable interactions[68][69][70][71][72][73][74]. Second, the researcher formulated a network based on the reading and intuition gained through first hand observation. Finally, that network was validated by evaluating the predictive ability of the network with real data.

The expert building the comparison network suspected the presence of feedback loops, but was comfortable building a network without any feedback loops; so, there was no need to trim them to arrive at an acyclic network. However, with the inclusion of additional variables such as bike availability there may be feedback loops. Such a loop would act as a negative feedback for the number of riders, since as riders goes up the availability will go down which will prevent more people from riding. The effects of this feedback loop could be captured by a dynamic BN which feeds the availability of bikes variable back into the number of riders variable in the next time step.

Commented [WM69]: added in section as per most recent feedback

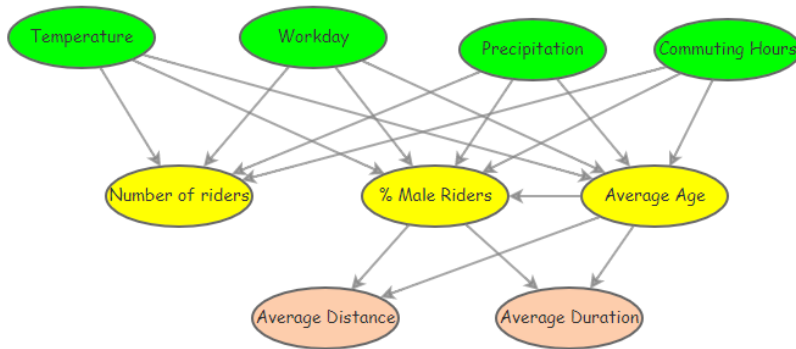


Figure 13: Expert Network for Divvy

4.1.3.1 Variables

From Divvydata.com and NOAA the researchers were able to obtain nine variables that describe the utilization of the bike share in terms of the environment, the demographics of riders, and the rides they take. Samples were collected every time a Divvy subscriber checked out a bike and all variables relate to the instantaneous value of the variables at the moment a bike was checked out (except for precipitation which was done on a daily basis).

Table 3: Variables in the Bike Share Dataset

Variable	Type of Variable	Calculation method (bin edges and units)	Characteristic of
Temperature	Discretized continuous variable in five bins	Interpolated between low at 6 AM and high at 3 PM [-200-32-50-68-86-200] degrees	Environment
Precipitation	Discretized continuous variable in five bins	Precipitation for a given day from NOAA [0- 0.05- 0.5- 1- 1.5- 5] inch	Environment
Workday	Boolean	Mon-Fri and not a public holiday	Environment
Commuting hour	Boolean	Commuting between 6-8 AM and 4-7 PM	Environment
Number of riders	Discrete Quartile Continuous Variable	Number of rides in progress at bike checkout time [0-48-88-152-399] current riders	Demographic
Gender Ratio	Discrete Quartile Continuous Variable	Gender ratio of riders at bike checkout time [0-23-28-32-100] % women	Demographic
Average age	Discrete Quartile Continuous Variable	Average age at bike checkout time [0-33.86-35.66-37.27-91] years	Demographic
Average distance	Discrete Quartile Continuous Variable	Average distance traveled at bike checkout time [0-1.257-1.403-1.518-5] miles	Ride
Average duration	Discrete Quartile Continuous Variable	Average duration of travel at bike checkout time [0-1145-1533-2398-10000] seconds	Ride

The variables in the samples can be grouped into three tiers. Variables in the environmental tier relate to variables that influence people’s choice to ride and how to ride. Variables in the demographics tier seek to describe the population of riders. Variables in the ride tier capture the way in which users are utilizing the system in terms of distance traveled and duration.

The top tier variables are those that relate to the environment and social conditions around a ride. This tier includes temperature outside, as calculated by interpolating between the daily high and low. The temperature is important because it will dictate the comfort of the ride and influence whether people ride or not. Second is the precipitation level, which will also affect the comfort of the ride. Third, is a variable that captures whether it is a workday and should relate to users reason for riding. Finally, the research has included

whether a trip is taken during commuting hours to indicate if the bike is being used by someone to get to work. These variables capture the relative comfort of the ride being taken and theoretically whether the ride is work related or for leisure. These variables will affect the demographics of riders and the characteristics of the rides being taken.

The second tier variables, which are affected by the first tier variables, describe the population using the bike share. There are three variables in this demographics tier. The number of riders currently on bikes when each ride begins captures the utilization of the bike share system at a given moment. This variable is one of the variables that the research is interested in increasing. The value of this variable could be influenced by the four variables in the environmental tier and any variables that were not accounted for. The gender ratio of current riders informs the utilization split between male and female and can highlight when one group is underutilizing. The gender ratio is also of particular interest to this research. The average age of current riders acts in a similar fashion to the gender ratio in that it informs designers what segment of the population is using at a given moment. These variables inform designers as to the number, gender breakdown, and age of the current riders.

The third tier variables relate to the utilization of the bike share by the users identified in the demographics tier under the influence of the variables in the environmental tier. These variables include the average distance and average duration of any ride in progress (once it is completed). The environmental variables are in green, the demographic variables are in yellow, and the ride variables are in salmon in Figure 14. The variables also had to have an ordering assigned over them for use in some of the algorithms. This ordering was determined by the same experts that specified the expert network used for

comparison. The ordering followed that the environmental variables would precede the demographic variables which would precede the ride variables. The variable ordering was determined to be temperature, precipitation, workday, commuting hour, number of current riders, gender ratio of current riders, average age of current riders, average distance of current riders, and average duration of current riders in that order.

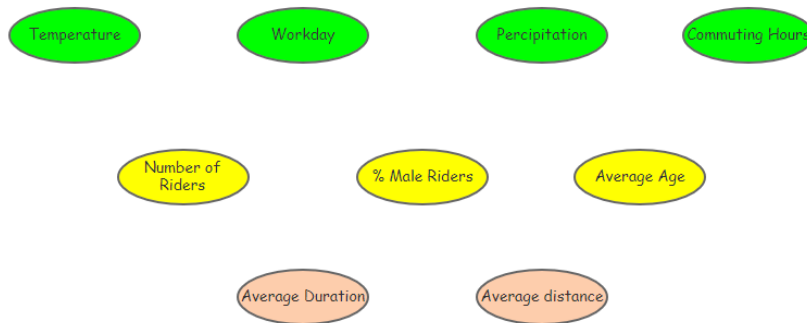


Figure 14: Three tiers of Variables

4.1.4 Datasets and assumptions

The three datasets presented all vary in the number of assumptions they satisfy. The idea is to go from a simulated dataset (represented by ALARM) which satisfies all the assumptions through an intermediary (represented by the OBD) to a dataset that satisfies none of the assumptions underlying structure learning (represented by Divvy). This progression allows for an examination of the effects of each of the assumptions and how they will affect the algorithm performance. The assumption each dataset satisfies or partially satisfies is outlined in Table 4.

Table 4: Assumptions and datasets

	ALARM	OBD	Divvy
Causal Sufficiency	√	Partially √	X
Markov Assumption	√	X	X
Faithfulness Assumption	√	X	X

Commented [TC70]: before submitting, make sure tables don't spill over pages.

4.2 Methodology

The methodology used to generate and evaluate the validity of BNs in this research has five distinct steps. These steps, in order, are pre-processing/characterizing data \mathcal{D} , using that data \mathcal{D} to learn a BN structure \mathcal{G} , using the data \mathcal{D} and structure \mathcal{G} to learn the BN's parameters Θ , evaluating the ability of the BN to perform inference on a set of holdout data, and comparing the BN structure to an expert's intuition. The steps can be seen in Figure 9.

4.2.1 Pre-processing/Characterizing data

Pre-processing the dataset \mathcal{D} for use in algorithms involved aligning data from different datasets such that they coincided temporally and spatially and discretizing the contents of the datasets. This step also involves identifying what assumptions are valid in the datasets.

4.2.1.1 Characterizing Datasets

In order to characterize a data set the validity of the three assumptions discussed by Margaritis need to be verified [42]. The first assumption to be investigated is the Causal

Sufficiency Assumption, work has been done in trying to verify this assumption's validity using instrumental variables [42][1][46]. The approach taken in this work is a qualitative method that can be augmented with a quantitative analysis assuming that the correct instrumental variables are found and data is present. The process involves determining if a potential causal relationship $A \rightarrow B$ is valid or is the result of a latent variable in the system. The basic graphical structure explaining how instrumental variables operate can be seen in Figure 15.

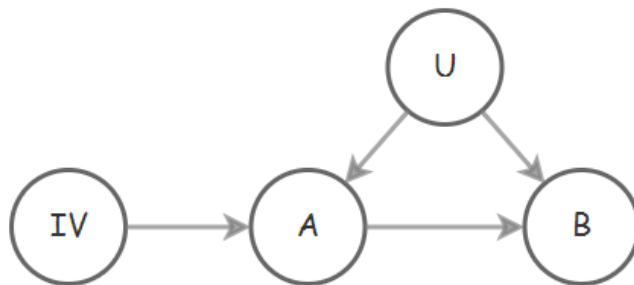


Figure 15: Instrumental Variables

In this system there is a potential causal relationship between A and B, however, that relationship may be obscured or be a result of an unknown variable U. To determine if A is in fact effecting B an instrumental variable can be introduced into the system. To serve as an instrumental variable IV must be correlated with the variable A and must be uncorrelated with the latent variable U as represented in Figure 15 [1]. By observing how the instrumental variable IV relates to B one can exclude the effects of variable U on the relation between A and B [46]. By changing the instrumental variable and observing the changing relations one can test for potential latent variables U in particular domains. This approach also allows for researchers to hone in on the latent variable by using a list of

instrumental variables that do and do not share relationships with the potential latent variable's domain. This can be done quantitatively if data is available or qualitatively as a systematic thought process.

The Markov Assumption is tested by using the qualitative test described by Lemmer. In this work the system is broken down to evaluate if the interaction of the different components can be described in terms of identifiable signals; if that is not the case then the Markov Assumption is likely not appropriate [2].

In order to test the Faithfulness Assumption Zhang/Spirtes have come up with a process by which two algorithms are run and if they produce the same result the dataset must be Faithful to some graph [3]. The algorithms that are run are the PC algorithm and the conservative PC algorithm which differ in their assumptions when evaluating shielded and unshielded triples [57][3]. When the output of these two algorithms are equivalent it means that the dataset is faithful to some casual graph.

4.2.1.2 Data Misalignment in the Divvy dataset

The separate datasets were aligned temporally as possible to reduce the effects of combining different datasets. The NOAA data contained the high and low temperature for each day. To account for variability during the day, temperature values were interpolated between the daily high temperature at 3:00 PM and the daily low temperature at 6:45 AM. Spatial misalignments are still present, however, as the weather data are from the airport and the bicycle data vary in location.

4.2.1.3 Discretization

Choosing how to discretize the variables that will go into the BN structure learning algorithms involves trading off higher granularity for computational feasibility based on the amount of training data available. Higher granularity provides more information on behaviors and causes, but also requires larger sample sizes to represent the full set of potential states.

The discretization of the continuous variables in the two observational datasets, OBD and Divvy, were defined so that the maximum training dataset available would be more than thirty times the size of the variables state space. The simulated ALARM values were already discretized. Thirty was assumed to be enough samples to inform the parameters of a conditional probability table the size of the state space. This approach was used to ensure that there were enough data points to fill out the conditional probability tables.

When variables were purely continuous the discretization defined an equal number of samples for each of the discrete states of the variable. This ensured that there would be instances present to inform learning of the entire state space. When variables had categorical analogs, such as temperature which can be considered sub-freezing, cold, comfortable, and hot, the discretization corresponded with the analogous categories. For example, in temperature freezing was taken as the top of the lowest bin because it represents a categorical change and the bin sizes were made to have a range of 18 degrees in order to have an appropriate number of discrete states. An appropriate number of discreet states is defined in the paragraph prior to this one. These two discretization methods were

used for the observational datasets to ensure there was enough data to learn the parameters in the conditional probability tables and that variables that were categorical in nature were represented in a manner that was informative.

4.2.1.4 Sample Sizes

For the ALARM network ten data sets were generated by sampling the true BN, each of the ten sampled datasets has 200,000 instances. Each of these ten 200,000 instance datasets was given to the four algorithms. The datasets were then reduced by a factor of ten to get ten 20,000 instance datasets. They were reduced to 10,000; 5,000; 2,500; 1,250; 625; 312; and 156 size instance datasets. There were ten distinct datasets each of which was broken down into 9 different size subsamples as shown in Table 5. This gave a total of 90 samples for each of the four algorithms to run with. A holdout sample of 200 instances was also generated for use as the test set.

Table 5: Sample size Breakdown ALARM

Number of instances in sample	Number of samples
200,000	10
20,000	10
10,000	10
5,000	10
2,500	10
1,250	10
625	10
312	10
156	10

The OBD dataset consisted of 242,951 of which 240,521 instances were used as training data and 2,430 instances were removed for use as the testing set. The data was compiled from one vehicle driving around the Singapore metropolitan area. The holdout dataset was generated by pulling every 2000th instance from the initial 242,951 instance dataset in order to get a representative sample. The manner in which the training data was split up into sub samples is complicated and best explained with Table 6, but for each splitting of the dataset the number of datasets doubled. The number of data samples was doubled every time the size of the data samples was halved to get as many sub samples as possible for the learning algorithms. The total number of unique datasets given to the four algorithms was 255 for OBD. In the weighting in the final Average BN all tiers were equally weighted meaning the 128 samples at the smallest size had the same influence on the graph as the 1 network learned with the full dataset. The reason this data approach was not done for the ALARM dataset was that it would not change the outcome since the ALARM results are simulated. This was not done for the Divvy dataset because it was larger and the same process would have given 2047 unique datasets which the researchers calculate would have taken a week of runtime to evaluate. The resulting standard deviations shown in Table 9 in the results section indicate that the effects of the number of samples on the results for each tier are small.

Commented [WM71]: This section was added as per the last feedback from you.

Table 6: Sample size Breakdown OBD

Number of instances in sample	Number of samples
240521	1
120260	2
60130	4
30065	8
15032	16
7516	32
3758	64
1879	128

The initial Divvy dataset contained 1,663,394 instances and every 2000th instance was removed to generate an 831 instance sample that would serve as the test set. The remaining 1,662,562 instance sample was then broken down into 10ths, 100ths, 200th dataset, then 400th, then 800th, then 1600th, then 3200th, then 6400th, and finally 12800th datasets. The instances were broken up by taking every 10th, 100th, 200th, etc. in sequence so that the data would be representative of the entire year. This sampling method was also done with the training data. The breakdown is best illustrated in Table 7. The Divvy dataset does not satisfy any of the three assumptions that underlie structure learning according to the evaluation used in this work.

