PLEASE TYPE

THE UNIVERSITY OF NEW SOUTH WALES
Thesis/Dissertation Sheet

Surname or Family name: Wang

First name: Bing Other name/s:

Abbreviation for degree as given in the University calendar: PhD

School: School of Engineering and Information Technology Faculty: UNSW Canberra

Title: Autonomous hypothesis generation for knowledge discovery in continuous domains

Abstract 350 words maximum: (PLEASE TYPE)

Advances of computational power, data collection and storage techniques are making an overwhelming amount of new data available every day. This situation has given rise to the hypothesis generation research. Hypothesis generation research adopts techniques from machine learning and data mining to autonomously uncover causes in the form of previously unknown hidden patterns and models from data. Those patterns and models can come in different forms (e.g. rules, classifiers, clusters, causal relations). In some situations, data are collected without a priori supposition or imposition of a specific research goal or hypothesis. For example, in sensor networks, sensors record massive amounts of data. In these data, not all forms of relationships can be described in advance. Moreover, the environment may change without a priori knowledge. In a situation like this one, hypothesis generation techniques can potentially provide a paradigm to gain new insights about the data and the underlying system. This thesis proposes a general hypothesis generation framework, whereby assumptions about the observational data and the system are not predefined. The problem is decomposed into two interrelated sub-problems: (1) the associative hypothesis generation problem and (2) the causal hypothesis generation problem. The former defines a task of finding evidence of the potential causal relations in data. The latter defines a refined task of identifying casual relations. A novel association rule algorithm for continuous domains, called functional association rule mining, is proposed to address the first problem. An experimental causal search algorithm is then designed for the second problem. It systematically tests the potential causal relations by querying the system to generate specific data; thus allowing for causality to be asserted. Empirical experiments show that the functional association rule mining algorithm can uncover associative relations from data. If the underlying relationships in the data overlap, the algorithm sometimes decomposes these relationships into their constituent non-overlapping parts. Experiments with the causal search algorithm show a relative low error rate on the retrieved hidden causal structures. In summary, the contributions of this thesis are: (1) a general framework for hypothesis generation in continuous domains, (2) a new functional association rule mining algorithm, and (3) a new experimental causal search algorithm.

Declaration relating to disposition of project thesis/dissertation

I hereby grant to the University of New South Wales or its agents the right to archive and to make available my thesis or dissertation in whole or in part in the University libraries in all forms of media, now or here after known, subject to the provisions of the Copyright Act 1968. I retain all property rights, such as patent rights. I also retain the right to use in future works (such as articles or books) all or part of this thesis or dissertation.

I also authorise University Microfilms to use the 350 word abstract of my thesis in Dissertation Abstracts International (this is applicable to doctoral theses only).

..... Signature ..... Witness 26 / 11 / 2014 Date

The University recognises that there may be exceptional circumstances requiring restrictions on copying or conditions on use. Requests for restriction for a period of up to 2 years must be made in writing. Requests for a longer period of restriction may be considered in exceptional circumstances and require the approval of the Dean of Graduate Research.

FOR OFFICE USE ONLY Date of completion of requirements for Award:

THIS SHEET IS TO BE GLUED TO THE INSIDE FRONT COVER OF THE THESIS

ORIGINALITY STATEMENT

'I hereby declare that this submission is my own work and to the best of my knowledge it contains no materials previously published or written by another person, or substantial proportions of material which have been accepted for the award of any other degree or diploma at UNSW or any other educational institution, except where due acknowledgement is made in the thesis. Any contribution made to the research by others, with whom I have worked at UNSW or elsewhere, is explicitly acknowledged in the thesis. I also declare that the intellectual content of this thesis is the product of my own work, except to the extent that assistance from others in the project's design and conception or in style, presentation and linguistic expression is acknowledged.'

Signed .....

Date 26/11/2014

Autonomous Hypothesis Generation for Knowledge Discovery in Continuous Domains

Bing Wang

M.Eng. (Control Theory and Control Engineering) Ocean University of China, China

B.Eng. (Automation) Ocean University of China, China

A thesis submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy at the
School of Engineering and Information Technology
University of New South Wales
Australian Defence Force Academy

Abstract

Advances of computational power, data collection and storage techniques are making new data available every day. This situation has given rise to hypothesis generation research, which complements conventional hypothesis testing research. Hypothesis generation research adopts techniques from machine learning and data mining to autonomously uncover causal relations among variables in the form of previously unknown hidden patterns and models from data. Those patterns and models can come in different forms (e.g. rules, classifiers, clusters, causal relations).

In some situations, data are collected without *a priori* supposition or imposition of a specific research goal or hypothesis. Sometimes domain knowledge for this type of problem is also limited. For example, in sensor networks, sensors constantly record data. In these data, not all forms of relationships can be described in advance. Moreover, the environment may change without *a priori* knowledge. In a situation like this one, hypothesis generation techniques can potentially provide a paradigm to gain new insights about the data and the underlying system.

This thesis proposes a general hypothesis generation framework, whereby assumptions about the observational data and the system are not predefined. The problem is decomposed into two interrelated sub-problems: (1) the associative hypothesis generation problem and (2) the causal hypothesis generation problem. The former defines a task of finding evidence of the potential causal relations in data. The latter defines a refined task of identifying casual relations.

A novel association rule algorithm for continuous domains, called functional association rule mining, is proposed to address the first problem. An agent based causal search algorithm is then designed for the second problem. It systematically tests the potential causal relations by querying the system to generate specific data; thus allowing for causality to be asserted.

Empirical experiments show that the functional association rule mining algorithm can uncover associative relations from data. If the underlying relationships in the data overlap, the algorithm decomposes these relationships into their constituent non-overlapping parts. Experiments with the causal search algorithm show a relative low error rate on the retrieved hidden causal structures.

In summary, the contributions of this thesis are: (1) a general framework for

hypothesis generation in continuous domains, which relaxes a number of conditions assumed in existing automatic causal modelling algorithms and defines a more general hypothesis generation problem; (2) a new functional association rule mining algorithm, which serves as a probing step to identify associative relations in a given dataset and provides a novel functional association rule definition and algorithms to the literature of association rule mining; (3) a new causal search algorithm, which identifies the hidden causal relations of an unknown system on the basis of functional association rule mining and relaxes a number of assumptions commonly used in automatic causal modelling.

keywords

Knowledge discovery, hypothesis generation, data mining, causal modelling, association rule mining, evolutionary computation, heuristic search, artificial neural networks, agent systems

Acknowledgement

I am deeply grateful to my supervisors Prof. Hussein A. Abbass and Dr. Kathryn E. Merrick, whose advice, support, encouragement and patience have been a great source of help in every aspect of my work.

The next big thank you goes to my fellow lab mates. Their timely advice, support and influence brought a substantial contribution to my learning. I acknowledge Dr. Jiangjun Tang, Dr. George Leu, Dr. Ayman Ghoneim, Mr. Bin Zhang, Mr. Murad Hossain, and Ms. Shen Ren. I am also grateful to my dear colleagues and friends: Dr. Li Li, Dr. Yanyan Liu, Dr. Hock Chuan Lim, Mr. Nizami Jafarov, Mr. Blaise Tardy, Mr. Sascha Fink, Dr. Haobo Zhang, Dr. Jane Koerner, and Dr. Sarah Rittner. I consider myself fortunate to have their friendship, constant support and encouragement during this journey.