Table 7: Sample size Breakdown Divvy

Number of instances in sample	Number of samples
1,662,562	1
166,256	10
16,625	10
8,312	10
4,156	10
2,078	10
1,039	10
519	10
259	10

4.2.2 Learning Bayesian Network Structure

Four different BN structure learning algorithms, the K2, PC, MMHC, and SCA algorithms were tested using the Causal Explorer toolkit [75] and the Bayes Net Toolbox [64] provided by Vanderbilt University and the University of Utah respectively. The algorithms were given the three datasets to learn the networks. The default parameters for the four algorithms in Causal Explorer and Bayes Net Toolbox were utilized for this research experiment and can be seen in Table 8. For each datasets the algorithms were run multiple times for each sample size, as described in data pre-processing section and Table 5, Table 6, and Table 7. Different fractions of the original dataset were used to see how well the algorithms perform with the type of small datasets typical in sociotechnical systems.

Table 8: Algorithm Default Parameters

	K2	PC	SCA	MMHC
Scoring Function	K2	N/A	BDeu	BDeu
BDeu weight	N/A	N/A	10	10
Conditional independence test	N/A	G ²	Bayesian	G ²
Independence threshold	N/A	.05	.05	.05

4.2.3 Learning BN parameters

To populate the BN structures with parameters Θ the MLE function in the Bayes Net Toolbox was utilized with the learned BN structures. The MLE function was given the matrix representation of the learned BN structure $\mathcal{G}_{\text{learned}}$ and the original dataset \mathcal{D} used to learn that structure $\mathcal{G}_{\text{learned}}$ as inputs. The MLE then learned the appropriate parameters Θ_i for each variable node i to n . The MLE program requires that any graph \mathcal{G} used to learn parameters be acyclic. This requirement posed a problem for the PC algorithm which returns networks with bidirectional nodes. To learn the parameters for the PC network the edges of the PC algorithm's learned structure were oriented according to the same ordering given to the K2 algorithm. The parameters Θ learned are combined with the structure $\mathcal{G}_{\text{learned}}$ and resulted in a complete BN capable of predictive inference.

4.2.4 Evaluating Predictive Inference Accuracy

To evaluate the quality of the BN learned, this research tested the learned BN's ability to perform predictive inference. The ability to perform predictive inference is measured by the BN's ability to accurately predict the value of each variable in the BN given all of the other variables in the BN. This test was achieved by utilizing a holdout set

$\mathcal{D}_{\text{holdout}}$ taken out of the original datasets \mathcal{D} used for training. For example, if each sample in a dataset has ten variables the first variable would be removed and the other nine would be given to the inference program along with the learned BN. The inference program (junction tree algorithm) then uses the nine variables it has and the BN learned to guess the missing variables value. This process was then repeated by excluding the second variable and giving the program the first and third through tenth variables. This process was performed for each of the variables in the sample and was performed over a large number of samples that were representative of the original dataset \mathcal{D} . The result was a percent accuracy for the prediction of each of the variables in the BN given all of the other variables in the BN. The prediction accuracy for each of the variables was averaged together to give a single accuracy for the learned BN.

4.2.5 Evaluating structure against expert model

The predictive accuracy of the learned network $\mathcal{G}_{\text{learned}}$ is a necessary but insufficient evaluation of the quality of the BN structures. A naïve BN can perform inference effectively, but does not necessarily provide a designer with system insight. As discovered in a study by Yu et al, non-experts or engineers with incomplete domain knowledge have difficulty identifying coupled variables from experimenting with predictive models [76]. The learn BN structures, which visually present variable couplings, were thus compared against the true or the most likely BN structure as determined by equations and subject matter experts. This comparison of structures was achieved by looking at the Structural Hamming Distance (SHD) and a metric built off of the SHD called the Modified Structural Hamming Distance (MSHD) [77].

4.2.5.1 SHD and MSHD

The SHD is the number of changes needed to get from one network to another, the smaller the SHD the closer the two networks are to one another. The SHD can be defined mathematically and is done so in the work of Peters and Buhlmann [78]. Essentially, the SHD is a count of the number of edge additions, edge deletions, and edge reversals that are needed to go from one network structure G_1 to a comparison network structure $G_{Compare}$. The SHD was developed by Tsamardinos and is used to evaluate the structural differences between different causal networks to have a metric fully oriented towards structural discovery, rather than inference [50].

This research introduces a second metric called the Modified Structural Hamming Distance (MSHD) which is equivalent to the SHD across an undirected DAG [50]. This definition means that the MSHD is the SHD minus the penalty for incorrectly oriented edges. This metric was utilized because some of the algorithms used in this research are given the ordering over variables as an input or post processing step, while others are not given the ordering. That selective omission of information means the algorithms that have the ordering will never make an error in the direction of an edge and will never suffer a penalty for incorrect edge orientation. For that reason, the SHD is not ideal for comparing algorithms that have the node ordering and algorithms that do not have the node ordering. The K2 algorithm is given the variable ordering as an input and the PC algorithms results are post processed such that the final output is acyclic and follows the ordering across the variables. Conversely, the SCA algorithm and the MMHC algorithm are not provided with the ordering and make errors in edge direction that will not be present in the K2 and post processed PC network structures.

4.3 Chapter Summary

This chapter introduces the ALARM, OBD, and Divvy datasets, the variables in each of those three networks, and how those datasets were broken up for the research. The true network for the ALARM dataset was presented and the methodology to determine an expert network for the OBD and Divvy systems was presented. The chapter also introduced a five step general methodology for learning and evaluating a BN learned from data. These steps are pre-processing/characterizing data \mathcal{D} , using that data \mathcal{D} to learn a BN structure \mathcal{G} , using the data \mathcal{D} and structure \mathcal{G} to learn the BN's parameters Θ , evaluating the ability of the BN to perform inference on a set of holdout data, and comparing the BN structure to the true or expert network for the system.

CHAPTER 5. RESULTS AND ANALYSIS

5.1 Results

The results for each of the three datasets are graphs consist of the runtime, density of learned networks, SHD and MSHD from the expert/true network, and predictive inference accuracy for the different size training datasets. The runtimes and densities of the algorithms with each of the three datasets are compared to show how different dataset qualities change the behavior of the algorithms. Then alternative networks describing each dataset are compared to the BN and an average of all of the BNs learned known as an Average BN. These comparisons show how the BNs learned from data stack up against an unconnected, fully connected, correlation based, and expert built networks. All the information presented gives more insight into how the different assumptions about the datasets influence the performance of algorithms as measured by predictive inference accuracy and structural distance from an expert or true network structure.

5.1.1 ALARM Dataset

The ALARM dataset has a known true structure and the training samples used for learning and inference were generated by randomly sampling the true network. The network consists of 37 nodes connected with 42 edges [51]. This network serves as the control case because the true network structure is known and the data generated is representative of that true network structure. Since it is simulated data it will meet all three assumptions underlying causal BN structure learning and the methodology for determining valid assumptions did not need to be applied.

Commented [MWD72]: added

5.1.1.1 Runtime for Algorithms

The runtime of the four different algorithms with the different sized sample sets can be seen in Figure 16. The computations were performed on an Intel Core i7-4790 CPU at 3.60 GHz and with 16 Gb of RAM. From Figure 16 it can be seen that the slowest algorithms was the SCA and the fastest was the MMHC for most of the sample sizes, except for the largest sample size in which the K2 algorithm was the fastest.

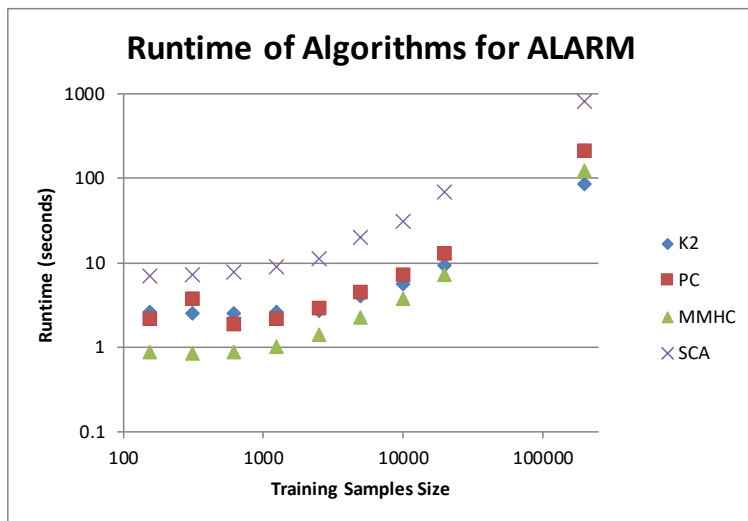


Figure 16: Runtime of Algorithm for ALARM

5.1.1.2 Density of structures

The maximum number of edges possible for a network of 37 variables is 666 edges and the true network has 42. The number of edges in the PC and K2 algorithms decreases as more samples are provided, as seen in Figure 17. Above 1,000 samples the K2, MMHC,

and PC algorithms all learn between 40 and 50 edges. Meanwhile, the SCA algorithm returns a denser network of around 60 edges even with the largest training data set.

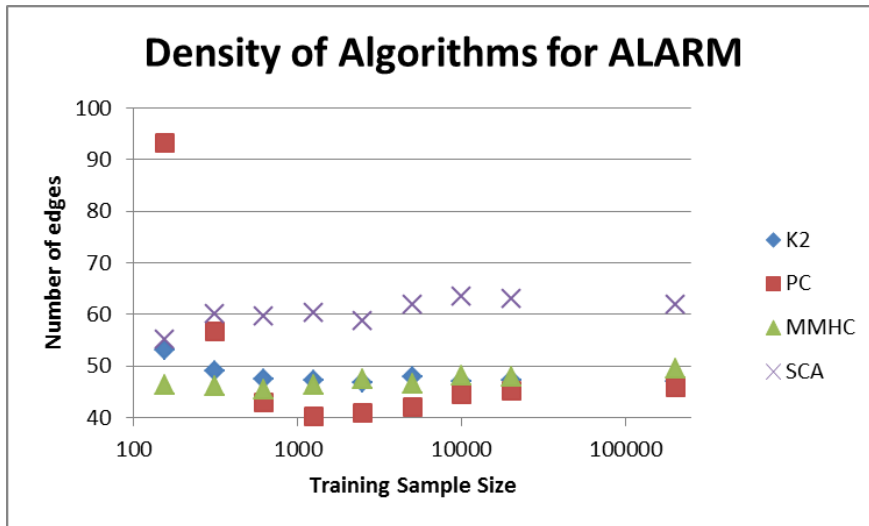


Figure 17: Density of Algorithms for ALARM

5.1.1.3 SHD and MSHD of structures

The MSHD and the SHD measure the number of edge additions, deletions, and reversals needed to go from the structure learned by the different algorithms $\mathcal{G}_{\text{learned}}$ to the true structure of the ALARM network $\mathcal{G}_{\text{true}}$. The MSHD and the SHD will always be the same for the PC and K2 algorithms because the K2 algorithm was given the ordering over the variables as an input and the PC algorithm’s results were post-processed to orient the edges such that the resulting graph was acyclic as discussed in section 4.2.3.

The MSHD and the SHD for most of the algorithms decrease as the size of the training sample increases as seen in Figure 18 and Figure 19. The exception to this trend is

the SCA algorithm which learns a structure with 27 incorrect edges and 13 incorrect orientations even with the full ALARM dataset. The PC and K2 algorithms MSHD are 0 and 1 respectively with the full training dataset.

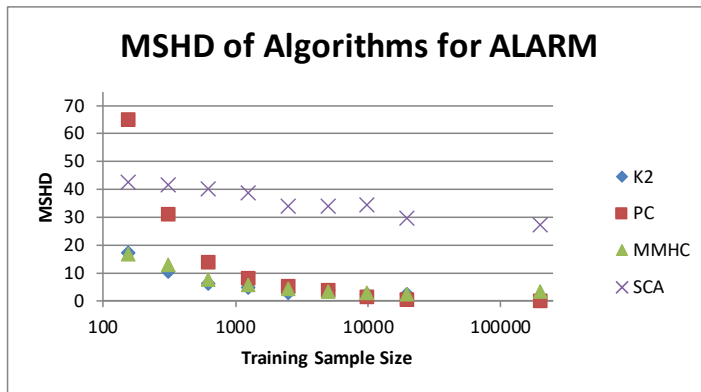


Figure 18: MSHD of Algorithms for ALARM

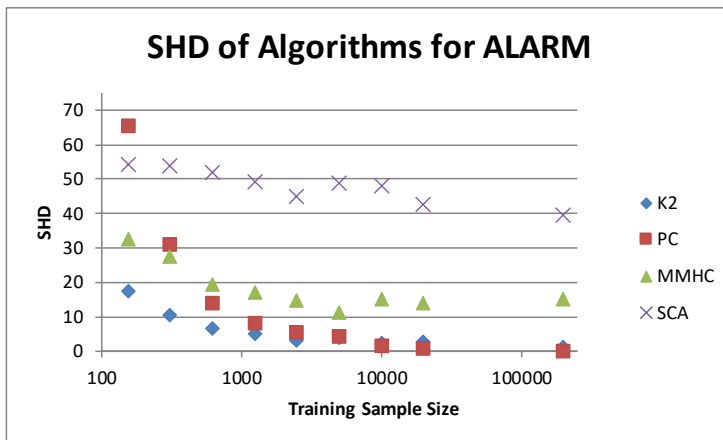


Figure 19: SHD of Algorithms for ALARM

5.1.1.4 Predictive Inference Accuracy of Learned Bayesian Networks

The predictive inference accuracy of the BNs learned given increasing sample sizes is presented in Figure 20. As all of the algorithms are given more samples their predictive inference accuracies increase. With 156 training samples the PC algorithm performs the worst followed by K2, SCA, and MMHC. Above 156 training samples the MMHC, K2, and PC algorithms all perform inference with accuracies within a 2.5% range. The SCA algorithm is consistently less accurate in predictive inference, but still performs within a 5% range of the other algorithms for predictive inference accuracy. An unconnected network was capable of inferring the correct missing variable 77.7% of the time with all of the sample sizes. A naïve BN would have been desirable as a secondary benchmark, however calculating the conditional probability table for such a network was beyond the computational capability of this research. The reason this computation is not possible is because it would require a 36 dimensional table for each of the 37 variables and each table would have roughly 8,660,000,000,000,000 unique entries.

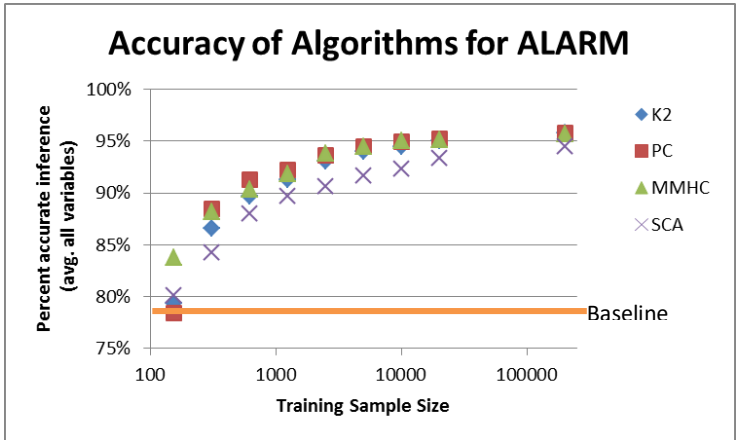


Figure 20: Accuracy of Algorithms for ALARM

5.1.2 OBD Dataset

The OBD dataset was pre-processed using the Pandas package in python and was imported to Matlab where the algorithms and the evaluation of the resulting structures was performed. This dataset serves as an example dataset for which known equations explain the interaction of the variables. This dataset also comes from a single source and can reasonably be assumed to partially satisfy the Causal Sufficiency Assumption based on the methodology built around instrumental variables [1]. The Markov Assumption and Faithfulness Assumptions are not satisfied. Since all of the assumptions are not satisfied the learned networks cannot be considered causal, but should inform as to the location and nature of causal connections [7].

Commented [MWD73]: added

5.1.2.1 Runtime for Algorithms

The runtimes for the OBD dataset given different training sample sizes can be seen in Figure 21. The algorithms with a conditional independence component (PC, MMHC, and SCA) are on average ten times slower than the K2 algorithm. The PC and MMHC algorithms always run within 5% of each other in terms of execution time.

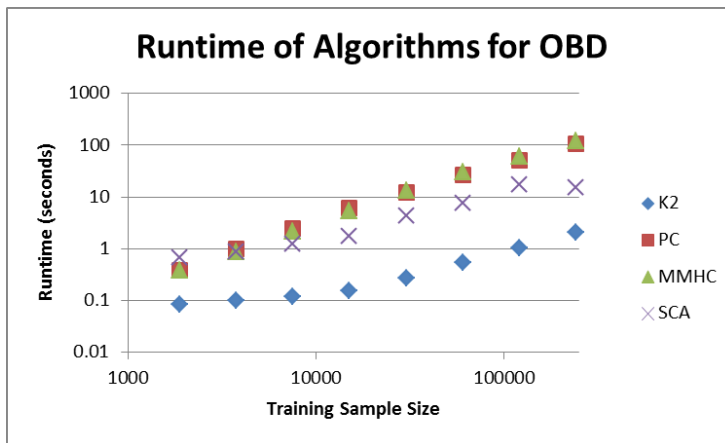


Figure 21: Runtime of Algorithms for OBD

5.1.2.2 Density of structures

The maximum number of edges for a network of eight variables is 28. As the number of training samples increases the density of the networks continues to increase. The PC algorithm tends towards being denser than the other three algorithms and arrives at a fully connected network with a 60,130 instance training dataset and any training set larger than that. The density of the other three algorithms increases steadily as training dataset size is increased, but they do not arrive at a fully connected network unlike the PC algorithm as seen in Figure 22.

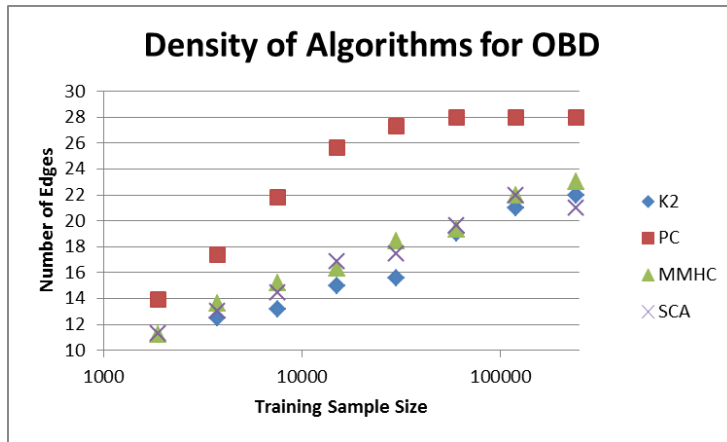


Figure 22: Density of Algorithms for OBD

5.1.2.3 SHD and MSHD of structure

The MSHD and the SHD were calculated with respect to the expert built network for the OBD data introduced in section 4.1.2. The expert network structure $\mathcal{G}_{\text{expert}}$ was based off of equations relating the parameters recorded in the OBD dataset. The MSHD and SHD results are plotted in Figure 23 and Figure 24. The SHD and MSHD both increase as the algorithms are given more data, indicating a trend away from the expert network and towards a fully connected network.

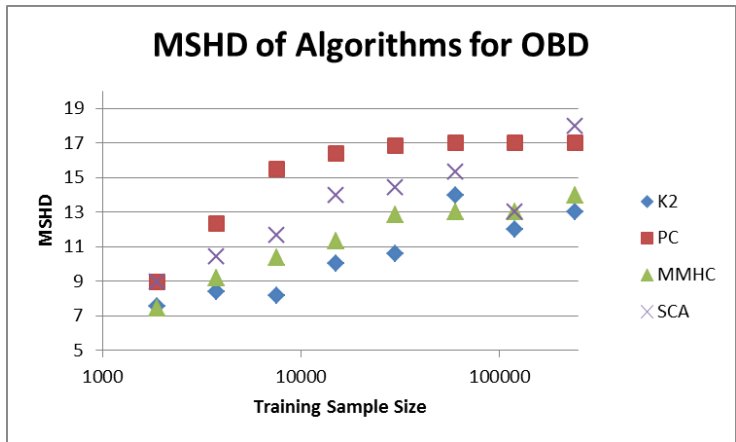


Figure 23: MSHD of Algorithms for OBD

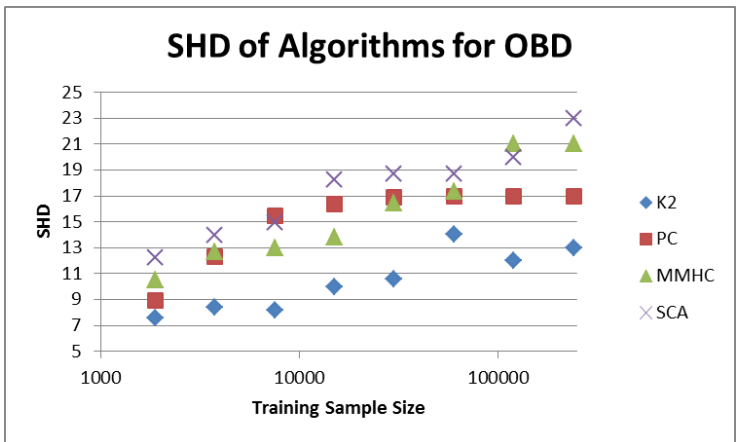


Figure 24: SHD of Algorithms for OBD

5.1.2.4 Predictive Inference Accuracy of Structures

The predictive inference accuracy of the structures and parameters learned from the OBD data ranged between 31% and 37%. The predictive inference accuracy of an unconnected BN with parameters learned using MLE was 31.7%. This unconnected BN is

analogous to finding the marginal probability of each variable in the network. The accuracy of the unconnected network is represented in Figure 25 by the orange line labelled “Base” and serves as a point of comparison. The accuracy of a fully connected BN, otherwise known as a naïve BN, is also represented in Figure 25 as an orange line labelled “Naïve”. Instances where the structure learned by the algorithms is fully connected naturally coincide with the performance of the naïve network. The hybrid (SCA and MMHC) and conditional independence (PC) algorithms tend to perform predictive inference with an accuracy somewhere between that of the unconnected and naïve BNs. In a few cases the algorithms learned network exceeds the performance of the naïve network. The score based K2 algorithm was consistently below the unconnected network.

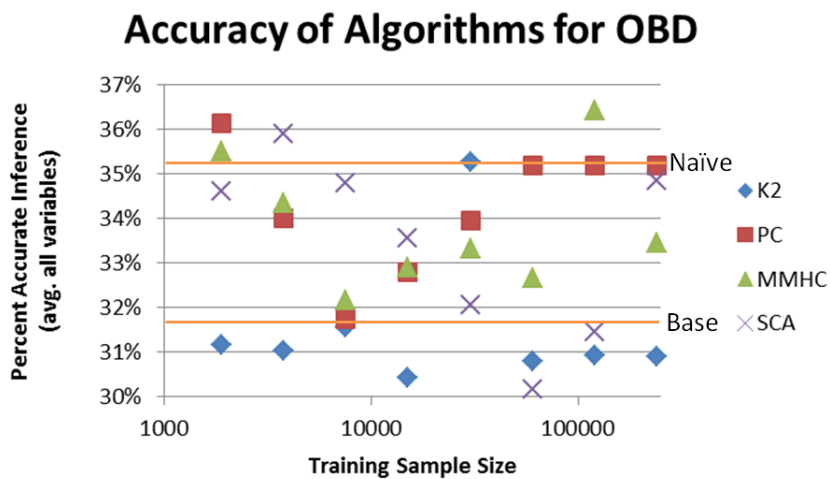


Figure 25: Accuracy of Algorithms for OBD

5.1.2.5 Standard Deviation of Results

The standard deviation of the results for of the ODB data can be seen in Table 9. These results show that the effect of the number of samples on the results for each tier are small and validates that the methodology used to break up the dataset did not have a detrimental effect on the results. Note that the standard deviation is calculated across all the samples for all of the four algorithms, which is why there is a standard deviation when the number of samples is one. That standard deviation is across the four algorithms' results.

Table 9: Standard Deviation for Results of OBD

Density STD	% Accuracy STD	MSHD STD	Number of Samples	Size of Sample	Avg Network Density
3	1.95%	2	1	240521	24
3	2.67%	2	2	120260	23
4	2.33%	2	4	60130	22
5	3.12%	3	8	30065	20
4	3.07%	3	16	15032	18
3	3.26%	3	32	7516	16
2	3.63%	2	64	3758	14
2	3.57%	1	128	1879	11

Commented [C74]: How do you have a STD for 1 sample? Did you run it ten times with each sample? Was the next run 20 times? 10 times each sample?

WM - The STD is across all four algorithms results so if you run it once with four different algorithms you still have a set of four. The STD is

5.1.3 *Divvy Dataset*

The Divvy dataset serves as the sociotechnical case study and does not satisfy the Causal Sufficiency Assumption, Markov Assumption, or Faithfulness Assumption. The Causa Sufficiency Assumption was deemed invalid by the methodology built around instrumental variables. The Markov Assumption was also considered invalid as per Lemmer's qualitative method and as a result the Faithfulness Assumption must also be

invalid, so the process developed by Zhang/Spirtes was not implemented. The Divvy dataset contained 1,662,562 instances in the largest training dataset.

5.1.3.1 Runtime for Algorithms

The runtimes of the four algorithms are shown in Figure 26 for the Divvy dataset. For the smallest datasets, the SCA is the slowest running in 0.8 seconds whereas the MMHC and the PC algorithm run in 0.2 seconds. The K2 algorithm is the fastest running in under 0.1 seconds. As the sample sizes increase all the runtimes increase and the PC and MMHC algorithms becoming the slowest as the training datasets increase to 4,154 instances and above.

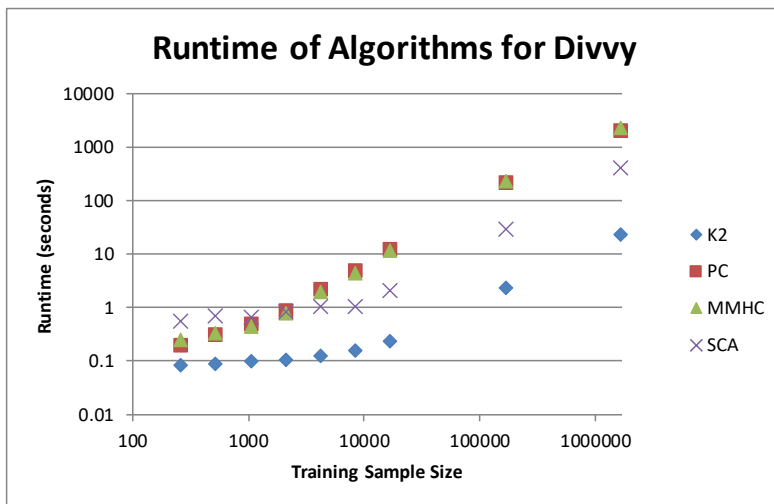


Figure 26: Runtime of Algorithms for Divvy

5.1.3.2 Density of structures

The maximum number of edges for a network of nine variables is 36. The density of all of the networks increases as the sample size is increased and MMHC, K2, and PC all obtain fully connected networks with the full dataset as seen in Figure 27. The PC algorithm also consistently has between 12 and 6 more edges than the other three algorithms.

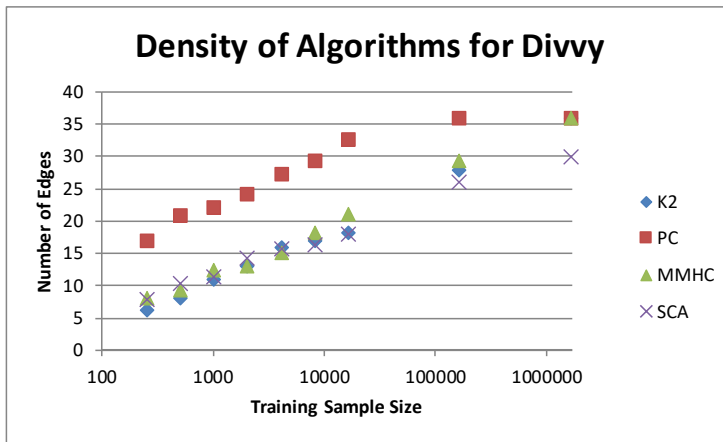


Figure 27: Density of Algorithms for Divvy

5.1.3.3 SHD and MSHD of structures

The MSHD and the SHD were calculated with respect to the expert built network for the Divvy data. This network was specified by a researcher working on the Divvy bike share data and was introduced in section 4.1.3. The resulting MSHDs and SHDs between the learned networks G_{learned} and expert network G_{expert} can be seen in Figure 28 and Figure 29 respectively. As with the OBD data, SHD and MSHD increase as the algorithms are given larger training datasets.