I wish to extend my gratitude to the academics and staff of the University of New South Wales – Canberra for their help along the way. My acknowledgement also goes to the anonymous reviewers for their constructive comments on the papers related to this thesis.

On a personal level, I sincerely thank my parents, Shufang Sui and Hao Wang, for the love, encouragement and strength they have been giving me throughout the candidature.

The financial support of the China Scholarship Council and the University of New South Wales – Canberra Campus are gratefully acknowledged. I also would like to acknowledge the use of NCI intersect super computing facilities.

Bing Wang

Canberra 2014

Certificate of Originality

I hereby declare that this submission is my own work and that, to the best of my knowledge and belief, it contains no material previously published or written by another person, nor material which to a substantial extent has been accepted for the award of any other degree or diploma at UNSW or any other educational institution, except where due acknowledgement is made in the thesis. Any contribution made to the research by colleagues, with whom I have worked at UNSW or elsewhere, during my candidature, is fully acknowledged.

I also declare that the intellectual content of this thesis is the product of my own work, except to the extent that assistance from others in the project's design and conception or in style, presentation and linguistic expression is acknowledged.

Bing Wang

Contents

List of Figures

List of Tables

List of Acronyms

AHGP	Associative hypothesis generation problem
ANN	Artificial neural network
ARM	Association rule mining
BP	Backpropagation
CCEA	Cooperative co-evolutionary algorithm
CCFARM	Cooperative co-evolution functional association rule mining
CHGP	Causal hypothesis generation problem
DE	Differential evolution
DE-LFARM	Differential evolution based linear functional association rule mining
EA	Evolutionary algorithm
ES	Evolution strategy
FAR	Functional association rule
FARM	Functional association rule mining
FFNN	Feedforward Neural Network
GA	Genetic algorithm
GA-LFARM	Genetic algorithm based linear functional association rule mining
GAR	Genetic association rule mining algorithm
LFAR	Linear functional association rule
LFARM	Linear functional association rule mining
MODENAR	Multi-objective differential evolutionary algorithm for mining numeric association rules
PBIL	Population based incremental learning
PBIL-LFARM	Population based incremental learning for linear functional association rule mining
PC	Principle component
PCA	Principle component analysis

List of Publications

Bing Wang, Kathryn E. Merrick, Hussein A. Abbass. Developing Attention Focus Metrics for Autonomous Hypothesis Generation in Data Mining. In proceedings of 9th international conference on simulated evolution and learning (SEAL2012). LNCS Springer, 2012.

Chapter 1

Introduction

1.1 Thesis Overview

Traditional scientific research (or knowledge discovery) starts with a hypothesis suggesting an interpretation or description of a phenomenon. This hypothesis becomes the foundation for all further inferences and experiments. Its construction is heavily dependent on the researcher's vision and skills, such as observation, domain knowledge, reasoning, imagination and creativity. Once constructed, a hypothesis leads to the design of the experiments and data collection required to test it. Therefore, this type of research is often called hypothesis-testing research (or hypothesis-driven research).

The development of penicillin (in 1877 by Louis Pasteur) is a classic example of constructing a hypothesis based on a researcher's competencies [141]. Initially, Pasteur simply observed how anthrax bacilli stopped growing in the presence of mould. Then using his domain knowledge and creativity, he formed the initial interpretation of the phenomenon, i.e., the hypothesis. As a result, experiments were designed. Eventually, penicillin was developed.

However, in hypothesis-testing research, constructing a hypothesis is a slow and incremental process, which requires the accumulation of domain knowledge and the interest of researchers in progress towards new knowledge. Advances in

computational power, as well as data collection and storage techniques have provided researchers an overwhelming amount of data. Although critically important, the rates at which hypotheses are manually constructed and experiments designed to extract knowledge from data, have fallen behind the speed of data accumulation. This situation has given rise to another type of knowledge discovery: hypothesis generation.

Hypothesis generation is related to finding patterns from data through data mining or machine learning tools [12, 184, 99, 50, 98, 141]. In situations in which a prohibitive amount of data is collected, domain knowledge may not be sufficient for the construction of a complete hypothesis [98]. The automatic generation of hypotheses based on pattern searching and identification provides an alternative approach to, and accelerates the process of, knowledge discovery.

In hypothesis generation research, the definition of a 'hypothesis' can take different forms; for example, as a rule, cluster, classifier, graph or equation [10]. In this thesis, hypothesis refers to classical causal relations in a system. Causal relations can provide insights about a system, as characterised by the variables measuring it and can also be used to apply effective control to a system. In the literature, the hypothesis generation approach for a causal structure falls in the field of automatic causal modelling [181].

The problem that automatic causal modelling aims to solve is reconstructing the causal structure of a system from only observational data. This is made possible by the connection between a causal structure and conditional independence/dependence. A true causal structure which satisfies the Causal Markov Condition (CMC) encodes conditional dependencies among its variables [147]. That is, if its variables are connected by causal relations, they show certain dependency patterns. Conversely, if the dependency patterns of the variables are due to only the underlying causal relations, they can be tested to retrieve these relations from data, as demonstrated in automatic causal modelling algorithms [181].

However, retrieving a causal structure in this way makes a number of assump-

tions about the data and underlying causal structure. Firstly, causal relations are assumed to exist in observational data, with the underlying causal relations satisfying the CMC, while the dependency patterns of observational data are caused only by the underlying causal relations (i.e. the faithfulness assumption [147]). No latent variables exist in a system. When the observational data are comprised of continuous data, additional assumptions are included. The interrelations of continuous variables are assumed to be linear and the density distributions of their associated values assumed to be identical.

Instead of developing hypothesis generation approaches for a system confined by a number of assumptions, this thesis considers a more general scenario. In it, the interrelations of variables measured from a system are not assumed to be subject to only causal relations. In addition, no specific *a priori* knowledge is assumed to be available about the underlying structure of the system. The concern here is that there are situations, in which the observational data collected are not particularly for the purpose of retrieving causality. For example, in the big data problem [116], data are recorded on daily bases. No specific research purpose was set when the data were recorded. People are interested in analysing the data. However, what questions can be answered remains unknown. Hypothesis generation helps to form initial questions about how to use the accumulated data.

Another example is in the field of intelligent systems in which sensors record activity data from the environment and can perform pre-defined tasks to adapt to human activities through machine learning techniques [122]. Such tasks can be developed manually when we know the common activities in typical scenarios, e.g., offices, lecture theatres. However, we need new ways for an intelligent environment agent to hypothesise about how to adapt to non-standard scenarios for which knowledge about what the observational data are describing is not available.

A similar situation occurs in cyber security in which, due to the constant evolution of hacking activities, previous knowledge about abnormal activities in log data can get outdated. How to use log data to actively acquire updated insights into a system is an interesting research direction for which an agent that can generate new

hypotheses from data could be an advantage [174].

Generalising from the above scenarios, this thesis first defines the problem of hypothesis generation for continuous domains whereby a system is measured by a set of continuous variables for which no specific *a priori* knowledge regarding their underlying relations is assumed. Then, the proposed hypothesis generation problem takes the observational data of the system as input and considers the system's underlying casual structure as its final output. In a situation in which minimum domain knowledge is available, identifying causal relations could provide a maximum information about the target system.

The strategy in this thesis for dealing with such situations is to focus on developing hypothesis generation approaches in particular ones that can identify causal relations. The inspiration for this is that, without specific domain knowledge, causal relations give us direct insights into a system. This information is particularly useful as it provides solid *a priori* knowledge about how to understand and control the system. However, it is possible that the variables measured from a system do not possess causal relations. If so, the associative relations among them can be used as evidence for conducting causal hypothesis generation.

From the above, this generalised problem is further decomposed into two subproblems: an associative hypothesis generation problem (AHGP) and a causal hypothesis generation problem (CHGP). The AHGP defines a task of identifying the associative relations in given observational data, while the CHGP defines a task of identifying causal relations. This decomposition is as shown in Figure 1.1.

Given the proposed problem definitions, another main focus of this thesis is to develop approaches for solving them. These are designed separately. For the AHGP, firstly, a novel associative relation representation, termed the functional association rule (FAR) is proposed. The approach for searching for associative relations in the FAR form is then designed. To solve CHGP, an experimental causal search approach is introduced.

This thesis draws on association rule mining (ARM) to build a solution to the

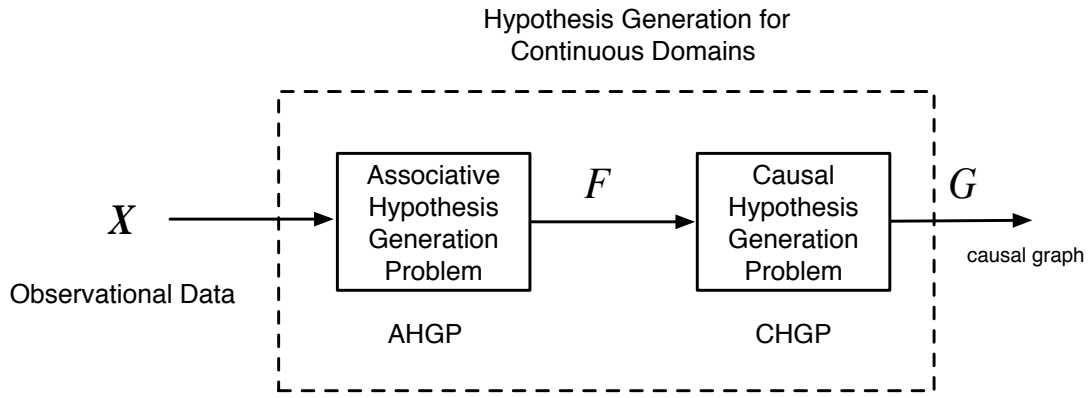


Figure 1.1: General view of hypothesis generation problem and sub-problems. **X**-observational data, **F** - a set of associative hypotheses. G - final causal hypotheses.

AHGP. However, in the literature, conventional ARM for continuous variables has a disadvantage when building up associations among these variables. Conventional ARM mainly converts variable values into intervals. By applying such discretisation, it can find associations among discretised sets rather than continuous variables. Therefore, an alternative rule form, a FAR, which directly represents the associative relations among variables, is proposed. A FAR groups and separates the related variables. Identifying the FARs in the observational data establishes the evidence and inputs for the CHGP.

Two specific types of associative relations are investigated as FARs: linear associative relations and general associative relations. In this thesis, a linear regression model is adapted for the former and artificial neural networks (ANNs) for the latter. As the downward closure property often adopted in ARM does not apply in the FAR definition [6], in this thesis, functional association rule mining (FARM) is cast as a heuristic search process. Its search space is infinitely large since the number of possible combinations is large enough to be considered unbounded. It is also deceptive since similar combinations can perform quite differently. It is multimodal since we are expecting there to be multiple valid rules existing in the data. For dealing with such a difficult, rugged, multimodal search space, evolutionary algorithms are often more efficient than other techniques [208]. An evolutionary algorithm (EA)

works with a population of candidate solutions that permits concurrent exploration of different parts of the search space, with a crossover operator working on multiple different genes at the same time. It has the potential to preserve interrelated variables in its building blocks. EAs as heuristic search methods are known for their robustness and low sensitivity to noise, therefore they can be applied for designing the search approach. Three EA-based search methods are proposed for linear functional association rule mining (LFARM) and cooperative co-evolutionary algorithm based approach for the general functional association rule mining (FARM).

In this thesis, the FAR and its mining approaches serve as a solution to the AHGP. However, independent of the context of hypothesis generation, they also contribute to continuous variable ARM which provides a novel rule form to the definitions of association rules as well as the corresponding mining approaches. This suggests a new perspective on what associative knowledge can be mined from datasets besides discretising their variables. Many if not most real world datasets comprise continuous variables.

The existence of FARs provides evidence of potential causal relations with the results from FARM comprising the input for the CHGP. FARs also reduce potential causal relations to subsets of the original measured variables. As per the abovementioned general problem definition, *a priori* knowledge about the underlying structure of the system is not required. By relaxing the assumptions about underlying relations, our proposed solutions for causal hypothesis generation is based on the potential outcome of interventions. This has become popular in hypothesis testing research [15, 77, 153, 202]. This often relies on manually designing experiments on the variables of interest selected by domain knowledge. Since FARs group these variables, an intervention can be systematically applied on the associated variables. Then, an experimental casual search algorithm is designed to apply interventions on the potential causal relations specified by the FARs.

Empirical experiments are conducted to study the performance of the designed solutions, with comparisons of the proposed LFARM algorithms using complexity and perceptual selectivity metrics suggesting that there is no significant differences

among three EA-based LFARM approaches in these respects. Experiments on the general FARs indicate the similarity between those mined and the hidden associative relations in the observational data. The main factor affecting performance identified as the overlapping of the underlying relations. Experiments on the CHGP show that it has a low error rate in identifying the underlying causal relations. The error rate is influenced by the quality of the mined FARs. The remainder of this chapter restates this thesis research objectives and questions, summarises contributions and significance, and concludes with an overview of its organisation.

1.2 Research Objectives

The primary focus of this thesis is to formulate a generalised hypothesis generation problem and develop systematic approaches for solving it. This research is focused on the following specific objectives:

- The first objective is to define a general hypothesis generation problem for autonomously learning about unknown systems measured by a set of continuous variables. We also aim to identify sub-problem components that may assist with the design of solutions.

- The second objective is to design algorithms as solutions for the defined autonomous hypothesis generation problem, and its subproblem components. This step is focused on how the algorithm(s) can probe the system for potential useful knowledge. This objective will be measured by the features of the probing results, for example, the number of rules found, their complexity, their similarity to the known rules in the underlying system and comparison with other state-of-the-art algorithms in relevant fields.

- The third objective is to design algorithm(s) that can finalise the hypothesis generation process based on the probing process. This objective will be measured by error rate of the final hypotheses generated by the algorithm(s) compared with knowledge of interest in pre-designed systems.

The scope of the proposed problem is an unknown system that is measured by a set of continuous variables. It has assumed that there is limited specific domain knowledge about its underlying relations known in advance.