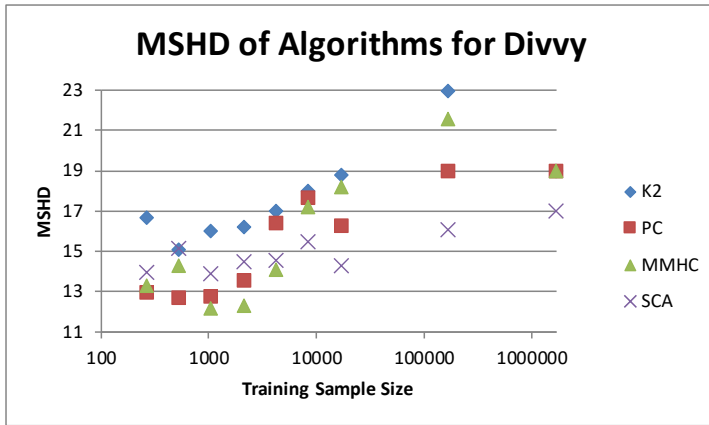


Figure 28: MSHD for Divvy

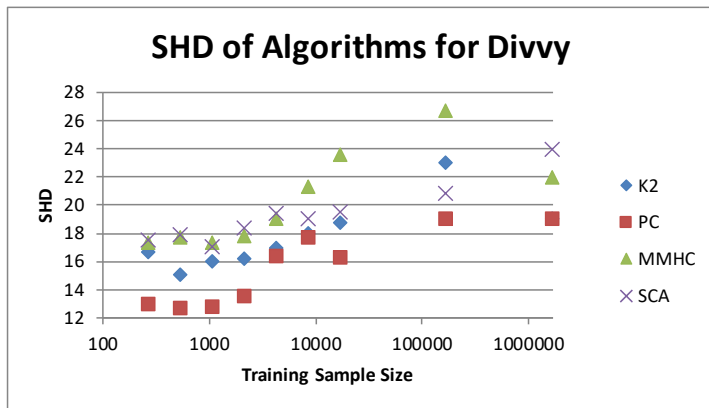


Figure 29: SHD for Divvy

5.1.3.4 Accuracy of Structures

The completely unconnected network with parameters learned using MLE could accurately predict the missing variable in the testing set 34.44% of the time and is represented by the orange line labelled “Base” in Figure 30. A completely connected or naïve BN with parameters learned using MLE could accurately predict the missing variable

34.39% of the time and is represented by the orange line labelled “Naïve” in Figure 30. The networks and parameters learned by the four algorithms have inference accuracies for all sample sizes between 34.52% and 34.1% in Figure 30.

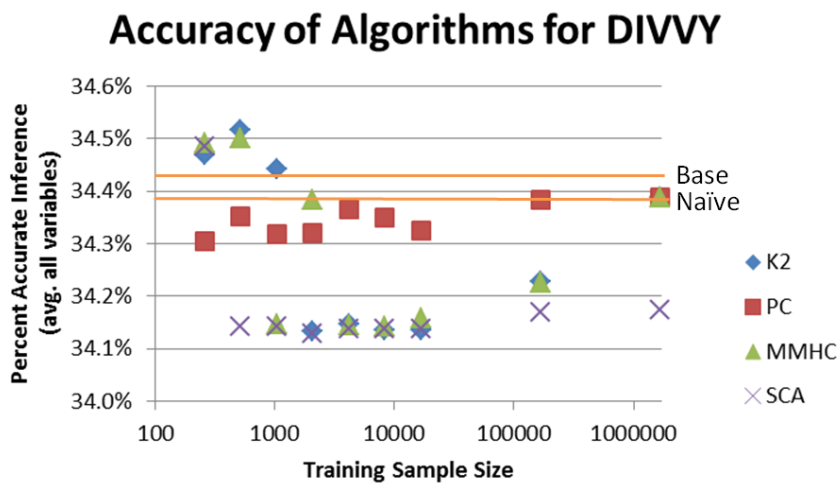


Figure 30: Accuracy of Algorithms for Divvy

5.1.3.5 Standard Deviation of Results

For the Divvy dataset the resulting standard deviations can be seen in Table 10. These results show that the effect of the number of samples on the results for each tier are small and validates that the methodology used to break up the dataset did not have a detrimental effect on the results.

Table 10: Standard Deviation of Results for Divvy

Density STD	% Accuracy STD	MSHD STD	Number of Samples	Size of Sample	Avg Network Density
5	0.09%	2	1	1662562	35
4	0.08%	2	10	166256	30
6	0.08%	1	10	16625	23
5	0.10%	1	10	8312	20
5	0.10%	1	10	4156	19
5	0.14%	1	10	2078	16
5	0.13%	2	10	1039	14
5	0.16%	2	10	519	12
4	0.09%	1	10	259	10

Commented [TC77R76]: Thank you

5.2 Alternative Network Comparisons for Observational Datasets

Comparison of the algorithms' results across the datasets indicates that the observational data present a number of difficulties for the BN learning algorithms. These difficulties can be seen in the execution time of the algorithms and the relative density of the learned structures. These difficulties with observational data are not necessarily unique to the BN learning algorithms, and the performance of the learning algorithms is compared to alternative models in this section. The alternatives investigated are an average of all the BN results, the expert built network, unconnected network, naive network, and a network built using correlations.

5.2.1 Time execution of Algorithms

The most striking difference in runtimes is that the MMHC and PC algorithms run much slower when the data is not simulated (OBD and Divvy datasets). In the simulated ALARM datasets the MMHC and PC algorithms followed different trajectories and were similar in duration to the K2 algorithm. However, in the two observational datasets the

MMHC and the PC algorithm performed within 5% of one another and are an order of magnitude slower than the SCA and K2 algorithms. The increased run-times indicate that the Divvy and OBD data did not have clear independencies in the data like the simulated ALARM data. Without clear independencies the PC algorithm and the MMPC portion of the MMHC algorithm took a long time to screen the parent variables. The MMPC and PC (which operate on a very similar basis) look for independence by testing connections conditioned upon a set of other variables. When there is no independence, the MMPC and the PC algorithms have to iterate through all the variables to arrive at the conclusion there are not any independencies in the network. For the observational datasets the K2 algorithm was the fastest. The K2's speed is due to the fact that the K2 algorithm is given the ordering across the variables as an input, drastically decreasing the number of potential structures in the search space.

There is an unexpected bump in the runtime at the second smallest sample size for the PC algorithm with the ALARM dataset; this bump is most likely due to an update that windows automatically initiated during the execution of the computations. This update pulled resources away from the algorithm that slowed the process down.

5.2.2 Density of Learned Structures

The density of the learned structures varies between the different datasets and gives insight into the nature of the datasets and the different algorithms. In the ALARM network the PC algorithm converges towards the true network structure and maintains a consistent number of edges after an initial decrease in density. This decrease in density occurs as the number of samples increases, because at small sample sizes it is difficult to satisfy the

statistical criteria for independence [18]. The MMHC and K2 maintain a consistent density for the ALARM dataset, this consistency is because the ALARM network datasets satisfy the three assumptions underlying BN structure learning. The SCA algorithm maintains a relatively constant density of between 55 and 65 edges for all of the different ALARM training sample sizes. The SCA algorithm returns a network denser than the true network because it allows for the addition of edges that meets its Bayesian information based threshold up to the specified maximum number of parents while other algorithms more aggressively exclude variables based on conditional independence [8]. While the structure of the SCA is denser than the actual ALARM network it is still capable of accurate predictive inference as seen in Figure 20. The reason for this is that while the algorithm is not necessarily learning the true parents it is learning variables which have a high degree of mutual information, this mutual information indicates that the variables are in or at least near the Markov Blanket of the node under investigation. Since the variables that are identified as parents in the SCA do have this high mutual information the network is still capable of performing accurate predictive inference based off the variables selected.

With the two observational datasets, the algorithms tend towards fully connected acyclic networks as the training dataset sizes increase, this trend is due to the fact that a small effect can become statistically significant with a large sample size in many statistical tests and scoring metrics [79] [54]. The concept of small effect and its influence in large data sets is explained in the paper “It's the Effect Size, Stupid” by Robert Coe [79]. If the three assumptions underlying structure learning were met in the two observational datasets then larger training datasets would just reinforce the independencies in the true network, as was the case for the ALARM network. The trend towards a fully connected graph is a

strong indication that one or more of the assumptions underlying BN structure learning is not valid, which corroborates with the findings of the characterization methodology. The increasing density in the network also explains the drift away from the expert specified structure as measured by SHD/MSHD.

5.2.3 OBD Data

The OBD dataset's variable interactions are expressed by equations governing the mechanical system in APPENDIX A. OBD GOVERNING EQUATIONS. The research methodologies borrowed for characterizing datasets determined that the OBD dataset partially satisfies the Causal Sufficiency Assumption and does not satisfy the Markov Assumption and the Faithfulness Assumption. The metrics being used to evaluate the validity of the learned BN are the SHD/MSHD against an expert generated network and the predictive inference accuracy of the BN.

An unconnected BN as seen in Figure 31 establishes a baseline predicative inference accuracy of 31.7% and has SHD/MSHD of 11 when measured against the expert built network.

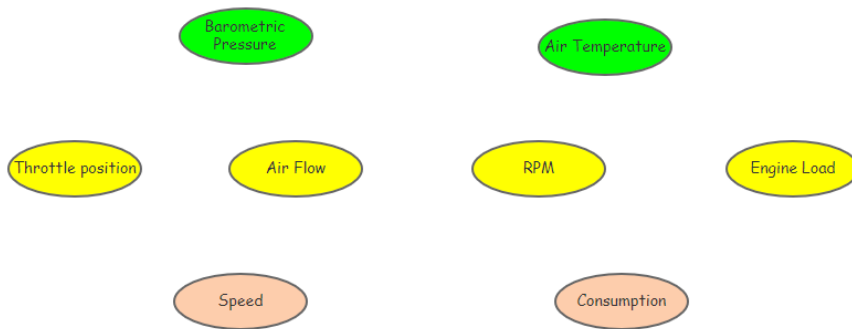


Figure 31: OBD Unconnected Network

In order of increasing sophistication, after the unconnected network is a Naïve BN, where all the nodes are connected to those after it in the variable ordering so as to maintain an acyclic graph structure. This Naïve Network performs predictive inference with an accuracy of 35.18% and has a MSHD/SHD of 17 and is not pictured.

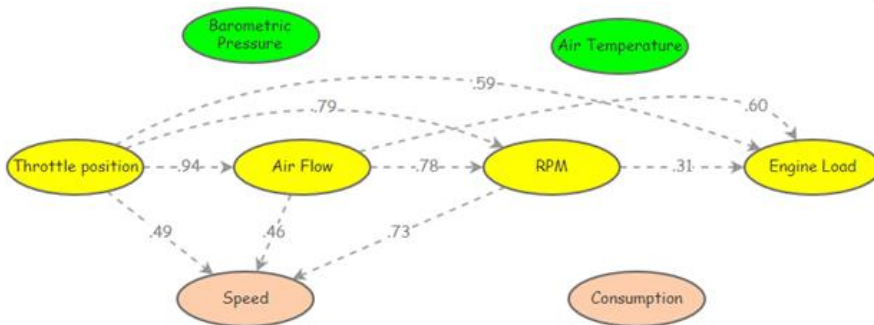


Figure 32: OBD Correlation Network with threshold 0.3 (arcs represent +/- correlation prior to discretization)

Increasing in refinement from the Naïve BN is a correlation network seen in Figure 32. This network was built by finding the correlation between the different variables and

adding an edge between any variables that were correlated above 0.3 as per statistical guidance for significance [80]. The edges are directed according to the determined variable ordering for the OBD data. This correlation network has a predictive inference accuracy of 36.45% and has a MSHD/SHD of 8.

The benchmark that has been used for the MSHD/SHD is the expert network introduced in section 4.1.2. This network can be seen in Figure 33 and was determined by using equations from ISO 15031-5 found in APPENDIX A. OBD GOVERNING EQUATIONS. The predictive inference accuracy of this expert network is 41.26% and the MSHD/SHD is 0 since it is being compared to itself.

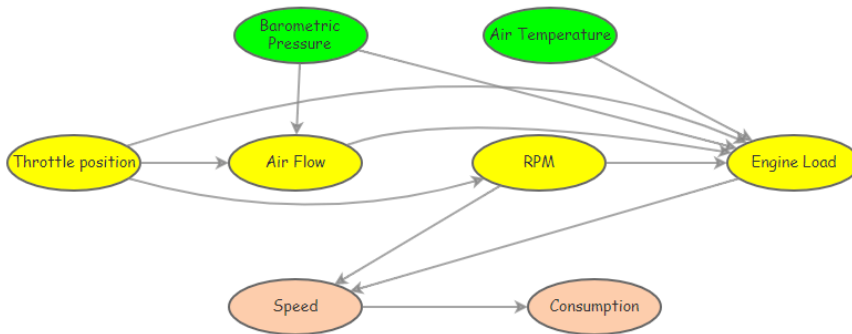


Figure 33: OBD Expert Network

From Figure 25 in section 5.1.1.4 it can be seen that the majority of the BNs learned with the four algorithms have a predictive inference accuracy that falls somewhere between the unconnected “Base” and the “Naïve” Networks. Their accuracies range from 30% to 36.5% which indicates that there is information about the variables states contained within the other variables.

The purely search and score algorithm (K2) tended to perform worse than the approaches based or partially based on conditional independence methods (SCA, MMHC, and PC). This is in line with other researchers findings that indicate that detecting causal relationships is performed better with conditional independence based methods than with search and score approaches.

Commented [MWD78]: discussing ok

The different BNs learned can be combined by utilizing what is called an Average BN approach. An Average BN is achieved by learning the best BN from data and then relearning the network many times with subsamples of the original data. The Average BN is then specified as containing only those edges that were learned in more than 50% of the total graphs [14] [15]. In this research instead of parametric bootstrapping a non-parametric subsampling approach was taken to avoid the issues of overly dense networks that were being learned with the full dataset. In the final average BN all tiers were equally weighted meaning the 128 BNs at the smallest size had the same influence on the average graph as the single BN learned with the full dataset. The threshold was set at 60% instead of 50% because it returned a better scoring structure for the OBD data.

Commented [WM79]: added this section to talk about weighting

When this Average BN approach was taken with the BNs learned with each size OBD dataset the resulting Average BN had a predictive inference ability of 43.03% and a SHD/MSHD of 5 from the expert network. The learned Average BN can be seen in Figure 34 and closely resembles that of the expert network structure other than the two environmental variables.