1.3 Contributions and Significances

In addressing the above research objectives, this thesis makes the following contributions:

- A new problem definition for hypothesis generation in continuous domains is proposed. This problem definition looks into situations where data are collected from unknown system, but domain experts have no sufficient knowledge to systematically construct hypothese about the underlying system, or manual hypothesis testing does not catch up with the speed of data accumulation. The novelty of our problem definition lies in the fact that it does not assume priori knowledge about the hidden structure in data. The proposed problem definition does not specially require that the unkown system is only constructed by causal relations. The underlying system can even include several independent systems. However, in literature, automatic causality investigation often poses a number of restrictions to the system, e.g. Causal Markovian Condition, faithfulness[147].

  Our hypothesis generation problem is further decomposed into two sub-problems, the AHGP and CHGP, which are presented in Chapter 3. Decomposing the problem into a hierarchical structure introduces two benefits. On one hand, decomposition enables a probing step, which makes the relaxation of underlying system possible. On the other hand, it introduces a useful pattern that also plays an important role in knowledge discovery, association pattern. By introducing decomposition, association is not just one-off investigation in data mining, but also preparation for investigating causal relations. The common critics on mining association that association relations may lead to misleading

interpretation (as association does not always imply causation) can be alleviated, as we integrate the study of association into causal hypothesis generation. Besides, due to this probing step of associative relations, when we proceed to investigate causal relations, constraints on the unknown system can be relaxed to a certain extend. This decomposition creates a mutual support and integration between associative relation study and causal relation study.

- Two FAR representations, LFAR and FAR, based on the linear regression model and ANNs to narrow down the search space for possible causal relations are introduced. They are used to probe associative relations among a set of variables. Novel algorithms for mining associative relations represented by LFAR and FAR are proposed.

  Search for associative relations in continuous domain falls into the field of association rule mining. However, the conventional methods convert variables into intervals, rather than catching variable level associations. Interval based association rule often makes it relatively difficult to progress to causal relation investigation, as the relations are among subsets of the variable values, rather than general relations among variables. The LFAR and FAR definition extends the association rule mining to reflecting variable level association. Such an extension is important as it forms the foundation for investigating whether there are causal relations hidden in the unknown system.

  In Chapter 3, we present the detailed definition of the FAR and LFAR, and aslo discuss the characteristics of the FAR and LFAR generation from a search perspective. Due to the new definition of FAR and LFAR, conventional ARM framework is no more suitable for generating FAR and LFAR from a given dataset. Therefore, we propose novel algorithms for mining such associative relations. In Chapter 4, LFARM approaches are designed according to the representation and search strategies discussed in Chapter 3, with three evolutionary algorithm-based mining approaches designed for the LFARM problem. Two metrics for evaluating the quality of the generated FARs by comparing the behaviours of different algorithms are proposed. Empirical studies show

that, in general, three algorithms have significantly different performances in the number of LFARs found and the computational time used. However, for the complexity and perceptual selectivity, there is no evidence that shows differences among the three approaches.

Linear association rules have the advantage of being simple and easily interpreted. However, such simplicity is also a barrier to them capturing complex relations among variables as it is common for the underlying relations of variables to form complex non-linear relations. Chapter 5 presents the general FAR form and its mining approach (FARM) for improving the representational capability of the linear FAR. Details of the implementation of the cooperative co-evolutionary algorithm for FARM are provided, with empirical experiments showing that the general FARM can uncover the associative relations. The factor that influences the performance of the algorithms is identified as the overlapping of the underlying relations. This algorithm is also compared with two state-of-the-art evolutionary algorithm based association rule mining approach [119, 8]. Although, our problem definition is different from interval based association rule, since they both deal with continuous variables, we conduct comparison experiments, and FARM has shown competitive performance.

- An experimental causal search algorithm for causal hypothesis generation based on FARs is proposed. FARs place the given variables into groups, where those within one group are interdependent, and causal hypotheses are built on these confined variable sets. We present this algorithm as an agent architecture, which systematically applies interventions on the interrelated variables and, according to their consequences, can establish the causal relations. Conventional automatic causal modelling requires that the interrelations reflected among variables are only caused by causal relations [147, 181]. This assumptions imply that a certain understanding/domain knowledge about the target system is necessary for modelling causal relations. With rapid development of monitoring techniques and accumulation of data, domain knowledge is often unavailable or not sufficient to confirm the characteristics of the underlying

structure. Our approach relaxes this condition, it does not require specific assumptions between the underlying structure and observational data. This characteristic allows it to be used for exploring new environment with limited prior knowledge. These content is presented in Chapter 6. Two sets of experiments (50 and 25 datasets respectively) on synthetic datasets with causal relations of different complexities are conducted and used to investigate the algorithm's performance in retrieving the underlying causal relations, with the factors that affect its performance discussed. In addition, a play-board environment is designed to introduce context for the causal investigation, with the experiments on its underlying causal relations revealing two other factors that influence the experimental causal search algorithm.

Chapter 2

Literature Review

2.1 Introduction

Hypothesis generation serves a complementary role to hypothesis testing in knowledge discovery. It uses data mining and machine learning techniques to automatically find patterns and models in data. It accelerates the knowledge discovery process in data–rich fields where the domain knowledge for manually designing individual hypotheses is often limited [98]. However, in general, it still relies on a certain amount of domain knowledge to interpret semantics, formulate interesting knowledge and define the regularities for generating corresponding hypotheses. For example, one hypothesis generation study of genomics conducted by King et al. [99] first encodes the background knowledge of biochemical equations of aromatic amino acid synthesis pathways. Its goal is to generate connections between genetics and biochemistry, with a logical model of yeast metabolism attached as a priori knowledge and the hypotheses automatically generated by abducting the different probabilities from the model.

For the common hypothesis generation research studies in the literature, data are often collected assuming a certain purpose and their relevance to certain confined fields. However, although these studies have general research goals and certain domain knowledge of the fields, hypothesis testing research cannot catch up with

the speed of data accumulation [98]. In contrast, data can be logged on a very general basis without a specific purpose being initially set or a clear idea of how to use the data collected. When domain experts analyse such datasets, they are often biased by their existing knowledge and focus on the patterns that they are familiar with. Faced with such a situation, this thesis raises the question of whether the hypothesis generation paradigm can provide methods for establishing initial insights into, and understanding of, the system behind the data, thus avoiding human bias. As the scenarios considered place more constraints on the available domain knowledge, a hypothesis can be defined by the causal relations in the system behind the data. The advantage of such a definition of hypothesis is that, it not only provides a compressed representation of the data but also implies potential control strategies. However, without particular domain knowledge, it is also not known in advance whether any causal relation exists in the data. As potential causal relations join related variables, associative relations are often used as evidence and to refine the related variables for further hypothesis generation for causal relations.

The generalised problem on which this thesis focuses involves synthesising research from a number of different fields, including ARM, heuristic searching and causal models, which seek to answer similar questions. This chapter begins with Section 2.2 by reviewing the research field of automatic identification of associative relations, that is, ARM. As well as the general framework, its applications to continuous domains are also reviewed to determine how they can be adopted to the problem with which this thesis is concerned. Without specific domain knowledge of the underlying system for guiding hypothesis generation, a generation strategy often relies on heuristics. Consequently, evolutionary computation as a heuristic search approach is discussed in Section 2.3. The theory and notions of artificial neural networks (ANNs), and their connections to the representation of the associative relations of the proposed problem definition are reviewed in Section 2.4. The above three sections provide background knowledge for the work presented in Chapter 3 and Chapter 4. The principles and algorithms that focus on causal inference are presented in Section 2.5 and their relevance to the problem with which this thesis

is concerned are discussed. The types of hypothesis definitions commonly used in hypothesis generation research are reviewed in Section 2.6 to assist in further distinguishing the interest of this thesis from those of other hypothesis generation studies. Finally, Section 2.7 provides a summary of this chapter.

2.2 Hypothesis Generation for Associative Relations: Association Rule Mining

Strategies for generating hypotheses have included observing associations in research settings; for example, a study of the correlation between marital happiness and a certain gene (5-HTTLPR) [29]. On the one hand, although critically important, they are based on domain knowledge and interest of the individual researcher. On the other hand, some major discoveries have been the product of serendipity from observations beyond domain knowledge; for example, Louis Pasteur observed that the growth of the anthrax bacilli in a culture was inhibited when the bacilli were contaminated with moulds (1877), an associative relation that led to an important discovery in modern pharmacology. Focusing attention on such an association, requires the researcher to not be biased by his/her domain knowledge and be open to factors not covered by his/her background knowledge [141]. The substantial increases in data accumulation speeds have stimulated the rapid development of data mining. This offers the potential to break down barriers to the rapid growth of knowledge bases by making use of unbiased observations to find new leads for follow-up studies. The specific sub-field that deals with associative relations mining in data mining is association rule mining (ARM), the concept and techniques of which are discussed in the following sections.

ARM is a family of techniques that searches for associations in datasets. The original problem it addressed was finding a correlation among sales of different products (the shopping basket problem) [6]. It then became a focused theme in data mining research due to its simplicity, interpretability and adaptability to a broad

range of problems. Many efforts have been dedicated to this research and tremendous progress has been made. The following sections review ARM from three main perspectives: basic ARM framework, ARM approaches, and ARM for extended patterns.

2.2.1 Association Rule and Association Rule Mining Framework

An association rule represents a relationship between two sets of variables, specifying that their co-occurrence in a dataset exceeds some threshold, and its formal definition is as follows: Let $I = \{I_1, I_2, ..., I_m\}$ be a collection of items. A sub-set of I ($A \subseteq I$) is called an itemset and, if $k = |A|$, we call A a $k-$itemset. Let D be a dataset, each instance of which is a sub-set of I and is called a transaction (T) associated with an identifier (TID). Then, if $A \subseteq T$ holds, an A itemset is supported by a T. An association rule employs the expression $A \Rightarrow B$, where $A \subset I$, $B \subset I$ and $A \cap B = \emptyset$. It is supported by database D, which is the fraction of transactions containing $A \cup B$, with confidence used to describe the percentage of transactions in D, which contain both A and B.

$$support = P(A \cup B)$$

$$confidence = P(B|A)$$

Support and confidence are important measures of interestingness in terms of an association rule as they reflect its validity and certainty, and also provide the criteria for extracting association rules. ARM is a two-step process. Firstly, it identifies frequent itemsets in the data. An itemset (A) is frequent if $support(A) \geq min\_supp$ (a pre-defined minimum support threshold). Once all frequent itemsets and their support values are known, deriving association rules is straightforward. The following rule generation step checks the confidence of all the rules of the forms

$A \setminus B \Rightarrow B$, $B \in A$, $B \neq A \neq \emptyset$ and drops all those that do not exceed the minimum confidence value. It is sufficient to use the support values of the sub-sets of A to determine the confidence because of the Apriori property [7], that is, all non-empty sub-sets of a frequent itemset must also be frequent. Based on this, the mining task can be reduced to the problem of finding all itemsets that are frequent with respect to *min_supp*.

For practical applications, looking at all the sub-sets of I in order to find frequent patterns is not desirable as a linearly increasing number of items implies an exponentially increasing number of itemsets that need to be considered. Due to this exponential growth in complexity for identifying frequent patterns, naive exploration techniques are often intractable. If a boundary that separates the frequent and infrequent itemsets is also independent of any specific data and *min_supp*, the search space can be compressed. Substantial research efforts have been devoted to finding an efficient means of discovering frequent itemsets.

2.2.2 Association Rule Mining Approaches

The fundamental algorithm for ARM is the Apriori algorithm [7], which has had a great impact on a variety of later ARM research and, as the name implies, uses its Apriori property as prior knowledge to reduce exploration.

This algorithm derives candidate frequent itemsets using an iterative search method. For example, suppose that L_k represents a set of valid frequent itemsets of cardinality k (i.e., the support of each itemset in L_k exceeds the value of *min_supp*), with C_k its candidate set (i.e., the superset of L_k members of C_k, which can be either frequent or not frequent). In order to generate L_k, the Apriori algorithm uses the valid frequent sets of cardinality $k-1$, L_{k-1}. Supposing that the valid frequent set is L_{k-1}, the candidate set (C_k) is generated by applying the $\bowtie$ operator on its members. Given two frequent itemsets ($l_i = \{l_i(1), l_i(2), ..., l_i(k-1)\}$ and $l_j = \{l_j(1), l_j(2), ..., l_j(k-1)\}$), they can be merged into a k-itemset only when only their last items are different, as shown in Equation 2.1. The items in each itemset

are sorted in a certain order (e.g., alphabetically), and the dataset is scanned to determine support of the candidates in C_k. Then, the Apriori property is applied to compress the exploration space of C_k. If a sub-set of an itemset candidate in C_k is not in L_{k-1}, that can be deleted from C_k (by applying the Apriori property), a process detailed in Algorithm 1.

$$\begin{aligned} c = \{l_i \cup l_j | \quad & l_i, l_j \in L_{k-1} \wedge \\ & \{l_i(1), l_i(2), ..., l_i(k-2)\} = \{l_j(1), l_j(2), ..., l_j(k-2)\} \wedge \\ & l_i(k-1) \neq l_j(k-1)\} \end{aligned} \tag{2.1}$$

where c refers to a k-itemset.

2.2.2.1 Candidate Generation ARM Algorithms

The Apriori algorithm provides a standard search space reduction technique, mainly through its Apriori property, and also features a family of ARM algorithms, which generate candidates for identifying frequent itemsets. However, candidate generation still suffers from a very large number of candidate sets, which requires repeated scans of the database to check candidates by pattern matching. An extension of the Apriori algorithm [136] is delayed accrual, which is based on the observation that any support of itemset C_i ($|C_i| = \{k+1, k+2, ..., 2k\}$) can be the union of some pair of L_k itemsets. Thus, from a single scan of D, the support of all candidates of lengths k + 1, k + 2, ..., 2k can be computed. However, a trade-off has to be made between the time saved by reducing the number of dataset access and the number of false positives generated through the projection of C_k.