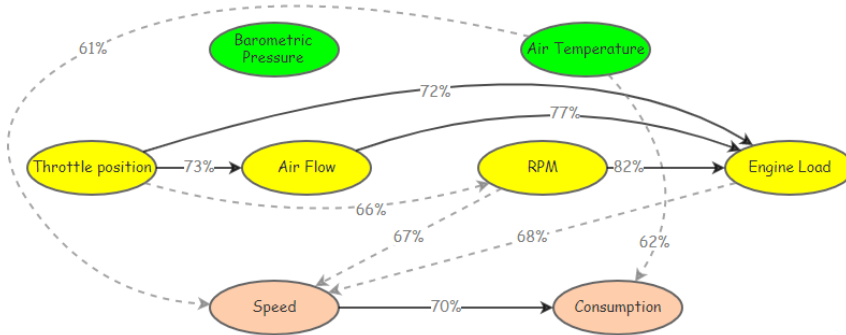


Figure 34: OBD Average BN with a Threshold of 60%

The Average BN is capable of better predictive inference than the expert network and recovers almost the same structure. The only difference is in the effect of the environmental variables which instead of being linked to “Air Flow” and “Engine Load” are linked to “Speed” and “Consumption”. The connections to “Speed” and “Consumption” in fact just barely made the 60% threshold and if the graph is evaluated without those connections it has a predictive inference accuracy of 42.83% and a SHD/MSHD of only 3. This low importance of the environmental variables makes sense when it is considered in the context of its environment, Singapore. Singapore has very little seasonal/daily variation in temperature and barometric pressure meaning that the effects of those two variables will be all but negligible. This result is interesting, because the Average BN has learned a network capable of higher predictive inference accuracy than the expert network and it has taken into account regional specific factors which might not be picked up by an expert unfamiliar with the region.

5.2.4 Divvy Data

The Divvy Dataset represents a dataset where multiple data sources have been combined and the system is believed to have many latent variables. The Causal Sufficiency Assumption, Markov Assumption, and the Faithfulness Assumption are all believed to be untrue for this dataset based on the methodology borrowed from Bowden/Turkington, Lemmer, and Zhang/Spirtes. Since these assumptions are untrue any learned structure should not be interpreted as a causal network according to Pearl [23]. However, the result can still suggest potential causal connections that can be substantiated by other means [23]. The two metrics being used to evaluate the validity of the learned BN are the SHD/MSHD against an expert generated network and the predictive inference accuracy of the BN.

An unconnected BN as seen in Figure 35 establishes a baseline predictive inference accuracy of 34.44% and has SHD/MSHD of 17 when measured against the expert built network. This expert built network was generated by a researcher developing a statistical model for the bike share system in Chicago.

Commented [WM80]: this section was added by me to further explain what I was talking about with the fact that the results, while not causal structures can suggest potential causal connections testable by other means not using observational data.

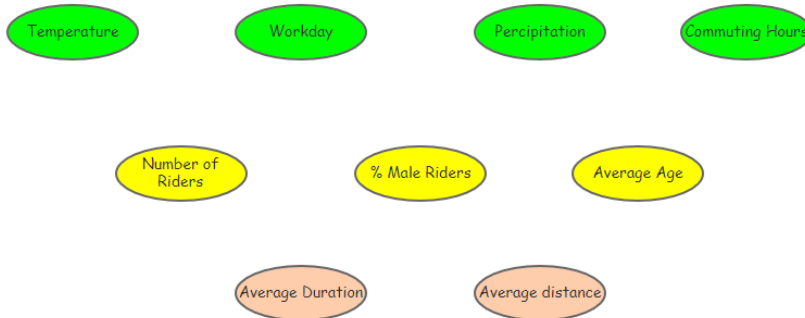


Figure 35: Divvy Unconnected Network

A Naïve BN for the Divvy data will have 37 edges. The Naïve Network is capable of performing predictive inference with an accuracy of 34.39% and has a MSHD/SHD of 20. The accuracy of the Naïve BN is lower than that of the unconnected network, contrary to what was found with the OBD data.

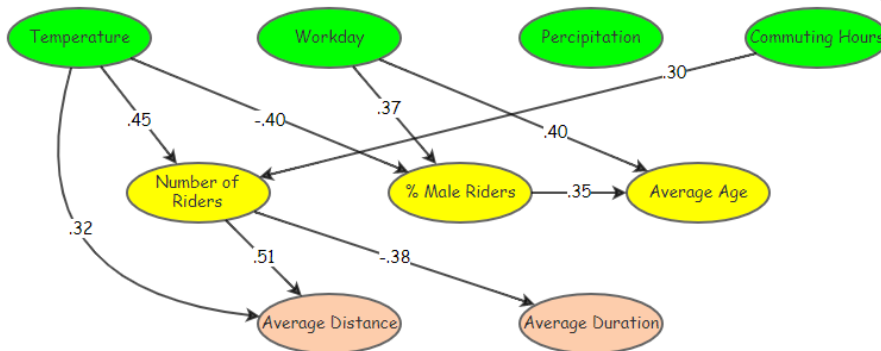


Figure 36: Divvy Correlation Network with a threshold of 0.3 (arcs represent +/- correlation prior to discretization)

A correlation network is the next step and is shown in Figure 36. The edges are directed according to the determined variable ordering for the Divvy data. This correlation network has a predictive inference accuracy of 34.6% and has a MSHD/SHD of 16. The correlation network has slightly better predictive ability than the unconnected network, but still remains within .2% range of the unconnected networks accuracy.

The benchmark expert network introduced in section 4.1.3 can be seen in Figure 37. The predictive inference accuracy of this expert network is 34.12% and the MSHD/SHD is 0 since it is being compared to itself.

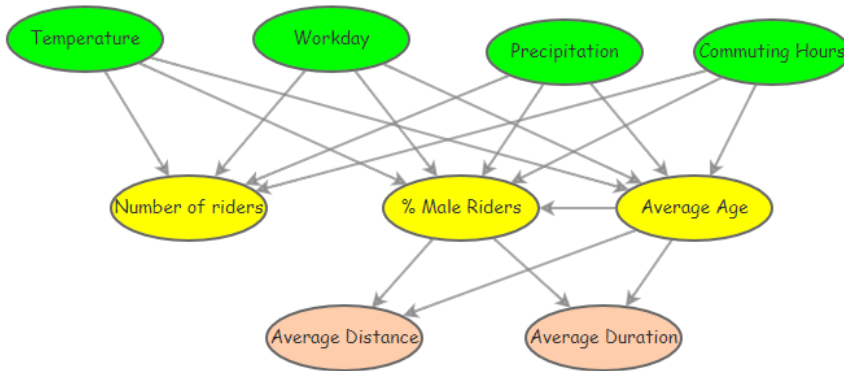


Figure 37: Divvy Expert Network

From Figure 30 in section 5.1.3.4 it can be seen that the majority of the BNs learned with the four algorithms have predictive inference accuracy that falls somewhere below both the unconnected “Base” and the “Naïve” Networks. Their accuracies range from 34.52% to 34.12% which indicates that there is not a lot of information about one variable’s state contained within the other variables in the network. The PC algorithm was the most consistent in terms of performance across the differing sized datasets; however it is difficult

to read too much into the difference between the performances of the algorithms when their results are all within a .5% range of accuracy.

The Average BN approach was taken using the BNs learned with each size Divvy dataset. In the average BN all tiers were equally weighted meaning the 10 BNs at the smallest size had the same influence on the average graph as the single BN learned with the full dataset. The threshold was set at 60% instead of 50% to be consistent with the OBD data's Average BN. The resulting Average BN was capable of a predictive inference accuracy of 34.13% and a SHD/MSHD of 13 from the expert network. The Average BN can be seen in Figure 38.

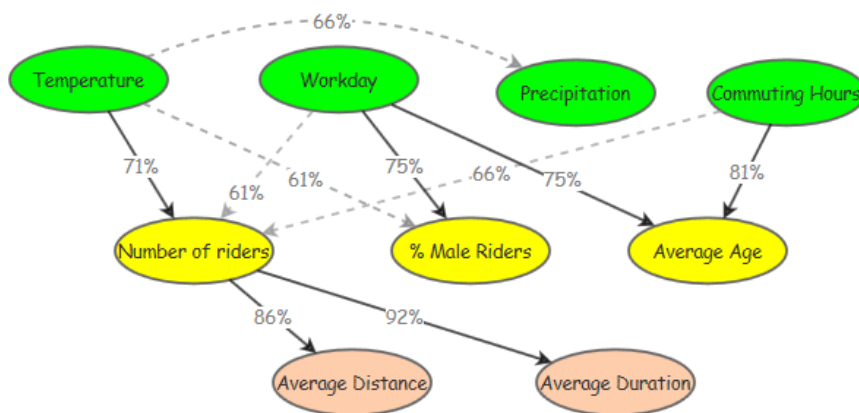


Figure 38: Divvy Average BN

The Average BN in this case is equal in its predictive inference accuracy to the expert network, both of which are below the naïve and unconnected network. Clearly the Divvy dataset is fundamentally different from the other two datasets under investigation. The range of accuracies found was always within a .5% band and the expert and Average

BN were the worst performing in comparison to the OBD data where they were the best performing in terms of predictive inference accuracy. The correlation network was still better than the naïve and unconnected network as was also the case in the OBD data.

5.3 Chapter summary

This chapter reviewed the results of the applied methodology for the ALARM, OBD, and Divvy datasets. The runtime, density of networks, SHD/MSHD, and predictive inference accuracy of the networks learned for each of the four algorithms was presented for each of the three datasets. The results of the three datasets were compared with one another on the basis of runtime, revealing that observational data slowed down the performance of the algorithms that heavily relied on conditional independence (PC and MMHC). The comparison of the densities learned between the different datasets showed that the algorithms tended to learn denser networks as they were given more observational data. This reinforces the idea that most observational datasets will not satisfy the three assumptions underlying effective learning.

For the OBD and Divvy datasets an Average BN was learned using a subsampling approach which averaged all of the learned graphs. These Average BNs were compared with networks that were naïve, unconnected, built by experts, and built based on correlation. For the OBD data the Average BN was capable of the most accurate predictive inference and was similar in structure to the expert specified network (which the Average BN outscored on inference). The unconnected, naïve, correlation, expert and Average BN for OBD had predictive inference accuracies ranging from 31.7% to 43.03% for the unconnected an Average BN respectively. The same comparisons were made for the Divvy

Average BN and the predictive inference accuracies ranged from 34.12% to 34.6% for the expert and correlation networks respectively. The low variation in predictive ability of the different networks supports the idea that there is not enough information in the Divvy dataset to make strong predictions.

CHAPTER 6. DISCUSSION

This chapter explains the assumptions in the three datasets and provides a summary of the results of the three datasets. It seems that analysis of observational data obtained from sensors and digital products, such as the Divvy and OBD data, requires a number of advancements. These advancements primarily relate to determining all the latent variables influencing systems and how variables are discretized once they are included in the training datasets. This chapter will also summarize how the Average BN learned from Divvy system data can add insights beyond what is provided by a simple correlation based network. The connections in the Divvy data found by the Average BN are examined using standard graphing techniques and support is found for all the learned relationships.

6.1 Data Assumptions and Algorithms

Table 11 shows the three datasets and their characteristics. It also provides the difference in predictive accuracies compared to an unconnected BN. As per the methodologies borrowed, the ALARM dataset satisfied all three assumptions, the OBD data partially fulfilled the Causal Sufficiency Assumption, and the Divvy data did not satisfy any of the assumptions. Since the ALARM data are generated by a BN, it is very unlikely that the samples would misrepresent the network structure and not satisfy the three assumptions underlying structure learning [40]. The ALARM network data thus produced the most accurate predictive inference and agreement with the true network. The OBD data provide information for all of the variables in the equations that calculate the different engine parameters. Additionally, portions of the network were deemed valid when evaluated using a method based around instrumental variables, so it partially satisfied the

Causal Sufficiency Assumption. The OBD performed predictive inference with accuracy significantly better than an unconnected network suggesting there was enough information in the network to inform the states of the variables in the network. The Markov Assumption and the Faithfulness Assumption were deemed invalid based on the methodology for characterizing datasets.

Table 11: Assumptions and Evidence

	ALARM	OBD	Divvy
Causal Sufficiency	True	Partially True	False
Evidence	Simulated	Instrumental Variables	Instrumental Variables
Markov Assumption	True	False	False
Evidence	Simulated	Lemmer's Signal Methodology	Lemmer's Signal Methodology
Faithfulness Assumption	True	False	False
Evidence	Simulated	Sub-condition	Sub-condition
Accuracy	Unconnected + 18.3%	Unconnected + 11.33%	Unconnected - .31%

For the Divvy data the Causal Sufficiency Assumption was found to be invalid based on the methodology around instrumental variables. There are a myriad of other factors that affect the utilization of the bike share system and only a small subset were included in this dataset, so this result is not unexpected. The results of the structure learning algorithms also supports this statement because all of the network structures had similarly poor predictive inference accuracy. The small variation in accuracies between the networks and other approaches indicates that the variables did not contain a lot of information about one another absent other informative latent variables. The Markov Assumption was shown

to be invalid according to Lemmer's methodology built around signals and as a result the Faithfulness Assumption could not be true. Should the Markov Assumption been valid then the methodology from Zhang/Spirtes would have been used [3].

The partial satisfaction of the Causal Sufficiency Assumption appears to be enough to perform data based discovery and predictive inference with accuracy above that of a naïve, unconnected, and correlation based network, assuming an Average BN approach is taken. If seeking to learn models capable of data based discovery and accurate predictive inference for sociotechnical systems like Divvy researchers should focus on finding and including latent variables present in the network to the datasets such that the methodology built around Instrumental Variables implies causal sufficiency.

6.1.1 Other contributing factors

Average BNs learned using sub sampling of the original two observational datasets provided additional insight. In the case of the OBD data the Average BN provided a better network as measured by predictive inference accuracy and SHD/MSHD from an expert network than any of the individual BNs learned. By averaging across the subsample learned structures, a model that avoided the issue of overfitting found in the individual sub samples and the issue of fully connected networks found in the networks learned with the entire dataset was learned. With the Divvy data the Average BN approach avoided the fully connected networks learned with the entire dataset, but did not present predictive inference accuracy significantly different than any of the individual networks learned. However, all the learned networks were within a .5% range for accuracy of predictive inference. The results with these two observational datasets lend support to the idea that an Average BN

approach is valuable for observational data in complex systems predicated on the condition that most significant variables are known and observed.