Dynamic itemset counting [32] aims to reduce the number of database scans required to determine the support of frequent patterns. Its main concept is that it allows support counting of larger frequent patterns during early scans of smaller frequent patterns using dataset partitioning and checkpoints. If, during processing, all $(k-1)$-itemsets and larger patterns (e.g., k-itemset) are determined to be frequent

Algorithm 1: Pseudo code of basic Apriori algorithm [6, 71]

Intput : transactional database (D), *min_supp*
Output: all frequent itemsets (L)

```
1  k = 1
2  L_1 = {frequent 1-items}
3  for k=2; L_{k-1} ≠ ∅; k++ do
4  |  C_k = apriori_gen(L_{k-1})
5  |  foreach transaction t in D do
6  |  |  C_t = subset(C_k, t);
7  |  |  foreach c ∈ C_t do
8  |  |  |  c.count ++
9  |  |  end
10 |  end
11 |  L_k = {c ∈ C_k | c.count ≥ min_supp}
12 end
13 Return L = ∪_k L_k
14
15 apriori_gen(L_{k-1})
16 for i = 1; i < sizeof(L_{k-1}); i++ do
17 |  for j = 1; j < sizeof(L_{k-1}); j++ do
18 |  |  if {l_i(1), l_i(2), ..., l_i(k-2)} = {l_j(1), l_j(2), ..., l_j(k-2)} ∧
   |  |  l_i(k-1) ≠ l_j(k-1) then
19 |  |  |  c = l_1 ⋈ l_2
20 |  |  |  if has_infrequent_subset(c, L_{k-1}) then
21 |  |  |  |  delete c
22 |  |  |  else
23 |  |  |  |  add c to C_k
24 |  |  |  end
25 |  |  end
26 |  end
27 end
28 Return C_k
29
30 has_infrequent_subset(c, L_{k-1})
31 foreach (k-1)-subset s of c do
32 |  if s ∉ L_{k-1} then
33 |  |  return TRUE
34 |  else
35 |  |  return FALSE
36 |  end
37 end
```

at a particular checkpoint, the count of occurrence starts until the scan reaches the same checkpoint during the next iteration. As, within a single scan, frequent patterns of multiple lengths can be checked, the overall amount of data access is reduced.

Direct hashing and pruning (DHP) [144], a hash-based technique, deals with efficiency from the angle of reducing the number of candidate frequent itemsets as the more generated itemsets, the more pattern matching is required. When scanning each transaction in the database to count the support of candidate k-itemsets, DHP accumulates information about candidate $(k+1)$-itemsets in advance by hashing the possible $(k+1)$-itemsets of each transaction into different buckets of a hash-table structure and increasing the sizes of their corresponding bucket counts. As a $(k+1)$-itemset with a corresponding bucket count below min_supp cannot be frequent, it is deleted from the candidate set. The algorithm also incorporates progressive dataset pruning to discard items and objects of no further use. As a transaction that does not contain any frequent k-itemsets cannot contain any frequent $(k+1)$-itemsets, it can be marked or removed from further data scans. This study showed significant speeding up of Apriori for short frequent itemset lengths, especially 2. A follow-up study of perfect hashing and pruning [142] used perfect hashing to eliminate the hash-table collision that affects the algorithm's performance in DHP.

The partition algorithm, which adopts a divide-and-conquer approach, is particularly suitable for very large databases and ideal for parallelisation [170]. It discovers all valid itemsets in two dataset scans and consists of two phases. The first divides dataset D into n non-overlapping partitions, in each of which frequent patterns are found and called local frequent itemsets. The data structure is then converted into one that for each itemset the record is the TIDs containing it. Consequently, one scan can find all local frequent items. Although not all local frequent itemsets are always frequent in reference to the entire dataset (D), they are candidate itemsets with respect to D. The collection of these itemsets forms the global candidate itemsets for D and a second scan of the dataset assigns the real supports to each candidate itemset to derive the final output. The efficiency of such an algorithm

is further improved in the studies conducted by Mueller et al. [136] and Lin et al [113]. The SPINC algorithm reduces processing time by dynamically processing global candidate frequent itemsets and starting their counting supports during the first scan, which results in reductions in scanning times. The study carried out by Lin et al. focuses on the partitioning process with the aim of eliminating data skew in the partition results, which may cause the generation of false candidates [113].

While many studies focus on the efficiency of mining algorithms, how the data's representation, organisation and access may affect performance is also a research interest. The data formats used in ARM in its very basic form are horizontal and vertical. That using the identifier TID is horizontal, that is, each row contains items, whereas a vertical data format uses an item to lead a record and each row contains the TIDs with that item. Equivalence CLAss Transformation (ECLAT) [212] is an approach for mining frequent itemsets using the vertical data format, which first intersects the TID set of every pair of frequent single items, with its following steps based on the Apriori property that candidate k-itemsets are constructed from frequent $(k-1)$-itemsets, until no more frequent itemsets can be found.

The bitmap-based algorithm optimises the efficiency of association rule discovery by transferring the data into a bitmap format (i.e., every couple of $< transaction - item >$ is represented by a bit in a bitmap array, with bit i encoding the presence or absence of the itemset in transaction TID_i) [64] . The efficiency of this algorithm emanates from its calculations of the supports of itemsets through manipulating the bitmap together with logical operators. The naive bitmap algorithm (NBM) works directly on the bitmap while the hierarchical bitmap algorithm (HBM) uses a bitmap index to take advantage of the sparsity of a typical bitmap.

The column-wise approach, which uses a column-based data access, is concerned mainly with datasets containing large items (i.e., each transaction contains many different items in contrast to datasets with large numbers of short transactions). It uses intersections to create candidate frequent itemsets and, similar to the Apriori algorithm, uses transactions with horizontal layouts but its advantage is limited by the characteristics of the data to be processed [49].

Yen and Chen [210] proposed a graph-based technique using a vertical bit layout of the data as a starting point (i.e., each item is converted to a binary array of the length of the number of transactions, where 0 means a transaction does not contain this specific item, otherwise the bit is 1). When putting two items through the logic AND operator, with the number of 1s in the result greater than min_supp, a directed edge added between two 1-items points to the one with the higher index (the items are sorted in a certain order). When generating a frequent k-itemset, the last item in the $(k-1)$-itemset is used to extend this itemset into a k-itemset. If there is a directed edge from the last $(k-1)$-itemset to another item, the $(k-1)$-itemset is extended to a k-itemset. The support of the new candidate k-itemset is calculated by applying the AND operator to the itemset's members.

Since the number of records in a dataset can be very large, one type of processing is to simply use a sample of the dataset. In contrast to the abovementioned data reduction method, which marks off transactions according to their usefulness for future scans, a sampling approach proposed by Toivonen et al. [194] samples dataset D and applies ARM to only the sampled data (S), whereby a trade-off between degrees of accuracy and efficiency need to be made. The sample size of S is such that the search for frequent itemsets can be undertaken in main memory and, since the mining is on S, as it is possible that some global frequent itemsets will be missed, a lower support threshold is used for the sample data. After the frequent itemsets (L_s) are found, the original data are used to recalculate the support of them. The sampling approach is especially beneficial when the efficiency of the application is of utmost concern.

Candidate generation algorithms based on the Apriori property can compress the candidate size and there is a rich body of research on improving their efficiency. The two problems of the candidate generation approach mentioned above (i.e., a huge number of candidate sets and repeated scans of the database) can be reduced but not avoided. In contrast, frequent pattern growth algorithms eliminate the need for candidate generation through the creation of pattern growth trees and conditional databases.

The first frequent pattern-growth method (FP-growth) [72], which was proposed by Han et al. to mine a complete set of frequent itemsets without candidate generation, works in a divide-and-conquer way. The first scan of the data extracts frequent 1-itemsets and sorts them in descending order. Then, the database is scanned again to compress the transactions and put item-count information into a FP-tree, which is constructed as follows: the algorithm first creates a root node (marked "null") and then adds a branch for every item in the data, with the order of the items in the branch according to the descending order of their frequencies. During this process if, for one transaction starting with an item, for example, I_1, there is already a branch in the tree with an I_1 connected to its root, this new transaction merges into the existing branch, but separates into a new branch at this item if the next one in the existing branch is different. A count number is attached to each node and, when parts of two transactions overlap on one branch, the counts of the overlapping nodes accrue. On a FP-tree, mining is a bottom-up process in which, starting with a frequent pattern of length 1, the algorithm searches the tree to collect all its prefix paths (the set of prefix paths in the FP-tree that occur with this pattern) and then constructs its conditional FP-tree. Items with counts in the collection of prefix paths of less than *min_supp* are dropped and frequent patterns constructed by concatenating the filtered prefix paths with suffixes, with the support of each itemset that of the least frequent item in that combination.

FP-growth has been shown to be effective in mining datasets with not too many different items but when this number increases, the size of the FP-tree typically expands exponentially due to the reduced number of commonly shared prefixes. Grahne et al. proposed an FP-growth* using an array-based structure to reduce the number of tree traversals [68], the algorithm of which relies on a density heuristic to determine the benefit of constructing an array and its instantiation is generally not guaranteed. The study of Wang et al. [197] improved the FP-growth algorithm by alleviating the need to generate conditional pattern bases. It processes the FP-List in a top-down order, recursively creating conditional FP-Lists.

2.2.2.2 Pruning

In many cases, the basic ARM produces an extremely large number of association rules, often thousands or even millions because, if a pattern is frequent, its sub-sets are all frequent, especially when the *min_supp* set is low. It is almost impossible for end-users to comprehend or validate such large numbers of complex association rules, especially as most can end up being unrelated or uninteresting. To overcome this problem, specific patterns, such as the closed and maximal frequent patterns [145, 97], have been proposed.

It is usually assumed that domain experts know what patterns may be interesting and useful, and the contexts in which they have high possibilities of being discovered. Including this knowledge in the loop of the ARM process to confine the search space is one strategy for pruning, is known as constraint-based mining and includes rule, data and interestingness constraints [138, 104].

Rule constraints specify the form or condition of the rules to be mined. The meta-rule is a rule form whereby the user specifies the form or length of association of interest and the mining algorithm only generates rules of this specific form; for instance, $P_1(X, Y) \wedge P_2(X, W) \Rightarrow$ buys (X, "office software") , where P_1 and P_2 are the predicates the ARM algorithm can match using the attributes and values from a given database. In addition, a user can specify the length of the rule required. The use of the meta-rule as a syntactic or semantic filter to define the forms of interesting single-dimensional association rules was proposed by Klemettinen et al. [102], while a relation-based approach to the meta-rule-guided mining of association rules was studied by Fu et al. [61].

Rule constraints can also set limits on pattern spaces to reduce the number of patterns needed to be checked during an ARM process. According to the type of itemsets to be pruned, there are five categories of pattern mining constraints: (1) if an itemset does not satisfy an anti-monotonic constraint because none of its supersets can satisfy it, its supersets can be pruned; (2) if an itemset satisfies a monotonic constraint rule, so do all its supersets; (3) if a rule constraint is succinct,

the sets that satisfy it can be precisely generated, even before support counting begins; (4) convertible rule constraints describe the constraints that belong to none of the above categories but may become so if the items in an itemset are arranged in a particular order; and (5) inconvertible constraints cannot be transferred to any of the above four constraints. Methods for applying ARM under the above constraints have been discussed in various studies [138, 104, 149]. Constraints can also be applied to the data space, a strategy that prunes pieces of data if they will not contribute to the subsequent generation of satisfiable patterns in the mining process, and can introduce a concept of hiding rules, as discussed by Wu et al. [207].

Interestingness constraints, such as support, confidence and correlation, can be applied after mining to filter out discovered rules. The statistical independence of rules in data mining is studied by Shapiro [150], while interestingness measures of association rules are discussed in a number of studies [5, 33, 16, 206], and those forming filter infrequent patterns are studied in the work of Jin et al. [93]. Ontology-based domain knowledge can be encoded into the post-mining process to preserve only patterns of interest to users [117].

2.2.3 Association Rules for Extended Patterns

In real-world applications, as end-users are often interested in specific frequency patterns, which may require the data types to receive specific treatments, there is a need for research on extended frequent patterns.

Multilevel and multidimensional patterns In many applications with multilevel and multidimensional patterns, it is difficult to find strong associations among primitive data items due to the sparsity of data. However, using an association rule in high-level abstraction can reasonably efficiently reveal interesting patterns in certain applications. Multilevel association rules provide sufficient flexibility for mining and traversal at multiple levels of abstraction whereby one can first mine high-level association frequent itemsets and then only those itemsets the corresponding high-level itemsets of which are frequent [185, 70]. Redundant rules can be

filtered out if lower-level rules can essentially be derived based on higher-level rules and the distributions of corresponding items [185], while efficient mining can also be achieved if *min_supp* at different levels varies. If the LHS of a rule includes multiple predicates, e.g., $age(X, \text{“}20...29\text{”}) \wedge income(X, \text{“}52K...58K\text{”}) \Rightarrow buys(X, \text{“}iPad\text{”})$, it is a multidimensional association rule.

Infrequent and negative patterns an infrequent (or rare) pattern is one with a frequency support below (or far below) a user-specified *min_supp* threshold while a negatively correlated pattern is one in which itemsets X and Y are both frequent but rarely occur together and, therefore, are negatively correlated. Mining rare patterns by pushing group-based constraints was proposed by Wang et al. [196] and mining negative association rules discussed by Savasere et al.[169].

Approximate/Compressed patterns the concept of approximate/compressed patterns was proposed to control the number of patterns found by ARM. A compressed pattern is used to present a pattern cluster. From this respect, frequent patterns are viewed as a set of patterns grouped together based on their pattern similarity and frequency support. In a study conducted by Pei et al. [148], frequent patterns are grouped based on their support, and then the most representative patterns are found for each group. A formulation with the minimum description length (MDL) principle is proposed for selecting representative patterns by Siebes et al.[176].

Colossal pattern some applications may need to mine high-dimensional data; for example, in micro-array data analysis in bioinformatics, researchers are more interested in finding large patterns (e.g., long sequences) than small ones since they usually carry more significant meaning and are called colossal patterns. Zhu et al. [214] investigated a novel mining approach, Pattern-Fusion, for efficiently finding a good approximation to a colossal pattern in which a pattern is discovered by fusing its small fragments in one step whereas incremental pattern growth mining strategies, such as those adopted in Apriori and FP-growth, have to examine a large number of mid-sized ones. There are other patterns proposed in the literature, such as Chiu's nested associative pattern, which focuses on core associative structure

extraction from a dataset [40], and a study that applied different weights to itemsets and transactions [191].