With better data discretization the OBD data might be able to better satisfy the Markov Assumption and Faithfulness Assumption and improve its predictive inference accuracy. Ideally one would have a high number of discrete states which would allow for linear relationships to be captured appropriately [81]. If these linear relationships were captured then the independencies in the true structure would likely become more evident in the generated data. However, increasing the number of discrete states would require significantly more training data and there is a limit to how much can be effectively processed. Beyond increasing the number of discrete states the ranges of the discrete states could be determined as part of the learning process rather than being formulated by grouping into tertiles. BN learning approaches which explore possible discretizations during the learning process exists and could be applied in this case [82]. By modifying the ranges used for discretization rather than increasing the number of discretizations one can take advantage of any nonlinear aspects of the variables' relationship. Similar arguments can be made for the Divvy dataset assuming the Causal Sufficiency Assumption was better satisfied.

6.2 Bike share Case Study

While the Divvy data did not provide predictive accuracy at the level of the OBD or ALARM data, future efforts and modifications to the algorithms and preprocessing procedures may provide more accurate models. This subsection examines the resulting

Divvy graphs in more detail to demonstrate how future graphs might be used to come up with sustainable design insights.

6.2.1 Learned relationships

The design goal was to maximize the utilization of the bike share which equates to increasing the “Number of Riders” node, therefore the research looked at the variables that are parents of this node in the causal diagram. The research was also interested in seeing how the “Number of Riders” could be increased by increasing utilization among women who have traditionally been under represented. Therefore, variables that were parents of both the “Number of Riders” and the “% Male Riders” are of specific interest.

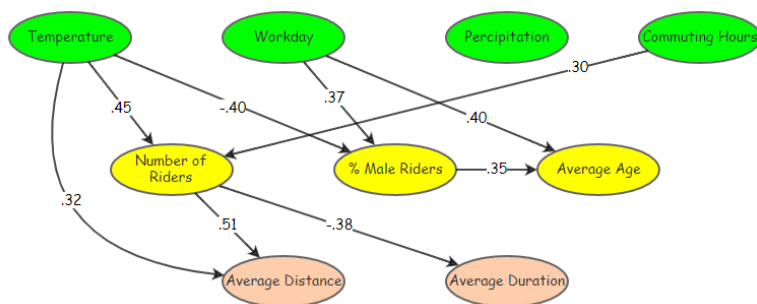


Figure 39: Divvy Correlation Network with a threshold of 0.3 (arcs represent +/- correlation prior to discretization)

A preliminary system diagram generated using significant correlation to define edges is shown in Figure 39. In the preliminary system diagram the two variables that influence the “Number of Riders” node are the “Commuting Hours” and “Temperature” nodes. The commuting hours is something that is outside of the bike share designers’ control. The second variable that the preliminary network identified as connected is the

Commented [MWD81]: changed

Commented [MWD82]: changed

“Temperature” node. The “Temperature” node is also connected to the “% Male Riders” node making it a variable of interest. Thus, the relationship between temperature and gender diversity should be investigated by designers to see how they might modify or otherwise make use of the relationship.

The validity of the temperature correlations was verified using standard graphing techniques. In Figure 40 a relationship between the gender ratio of riders and the average temperature was found. This graph was generated by looking at any time during the year where there was a specific gender ratio of riders, say half men and half women, and the averaging the temperature across every instance that the same gender ratio was observed. Figure 40 has a point for every unique gender ratio and the average temperature at which that ratio occurred. The graph indicates that below 60 degrees Fahrenheit there are often more male riders (towards the left). Above 60 degrees the ratio tends to be less strongly skewed towards male riders (more observations towards 20-50% women).

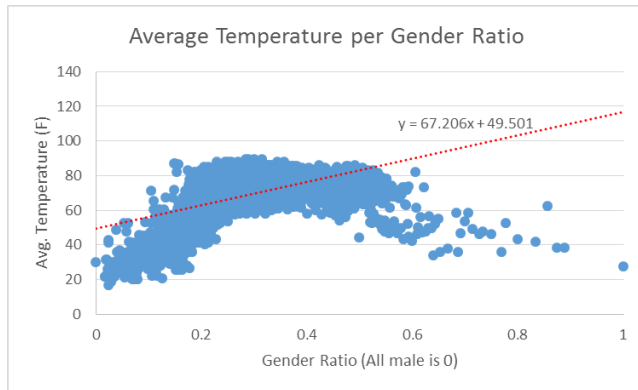


Figure 40: Average Temperature per Gender Ratio

The relationship between the temperature and the number of riders was also investigated and is presented in Figure 41. Similar to the previous graph, each point represents a specific number of riders and the average of the temperature observed any time that number of people was riding. The number of riders increases as the temperature increases and that below sixty degrees there are significantly fewer riders.

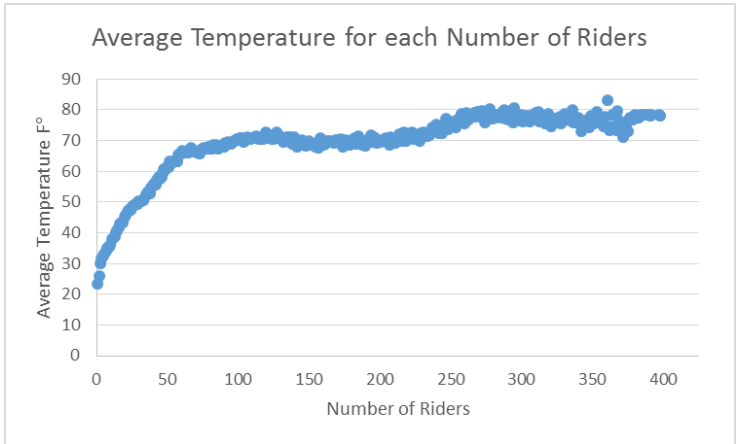


Figure 41: Average Temperature for each number of Riders

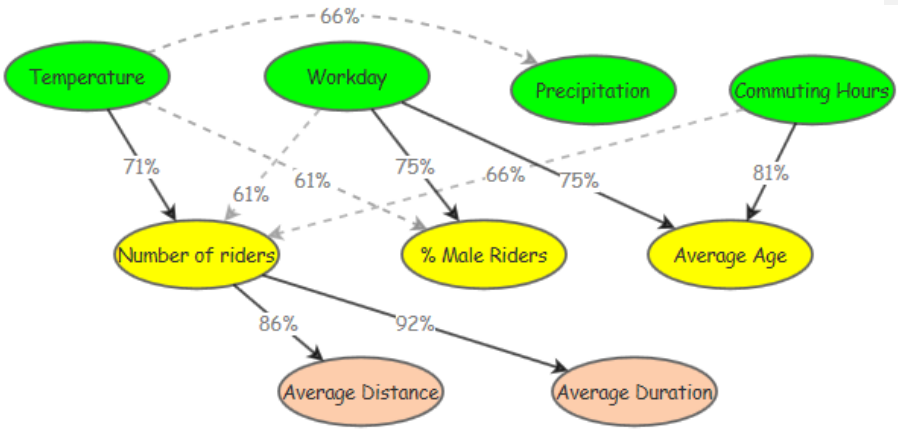


Figure 42: Average BN Structure Divvy

The Average BN in Figure 42 closely resembles the preliminary diagram based on correlation, except that it does not connect “% Male Riders” and “Average Age” since the algorithms recognized that the two are conditionally dependent through the “Workday”

variable. The algorithms also learned three more connections in the data than were found by the simple correlation method. These connections are “Commuting Hours” with “Average Age”, “Workday” to “Number of Riders”, and “Temperature” to “Precipitation”. All three of these connections make intuitive sense and two out of three are included in the expert made network. The “Workday” to “Number of Riders” connection is interesting because it affects both the “% Male Riders” and the “Number of Riders” and suggests another potential area to look for design solutions in addition to the “Temperature” variable.

The “Workday” variable’s relevance was also investigated using standard graphing techniques and the results support the structure of the Average BN. Figure 43 is a graph that relates the gender ratio and the percentage of the time that gender was observed on a workday. The graph shows that there is a bias towards a male dominated gender ratio on workdays, which is what was indicated by the Average BN.

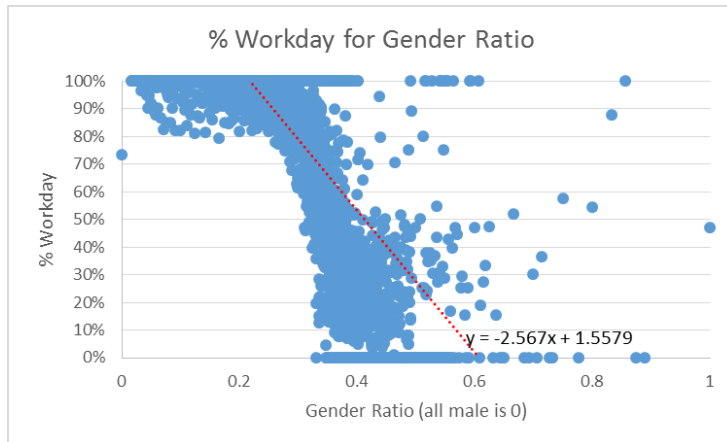


Figure 43: % Workday vs Gender Ratio

The graph in Figure 44 shows for each number of riders what percentage of the time that number of riders was on a workday. The graph indicates that there were never over 210 people simultaneously riding bikes when it was not a workday. There is a zig zag formation when going from zero riders to 150 riders. The reason for this relationship is not known at this time and warrants further investigation.

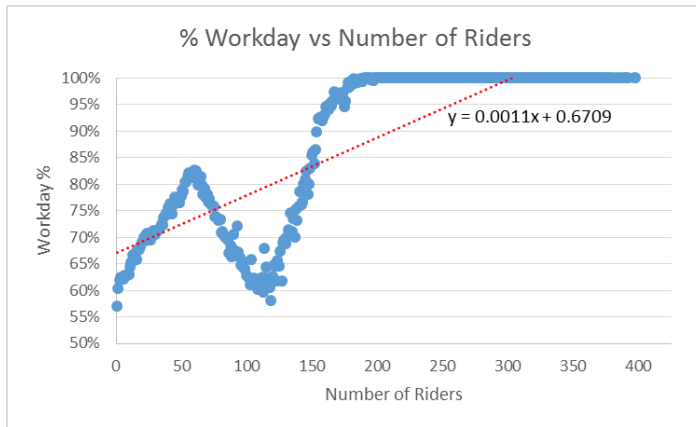


Figure 44: % Workday vs Number of Riders

The resulting insights from the two networks indicate that the “Workday” and the “Temperature” should be investigated. Both of these variables complete paths between the “Number of riders” and the “% Male Riders” nodes. The “Workday” connection was only revealed through the Average BN approach. The failure to perform accurate predictive inference suggests that the number of variables currently used for modeling the bike share is insufficient.

6.2.2 *Qualitative Verification of Learned Relationships*

The relationships learned through both the BN structure learning approaches (without the three assumptions being valid) and the simple correlation approach are not causal. They may provide an indication of what causal connections exist, but cannot prove that assertion. In order to prove a causal connection, qualitative and experimental means must be employed. One potential qualitative approach for verifying the existence of causal connections is to survey users of the Divvy system. The corollary relationships learned

with the BN structure learning algorithms can be used to frame questions to riders. A network with particular node labels omitted could be presented to riders who would be tasked with filling in the missing variable. The network could be given to riders tasked with labelling latent variable nodes, the location of which can be inferred by the structure of the network. Such latent variables might include the presence of ice on the roads, the amount of baggage someone has with them, or whether a person has children with them. The true number of latent variables is potentially very large, but will definitely include station level effects and a more comprehensive segmentation of riders based on reason for utilization. One variable that is believed to exist is a metric gauging the enjoy ability of ride which should have the distance and duration of rides as children as well as the number of riders. This variable will likely have weather and workday variables as parents. As more latent variables are identified and the size of the graph increases it will become more and more computationally intensive and some of the exact methods will become intractable, further limiting our learning approaches [19].

6.3 Chapter Summary

This chapter compares the performance of the learning algorithms with different datasets, each with different assumptions being valid. It then indicates that further work on identifying latent variables in the network will improve the performance of the structure learning algorithms. Secondary to the identification of the latent variables in a system is the proper discretization of the variables present in the system. Both will enhance the learned structures performance. The chapter also indicated that with the OBD observational data the algorithms with conditional independence aspects tended to perform better in predictive inference. So the primary insight into data for real world systems is that it must

contain all of the variables in question, thereby satisfying the Causal Sufficiency Assumption. The primary insight into the types of algorithms was that explicit tests of independence had value for the observational data and learning diagrams that inform causal relationships.

Commented [MWD83]: removed

Commented [MWD84]: added

The chapter indicates that utilizing an Average BN approach with sub sampling can avoid the issue of learning fully connected networks when training with large samples of observational data. The validity of the Average BN for the Divvy system was also examined and it was found that the connections drawn between gender ratio and number of riders through the intermediate variables workday and temperature were supported when further investigated by other means. The corroboration of the statistical check of the Average BN learned connections supports the idea that Average BNs and BNs can be used to inform design decisions on a system level.

CHAPTER 7. CONCLUSIONS AND FUTURE WORK

This research has highlighted some of the difficulties in utilizing BN structure learning for creating models capable of providing design insight into sociotechnical systems. This chapter serves to summarize and address the work that still needs to be completed to make BN structure learning a useful tool for sociotechnical system design. There exist both observational sociotechnical data and computational resources capable of learning from data. However, this sociotechnical data often does not satisfy the assumptions that underlie learning causal networks. A methodology is needed for characterizing datasets a priori and a guide for selecting a causal structure learning algorithm and post-processing approach based on that characterization. As illustrated in Figure 45, this thesis has started in the direction of learning real world models and has determined the requirements for the data used for learning real world models and the requirements of the algorithms that will do the learning.

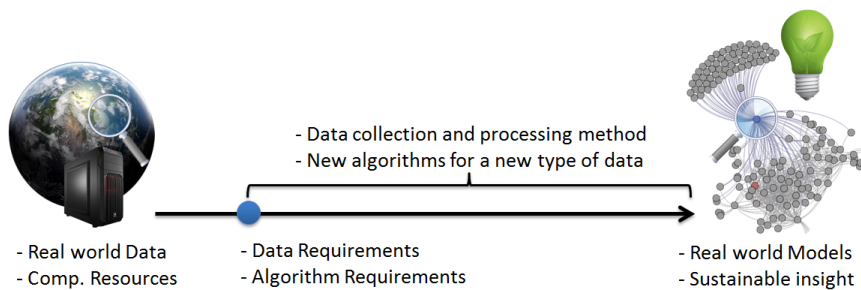


Figure 45: Research Direction

The data must satisfy the three assumptions underlying BN structure learning as described by the methodologies of Bowden/Turkington, Lemmer, and Zhang/Spirtes

[1][2][3]. From the OBD Average BN one can conclude that at least partial satisfaction of the Causal Sufficiency Assumption as outlined by Bowden/Turkington is sufficient to perform predictive inference and knowledge discovery [1][46]. That conclusion means satisfying the Markov and Faithfulness Assumptions is secondary to satisfying the Causal Sufficiency Assumption and researchers looking to improve their ability to learn causal models should focus on the discovery of latent variables.

Commented [MWD85]: changed

Conditional independence and hybrid approaches were capable of higher predictive inference accuracies than the purely score based approach with the OBD dataset. The predictive inference of the different classes of algorithms in the Divvy dataset was deemed to contain little insight since it only varied by .5%. The MSHD from expert networks with the two observational datasets was inconclusive; with the search and score approach being closer to the OBD expert network and the conditional independence approaches being closer to the Divvy expert network. From this evidence a tentative conclusion can be drawn to say that the conditional independence and hybrid methods are better for the observational data, where fewer assumptions hold true, because they have implicit tests of independence. These independence test based methods have historically been cited as better for determining causal connections and this research would support that claim based on the OBD data findings [49][16].

Commented [MWD86]: speaking theoreticallu

This work concludes in saying that data used for learning real world models that provide insight into systems causal structure must at least partially satisfy the Causal Sufficiency Assumption by containing all the relevant variables for an area of the causal diagram as determined by instrumental variables. And that algorithms based at least partially on tests of independence are better for determining real world causal relationships.

Commented [MWD87]: added to be less strong

The results of those algorithms can be further improved by averaging over graphs learned with different size sub samples to arrive at an Average BN.

7.1 Future work

Future work in this research area consists of two main pushes. The first push is to continue to use the dataset characterization methodology outlined to characterize observational datasets. With each characterized dataset, a litany of algorithms and post-processing techniques should be used and their results compared. This will further develop the guidelines for selecting a causal structure learning algorithm for a given dataset. Such a guide would benefit anyone attempting to use causal structure learning with complex datasets. Other datasets that might be evaluated are some of the real life datasets already used in the BN structure learning community including the iris, balance, thyroid, liver, ecoli, abalone, diabetes, post operative, yeast, breast cancer, shuttle, tic tac toe, bc wisconsin, glass, page blocks, heart cleveland, heart hungarian, heart statlog, wine, and adult datasets might be characterized and evaluated with different algorithms [83][84]. These datasets have been used to compare BN structure learning algorithms in the past and would have many points of comparison available in current literature [56][84][85]. Other algorithms should also be investigated including the IC*, FCI, TPDA, KS, LCD2, GS, and IAMB algorithms because of their different mathematical principles and because they can all be easily accessed using the same causal explorer toolkit used in this research [75][31].

The second area of interest is to learn how to process datasets such that they better satisfy the assumptions underlying structure learning algorithms. In order to better satisfy the three assumptions discussed by Margaritis such processing will have to detect and

discover latent variables to ensure complete system understanding [42]. This dataset processing will also discretize datasets so as not to lose information contained within. Approaches for addressing these two issues is discussed in detail in sections 7.1.1 and 7.1.2.

7.1.1 Latent Variable Detection

An immediate area for development is detecting and accounting for latent variables in datasets. Accounting for all of the latent variables present in a system will satisfy the Causal Sufficiency Assumption [34]. With this assumption satisfied the BN structure learning algorithms will perform well enough to recover graphs that can inform causal connections and perform more accurate predictive inference. There are many approaches for detecting latent variables some of which are qualitative and others quantitative. One qualitative indicator of a latent variable is a connection made between variables that cannot logically be related in a causal manner. Such a connection indicates that there is a latent variable unaccounted for. One such example is the connection between the “Number of riders” and “Average Distance”. These variables are not related because an increase in the number of bikes checked out will increase availability of empty docks. This connection implies that there is a common shared cause between the “Average Distance” and “Number of riders”. A researcher should investigate other variables that explain this relationship through surveys and observations.

Quantitative approaches to finding latent variables utilize variations on the algorithms discussed in this work [27] [86] [87]. Nir Friedman has also done work with an augmented version of the Structural Estimation Maximization (SEM) algorithm [88]. This structure based approach to detecting latent variables works by identifying “cliques”. In

Commented [MWD88]: made les strong

this context a “clique” is a local structure, usually dense, that could be greatly simplified by the addition of another node [86]. In Figure 46 the left side image represents a “clique” and the right-hand side represents that same “clique” with the addition of a single missing variable. This variable significantly simplifies the structure and improves the score of the network as measured by information theoretic and Bayesian scoring metrics.

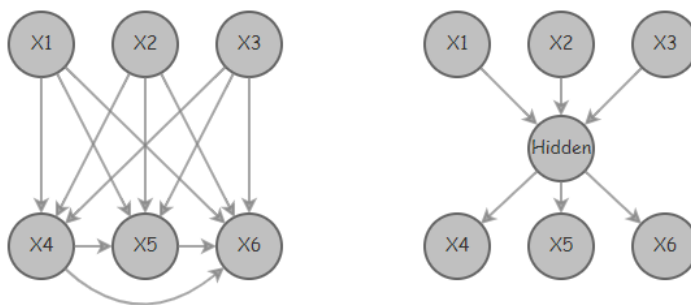


Figure 46: Clique and clique with latent variable

Further research will investigate both these qualitative and quantitative methods of detecting latent variables in networks that will be used to inform systems causal relationships. This means using BN structure learning techniques to infer the location of latent variables and then drawing upon qualitative survey data to identify that latent variable. The presence of further latent variables can be investigated using the method built around instrumental variables. By identifying relevant latent variables the validity of the Causal Sufficiency Assumption can be improved and the ability to learn structure and perform inference greatly increased.

Commented [C89]: This is as far as I got reading the Dr. Fu responses.

WM – Ok I will try to extrapolate to the rest of the comments, you have already given me so much feedback, thank you! I know you are frustrated I will work harder

Commented [WM90]: Updated with the comments that you provided!

Commented [MWD91]: made less strong

7.1.2 Data Discretization

Discretizing continuous data for use in BN learning is known to be problematic, especially when the data relationships are linear [89]. If continuous variables are discretised into only a few intervals, then more dependencies are likely to be found than if the continuous variable is divided into many intervals. To avoid this overfitting, each interval to have a sufficient number of observations [90]. The results of the learning performed with discretized data will also change when the break points that describe discretization are changed [89]. The discretization chosen must be a valid representation of the state space for the learned network to be defensible [91].

Automated data discretization methods have been developed in the past [92][82] however, no satisfactory method has yet been determined [89]. As such the most common and successful approach currently is to rely on experts to specify the discretization of continuous variables [90]. Future work will aim to explore which of the current discretization learning approaches are the most applicable for the bike share system. There is also the possibility of developing a novel discretization algorithm that will provide a better pre-processing of the data and more insightful results from the subsequently learned structures.

7.2 Limitations

There are a few limitations in this work including the inability to discover cyclic networks and the inability to know exactly what assumptions are satisfied by the dataset. The methodology built from combining the work of Bowden/Turkington, Lemmer, and

Zhang/Spirtes is not complete in that it can often not tell exactly what portion of the system is causing the violation of the assumption or how to amend it.

[The inability to utilize cyclic graphs precludes the possibility of positive and negative feedback loops which are very likely present in real world systems. Negative feedback loops allow for systems to regulate themselves and maintain a stable operating state which is one of the emergent qualities of sustainability. Positive feedback loops result in the amplification of effects. Positive feedback is also an emergent property that relates to system sustainability and is likely to increase rapidly and result in behavior change for the system. The current work does not show how feedback loops can change the system over time, since it is just an instantaneous snapshot. The acyclic assumption makes the problem tractable, however, and can be relaxed or revisited to add cyclic chains. Dynamic BNs present a way to capture the cyclic nature of systems as a series of acyclic graphs and could be used in future works [52].]

The other limitation of this work is in the a priori characterization of the datasets at hand. The processes developed by Bowden/Turkington, Lemmer, and Zhang/Spirtes are incomplete in a few respects. For the causal Sufficiency Assumption the method of instrumental variables first developed by Bowden and Turkington can only help reveal the presence of latent variables in that it breaks down with nonlinear relationships as noted by Pearl [46]. Additionally in some scenarios, it can be very difficult to determine an appropriate instrumental variable to help make the subjective assessment of the presence of latent variables or cofounders [1]. The issue that exists with the work of Lemmer in determining the applicability of the Markov Assumption is that often times breaking down events into the prescribed single cause level is impractical and puts the data beyond

Commented [TC92]: Ok. This discussion isn't complete. Provide some thought on the limitation of acyclic as well as what is still useful about acyclic.

WM – limitation is the inability to capture feedback, however it still provides an instantaneous snapshot of the system and dynamic Bayesian networks can be used to address the feedback loops

Commented [C93R92]: One could have cyclic connections and be static.

WM - yes that is very true! "The current work does not show how feedback loops can change the system over time, since it is just an instantaneous snapshot"

measurement by any passive means [2]. Finally, the problem with the Zhang/Spirtes methodology for determining Faithfulness of a dataset is that the method presupposes that the Causal Sufficiency Assumption and Markov Assumption are valid [3].

APPENDIX A. OBD GOVERNING EQUATIONS

In this appendix the equations utilized to determine the expert network are reviewed. The first equation was the one utilized for the engine load variable [93].

$$Load\% = \frac{\text{current air flow}}{(\text{peak air flow WOT@STP as a function of RPM}) * \left(\frac{\text{Barometric}}{29.92}\right) * \sqrt{\left(\frac{298}{AAT + 273}\right)}}$$

- STP = Standard Temperature and Pressure = 25 °C, 29.92 in Hg Barometric,
- SQRT = square root
- WOT = wide open throttle
- AAT = Ambient Air Temperature (in °C)

The speed of the vehicle is a function of the RPM and what gear the vehicle is in. The gear will be reflected in the Engine load. So the Engine load and the RPM should be indicative of the speed of the vehicle [94].

$$\frac{(RPM * r)}{(R_1 * R_2)} = Vehicle\ speed$$

- RPM = engine speed, in revolutions/minute
- r = loaded tire radius (wheel center to pavement)
- R1 = transmission gear ratio
- R2 = rear axle ratio

Fuel consumption for the dataset was calculated when the data was originally compiled and was based off of the speed according to the data originators in Singapore.

$$Instantaneous\ Consumption = Speed * fuel\ efficiency\ factor$$

REFERENCES

- [1] R. J. Bowden and D. A. Turkington, *Instrumental variables*. Cambridge University Press, 1990.
- [2] J. F. Lemmer, "The causal Markov Condition, Fact or Artifact?," *ACM SIGART Bull.*, vol. 7, no. 3, pp. 3–16, 1996.
- [3] J. Zhang and P. Spirtes, "Detection of unfaithfulness and robust causal inference," *Minds Mach.*, vol. 18, no. 2, pp. 239–271, 2008.
- [4] S. Long, *Socioanalytic Methods: Discovering the Hidden in Organisations and Social Systems*. London: Karnac Books Ltd, 2013.
- [5] M. Dinar, J. J. Shah, Y. Park, and P. Langley, "PATTERNS OF CREATIVE DESIGN: PREDICTING IDEATION FROM PROBLEM FORMULATION," in *Proceedings of the ASME 2015 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference*, 2015, pp. 1–10.
- [6] G. H. Brundtland, "Report of the World Commission on environment and development: 'our common future,'" 1987.
- [7] S. Greenland, J. Pearl, and J. M. Robins, "Causal Diagrams for Epidemiologic Research," *Epidemiology*, vol. 10, no. 1, pp. 37–48, 1999.
- [8] N. Friedman, I. Nachman, and D. Peér, "Learning bayesian network structure from massive datasets: the 'sparse candidate' algorithm," *Proc. Fifteenth Conf. Uncertain. Artif. Intell.*, vol. 0, pp. 206–215, 1999.
- [9] S. Nadkarni and P. P. Shenoy, "Bayesian network approach to making inferences in causal maps," *Eur. J. Oper. Res.*, vol. 128, no. 3, pp. 479–498, 2001.
- [10] P. Weber, G. Medina-Oliva, C. Simon, and B. Iung, "Overview on Bayesian networks applications for dependability, risk analysis and maintenance areas," *Eng. Appl. Artif. Intell.*, vol. 25, no. 4, pp. 671–682, 2012.

- [11] J. Fiksel, “Sustainability and Resilience: Toward a Systems Approach,” *Sustain. Sci. Pract. Policy*, vol. 2, no. 2, p. 2006, 2006.
- [12] P. Weber, G. Medina-Oliva, C. Simon, and B. Iung, “Overview on Bayesian networks applications for dependability, risk analysis.pdf,” *Eng. Appl. ...*, 2012.
- [13] R. Hansmann, H. A. Mieg, and P. Frischknecht, “Principal sustainability components: Empirical analysis of synergies between the three pillars of sustainability,” *Int. J. Sustain. Dev. World Ecol.*, vol. 19, no. 5, pp. 451–459, 2012.
- [14] B. J. J. McCormick, L. K. Van Breda, and M. P. Ward, “Bayesian Network analysis of piglet scours,” *Sci. Rep.*, vol. 7, no. 1, p. 6202, 2017.
- [15] F. I. Lewis and B. J. J. McCormick, “Revealing the complexity of health determinants in resource-poor settings,” *Am. J. Epidemiol.*, vol. 176, no. 11, pp. 1051–1059, 2012.
- [16] S. Acid, L. M. De Campos, J. M. Fernández-Luna, S. Rodríguez, J. María Rodríguez, and J. Luis Salcedo, “A comparison of learning algorithms for Bayesian networks: A case study based on data from an emergency medical service,” *Artif. Intell. Med.*, vol. 30, no. 3, pp. 215–232, 2004.
- [17] L. E. Brown, I. Tsamardinos, and C. F. Aliferis, “A novel algorithm for scalable and accurate Bayesian network learning,” *Stud. Health Technol. Inform.*, vol. 107, pp. 711–715, 2004.
- [18] C. Shalizi, “Discovering Causal Structure from Observations,” pp. 477–492, 2013.
- [19] T. Silander and P. Myllymäki, “A simple approach for finding the globally optimal Bayesian network structure,” *Networks*, pp. 445–452, 2006.
- [20] P. Spirtes, C. Meek, and T. Richardson, “Causal Inference in the Presence of Latent Variables and Selection Bias,” *Proc. Elev. Conf. Uncertain. Artif. Intell.*, pp. 499–506, 1995.
- [21] N. Friedman and D. Koller, “Being Bayesian about network structure. A Bayesian approach to structure discovery in Bayesian networks,” *Mach. Learn.*, vol. 50, pp. 95–125, 2003.

- [22] “Divvy Data.” [Online]. Available: <https://www.divvybikes.com/system-data>. [Accessed: 01-Jan-2017].
- [23] J. Pearl and T. S. Verma, “A Theory of Inferred Causation,” *Stud. Log. Found. Math.*, vol. 134, pp. 789–811, 1995.
- [24] B. Bras, “Recurring and Unresolved Problems in Sustainable Design,” in *Volume 5: 25th International Conference on Design Theory and Methodology; ASME 2013 Power Transmission and Gearing Conference*, 2013, p. V005T06A044.
- [25] M. A. Babyak, “What You See May Not Be What You Get: A Brief, Nontechnical Introduction to Overfitting in Regression-Type Models,” *Psychosom. Med.*, vol. 66, no. 3, pp. 411–421, 2004.
- [26] F. W. Geels, “From sectoral systems of innovation to socio-technical systems: Insights about dynamics and change from sociology and institutional theory,” *Res. Policy*, vol. 33, no. 6–7, pp. 897–920, 2004.
- [27] N. Friedman, “The Bayesian Structural EM Algorithm,” *Proc. Fourteenth Conf.*, pp. 129–138, 2013.
- [28] O. Goudet *et al.*, “LEARNING FUNCTIONAL CAUSAL MODELS WITH GENERATIVE NEURAL NETWORKS,” *North*, 2017.
- [29] C. F. Aliferis CONSTANTINALIFERIS, A. Statnikov ALEXANDERSTATNIKOV, S. Mani, and X. D. Koutsoukos, “Local Causal and Markov Blanket Induction for Causal Discovery and Feature Selection for Classification Part I: Algorithms and Empirical Evaluation Ioannis Tsamardinos,” *J. Mach. Learn. Res.*, vol. 11, pp. 171–234, 2010.
- [30] “Markov blanket.” [Online]. Available: https://en.wikipedia.org/wiki/Markov_blanket. [Accessed: 01-Jan-2017].
- [31] R. Daly, Q. Shen, and S. Aitken, “Learning Bayesian networks: approaches and issues,” *Knowl. Eng. Rev.*, vol. 26, no. 2, pp. 99–157, 2011.
- [32] J. Schmidhuber, “Deep Learning in neural networks: An overview,” *Neural Networks*, vol. 61, pp. 85–117, 2015.

- [33] “Artificial neural network.” [Online]. Available: https://en.wikipedia.org/wiki/Artificial_neural_network. [Accessed: 01-Jan-2017].
- [34] T. J. T. Koski and J. Noble, “A Review of Bayesian Networks and Structure Learning,” *Math. Appl.*, vol. 40, no. 1, pp. 51–103, 2012.
- [35] “Bayesian Networks,” 2015. [Online]. Available: <https://rationalistramble.wordpress.com/2015/12/09/bayesian-networks/>. [Accessed: 01-Jan-2017].
- [36] B. J. J. McCormick, M. J. Sanchez-Vazquez, and F. I. Lewis, “Using Bayesian networks to explore the role of weather as a potential determinant of disease in pigs,” *Prev. Vet. Med.*, vol. 110, no. 1, pp. 54–63, 2013.
- [37] N. Friedman, M. Linial, I. Nachman, and D. Pe’er, “Using Bayesian Networks to Analyze Expression Data,” *J. Comput. Biol.*, vol. 7, no. 3–4, pp. 601–620, 2000.
- [38] M. Pittavino *et al.*, “Comparison between generalized linear modelling and additive Bayesian network; identification of factors associated with the incidence of antibodies against *Leptospira interrogans* sv Pomona in meat workers in New Zealand,” *Acta Trop.*, vol. 173, no. May, pp. 191–199, 2017.
- [39] J. O. Berger, B. Boukai, and Y. Wang, “Unified frequentist and Bayesian testing of a precise hypothesis,” *Stat. Sci.*, vol. 12, no. 3, pp. 133–160, 1997.
- [40] R. Scheines, “An introduction to causal inference,” *Canadian Entomologist*. 1997.
- [41] R. Lienhart, “Markov Equivalence,” *Multimedia Computing, Universität Augsburg*. [Online]. Available: http://www.multimedia-computing.de/mediawiki/images/5/55/SS08_BN-Lec2-BasicProbTheory_3.pdf.
- [42] D. Margaritis, S. Thrun, C. Faloutsos, A. W. Moore, and G. F. Cooper, “Learning Bayesian Network Model Structure from Data,” *Learning*, no. May, 2003.
- [43] J.-P. Pellet, “Finding Latent Causes in Causal Networks : an Efficient Approach Based on Markov Blankets Supplemental Material : Proofs,” *Mach. Learn.*, vol. 1, no. iii, pp. 1–3, 2004.

- [44] and R. S. Peter Spirtes, Clark Glymour, *Causation, Prediction, and Search*. Cambridge: The MIT Press, 2000.
- [45] J. Pearl, “Causality: Models, Reasoning, and Inference,” *Econom. Theory*, vol. 19, pp. 675–685, 2000.
- [46] J. Pearl, “Mediating Instrumental Variables,” no. December, pp. 1–10, 1993.
- [47] A. P. Dawid, “Beware of the DAG!,” *NIPS 2008 Workshop Causality*, pp. 59–86, 2008.
- [48] R. W. Robinson, “Counting unlabeled acyclic digraphs,” *Comb. Math.*, vol. 622, pp. 28–43, 1977.
- [49] D. Heckerman, “A Bayesian approach to learning causal networks,” *Proc. Elev. Conf. Uncertain. Artif. Intell.*, pp. 285–295, 1995.
- [50] I. Tsamardinos, L. E. Brown, and C. F. Aliferis, “The max-min hill-climbing Bayesian network structure learning algorithm,” *Mach. Learn.*, vol. 65, no. 1, pp. 31–78, 2006.
- [51] et al. Beinlich, Ingo A., “The ALARM monitoring system: A case study with two probabilistic inference techniques for belief networks,” *AIME*, vol. 89, pp. 247–256, 1989.
- [52] S. Kim, S. Imoto, and S. Miyano, “Dynamic Bayesian network and nonparametric regression for nonlinear modeling of gene networks from time series gene expression data,” *BioSystems*, vol. 75, no. 1–3, pp. 57–65, 2004.
- [53] D. Heckerman, D. Geiger, and D. M. Chickering, “Learning Bayesian Networks: The Combination of Knowledge and Statistical Data,” *Mach. Learn.*, vol. 20, no. 3, pp. 197–243, 1995.
- [54] A. M. Carvalho, “Scoring functions for learning Bayesian networks,” 2009.
- [55] G. F. Cooper and E. Herskovits, “A Bayesian Method for the Induction of Probabilistic Networks from Data,” 1991.

- [56] J. Cheng and R. Greiner, "Comparing Bayesian Network Classifiers," *Proc. Fifteenth Conf. Uncertain. Artif. Intell.*, pp. 101–108, 1999.
- [57] P. Spirtes and C. Glymour, "An Algorithm for Fast Recovery of Sparse Causal Graphs," *Soc. Sci. Comput. Rev.*, vol. 9, no. 1, pp. 62–72, 1991.
- [58] A. Murray-Watters, "The DM Algorithm : A Causal Search Algorithm for the Discovery of MIMIC Models , with an Attempt to Recover a Protein Signalling Network from a High-Dimensional Ovarian Cancer Dataset," Carnegie Mellon University, 2014.
- [59] I. Tsamardinos, C. F. Aliferis, and A. Statnikov, "Time and sample efficient discovery of Markov blankets and direct causal relations," in *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '03*, 2003, p. 673.
- [60] N. Friedman, "Inferring cellular networks using probabilistic graphical models," *Sci. Signal.*, vol. 303, no. 5659, p. 799, 2004.
- [61] M. P. Ward and F. I. Lewis, "Bayesian Graphical modelling: Applications in veterinary epidemiology," *Prev. Vet. Med.*, vol. 110, no. 1, pp. 1–3, 2013.
- [62] F. Liu, F. Tian, and Q. Zhu, "Bayesian Network Structure Ensemble Learning," *ADMA*, pp. 454–465, 2007.
- [63] A. H. Marshall, S. I. McClean, C. M. Shapcott, I. R. Hastie, and P. H. Millard, "Developing a Bayesian belief network for the management of geriatric hospital care," *Health Care Manag. Sci.*, pp. 25–30, 2001.
- [64] K. P. Murphy, "The Bayes Net Toolbox for MATLAB," *Comput. Sci. Stat.*, vol. 33, p. 2001, 2001.
- [65] A. L. Madsen and F. V. Jensen, "Lazy propagation: a junction tree inference algorithm based on lazy evaluation," *Artif. Intell.*, vol. 113, no. 1, pp. 203–245, 1999.
- [66] M. Bal, M. F. Amasyali, H. Sever, G. Kose, and A. Demirhan, "Performance evaluation of the machine learning algorithms used in inference mechanism of a medical decision support system," *ScientificWorldJournal.*, vol. 2014, no.

September, p. 137896, 2014.

- [67] "Data Access." [Online]. Available: <https://www.ncdc.noaa.gov/data-access>.
- [68] C. Rudloff and B. Lackner, "Modeling Demand for Bikesharing Systems," *Transp. Res. Rec. J. Transp. Res. Board*, vol. 2430, no. January, pp. 1–11, 2014.
- [69] E. O. Mahony and D. B. Shmoys, "Data Analysis and Optimization for (Citi) Bike Sharing," *Proc. Twenty-Ninth AAAI Conf. Artif. Intell. Data*, pp. 687–694, 2015.
- [70] J. Zhang, X. Pan, M. Li, and P. S. Yu, "Bicycle-sharing system analysis and trip prediction," *Proc. - IEEE Int. Conf. Mob. Data Manag.*, vol. 2016–July, pp. 174–179, 2016.
- [71] G. M. Dias, B. Bellalta, and S. Oechsner, "Predicting occupancy trends in Barcelona's bicycle service stations using open data," *IntelliSys 2015 - Proc. 2015 SAI Intell. Syst. Conf.*, pp. 439–445, 2015.
- [72] J. Froehlich, J. Neumann, and N. Oliver, "Sensing and predicting the pulse of the city through shared bicycling," *IJCAI Int. Jt. Conf. Artif. Intell.*, no. 3, pp. 1420–1426, 2009.
- [73] C. Ome and O. Latifa, "Model-Based Count Series Clustering for Bike Sharing System Usage Mining : A Case Study with the Vélib System of Paris," *ACM Trans. Intell. Syst. Technol.*, vol. 5, no. 3, pp. 1–21, 2014.
- [74] R. Rixey, "Station-Level Forecasting of Bikesharing Ridership," *Transp. Res. Rec. J. Transp. Res. Board*, vol. 2387, pp. 46–55, 2013.
- [75] C. F. Aliferis, I. Tsamardinos, A. R. Statnikov, and L. E. Brown, "Causal Explorer: A Causal Probabilistic Network Learning Toolkit for Biomedical Discovery," *Metmbs*, vol. 3, pp. 371–376, 2003.
- [76] B. Y. Yu, T. Honda, M. Sharqawy, and M. Yang, "Human behavior and domain knowledge in parameter design of complex systems," *Des. Stud.*, vol. 45, pp. 242–267, 2016.
- [77] M. De Jongh and M. Druzzzel, "A comparison of structural distance measures for

- causal Bayesian network models,” ... *Probl. Sci. Comput. Sci. Ser.*, pp. 443–456, 2009.
- [78] J. Peters and P. Bühlmann, “Structural Intervention Distance for Evaluating Causal Graphs,” *Neural Comput.*, vol. 27, no. 3, pp. 771–799, 2015.
- [79] R. Coe, “It’s the Effect Size, Stupid What effect size is and why it is important,” in *Annual Conference of the British Educational Research Association*, 2002, pp. 12–14.
- [80] M. M. Mukaka, “Statistics corner: A guide to appropriate use of correlation coefficient in medical research,” *Malawi Med. J.*, vol. 24, no. 3, pp. 69–71, 2012.
- [81] U. M. Fayyad and K. B. Irani, “Multi-Interval Discretization of Continuous-Valued Attributes for Classification Learning,” in *Proceedings of the International Joint Conference on Uncertainty in AI*, 1993, pp. 1022–1027.
- [82] N. Friedman and M. Goldszmidt, “Discretizing continuous attributes while learning Bayesian networks,” *Proc. 13th Int. Conf. Mach. Learn.*, pp. 157–165, 1996.
- [83] M. Lichman, “UCI Machine Learning Repository,” *University of California, School of Information and Computer Science*, 2013. [Online]. Available: <http://archive.ics.uci.edu/ml>. [Accessed: 11-Dec-2017].
- [84] T. Silander, T. Roos, P. Kontkanen, and P. Myllymäki, “Factorized Normalized Maximum Likelihood Criterion for Learning Bayesian Network Structures,” *Proc. Fourth Eur. Work. Probabilistic Graph. Model.*, pp. 257–264, 2008.
- [85] Z. Liu, B. Malone, and C. Yuan, “Empirical evaluation of scoring functions for Bayesian network model selection,” *BMC Bioinformatics*, vol. 13, no. Suppl 15, p. S14, 2012.
- [86] J. Binder, D. Koller, S. Russell, and K. Kanazawa, *Adaptive Probabilistic Networks with Hidden Variables*, vol. 29, no. 2. 1997.
- [87] P. Spirtes, T. Richardson, and C. Meek, “Heuristic greedy search algorithms for latent variable models,” *Proc. AI STAT’97*, no. 1934, 1997.

- [88] G. Elidan, N. Lotner, N. Friedman, and D. Koller, "Discovering hidden variables: A structure-based approach," *Adv. Neural Inf. Process. Syst.*, pp. 479–485, 2001.
- [89] P. Mylly Aki, T. Silander, H. Tirri, and P. Uronen, "B-Course: a Web-Based Tool for Bayesian and Causal Data Analysis," *Int. J. Artif. Intell. Tools*, vol. 11, no. 3, pp. 369–387, 2002.
- [90] L. Uusitalo, "Advantages and challenges of Bayesian networks in environmental modelling," *Ecol. Modell.*, vol. 203, no. 3–4, pp. 312–318, 2007.
- [91] J. Pitchforth and K. Mengersen, "A proposed validation framework for expert elicited Bayesian Networks," *Expert Syst. Appl.*, vol. 40, no. 1, pp. 162–167, 2013.
- [92] J. Dougherty, R. Kohavi, and M. Sahami, "Supervised and Unsupervised Discretization of Continuous Features," *Mach. Learn. Proc. 1995*, pp. 194–202, 1995.
- [93] "ISO 15031, Road vehicles -- Communication between vehicle and external equipment for emissions-related diagnostics -- Part 5: Emissions-related diagnostic services," 2015.
- [94] "Transmission Gear Ratio Equation and Calculator." [Online]. Available: https://www.engineersedge.com/mechanics_machines/transmission-gear-ratio-calculator.htm. [Accessed: 01-Jan-2017].