Quantitative association rule although typical ARM methods are applied on nominal data types, relational datasets often involve quantitative attributes, which can be discretised into certain intervals and treated as nominal data before an ARM method is applied. One approach for discretisation is binning [186] whereby attribute values can be discretised by applying equal-width, equal-frequency binning, and then replacing each bin value by a bin mean or median. However, this often results in a huge amount of association rules being mined. In order to overcome this problem, several methods, such as the data cube, clustering-based and statistical analyses, have been proposed.

Applying clustering techniques to mine quantitative association rules can be viewed from two aspects: clustering the attribute values or clustering the association rules. In addition to binning discretisation, clustering is another method for transforming numerical data into categories for ARM and can be applied on each quantitative attribute to find clusters that satisfy *min_supp*. Then, such a cluster can be combined with clusters or nominal values from another cluster to examine the support, with the Apriori property still suitable for pruning during this process. If the current combination does not satisfy *min_supp*, it is not necessary to proceed with further combinations. As for the association rules already generated, clustering is also useful for merging rules into more interesting and interpretable ones. Lent et al. proposed a BitOp method for clustering association rules generated from binned attribute values [111], and clustering two-dimensional quantitative association rules was studied using geometric properties [62, 211]. The main issue with these approaches is the preparation of the datasets before mining. The mining algorithms are applied to the discretised datasets, therefore the quality of the derived ARs relies on the quality of the discretisation process. As for the discretisation pre-processing, characteristics of numeric attributes are in general unknown and it is unrealistic that relevant prior knowledge is always available for determining the best discretisation scheme. Some researchers therefore have proposed to apply evo-

lutionary algorithms to automatically obtain variable intervals. Mata et al. [119] proposed a genetic association rule algorithm (GAR) to find frequent itemsets in numeric databases without needing to discretise the attributes, and the amplitudes of these intervals are decided by the evaluation function of the evolutionary process. However, the encoding is not effective for genetic operators to be performed, when the algorithm is applied to datasets with a large number of attributes. Alatas et al. [8] later proposed a differential evolution algorithm based association rule mining (MODENAR). Instead of searching for frequent itemsets, MODENAR focuses on mining ARs using evolutionary algorithms. Multiple objective functions are used in the proposed algorithm, so that the support and the confidence of potential ARs are both evaluated. There are also other objective functions incorporated to control interval amplitude and comprehensibility. Since the support and the confidence metrics are designed into the objective functions, there is no need to define thresholds for them. MODENAR is thus a database-independent approach.

The definition of an association rule can result in an exponential increase in the number of rules generated. Aumann et al. [13] proposed a new definition of the association rule to overcome this problem, which, rather than converting quantitative data into categorical items, considers distributions of the continuous data via standard statistical measures such as mean and variance. This is a rule of the form: $population\_subset \Rightarrow means\ of\ values\ for\ the\ subset$, where the mean of the subset is significantly different to the mean of its complement in the database (as validated by an appropriate test). Similarly, Zhang et al. proposed a statistical quantitative association rule form based on a statistical property in which the rule's RHS can be any statistic that satisfies its LHS [213].

In general, the techniques reviewed above either convert quantitative variables into intervals or study the behaviours of the subsets of quantitative variables.

2.3 Automatic Hypothesis Generation Strategies: Heuristic Searches

2.3.1 Evolutionary Algorithms

Evolutionary algorithms (EAs) are a class of heuristic methods developed from the idea of natural evolution and survival of the fittest, which were initially designed to model simple evolutionary systems [94]. In general, they embody four basic components in an evolutionary system:

1. populations of individuals competing for limited resources;
2. dynamically changing populations due to the births and deaths of individuals;
3. the concept of fitness, which reflects the ability of an individual to survive and reproduce; and
4. the concept of inheritance, which determines the resemblances between parents and their off-spring.

For the specific computational implementation of such a characterisation, classes of EAs, such as evolutionary programming (EP) [57], evolution strategies (ES) [173], genetic algorithms (GAs) [81], estimation of distribution algorithm (EDA) [107] and differential evolutionary (DE) [123], have been developed.

The paradigm of EP models the evolutionary process with a fixed-size population whereby each individual produces an offspring. These new individuals are merged into the current population to form selection candidates for the next generation, with the top half of the individuals in the enlarged population (according to their fitness values), surviving to the next generation. ES is based on the natural phenomenon that most organisms produce many offspring, with the characteristics of its dynamic that an offspring population of size λ is produced from a parent population of size μ (often $\lambda > \mu$) only through mutation and a new parent population

is generated from either both the previous $\mu+\lambda$ populations or the single λ offspring population.

The distinctive reproduction paradigm of a GA is that it uses a stochastic approach to select individuals to be parents in the mating pool, which is similar to the natural selection process [66, 81, 95]. Its fitness-proportion selection method assigns a probability to each individual according to its fitness in reference to the current population, with the selection process biased to prefer more fit to less fit individuals. An individual with an above-average fitness will produce more than one offspring while those with lower than average fitness values will have less than one offspring [81]. A typical selection operator adopting the above scheme is roulette wheel selection [66]. Another commonly used selection operator is tournament selection, in which an individual is selected by picking the best individual from a randomly chosen population subset [126]. In practice, the selection operator often needs to be adapted to the specific problem under study to assist the search for better solutions.

The new populations are produced through crossover and mutation operators. The crossover operator combines two chromosomes (parents) to produce offspring, simulating the natural mating phenomenon. If the offspring inherit the best characteristics from both parents, then they may perform better than the parents. There are a number of basic crossover operators: one-point crossover, two-point crossover, and uniform crossover [66]. The one-point crossover operator specifies a location on the parents' chromosomes, and all the genes beyond that location are swapped between the parents. The two-point crossover operator calls for two locations, and swaps the genes in-between. The uniform crossover operator uses a fixed mixing ratio to select genes from the two parents. The mutation operator in a GA introduces an unexploited gene into the population to prevent premature convergence. For binary chromosomes, the mutation operator is often implemented by flipping the binary code using a predefined mutation rate.

Learning classifier systems (LCSs) are hybrid machine learning techniques that adopts a GA for rule discovery, and incorporates reinforcement learning or other conventional machine learning techniques for evaluation. Holmes et al. [84] sum-

marised four basic components of LCSs: (1) a finite population of rules (called classifiers) representing the current knowledge; (2) a performance component regulating the interaction between the environment and the classifier population; (3) a reinforcement component assigning rewards to the classifier population; and (4) a discovery component evolving the population of classifiers. The early development of LCS algorithms started with the cognitive system proposed by Holland and Reitman [80]. The immediate drawbacks of early LCSs are the inherent complexity of the implementation and the lack of comprehension of system operation. Later with the development of reinforcement learning, Wilson introduced eXtended classifier systems (XCSs), which are distinguished by an accuracy based fitness, a niche GA and an adaptation of standard Q-learning for credit assignment. XCSs have gained popularity to date [45]. At the same time, there are also other LCS algorithms that have been developed. Stolzmann [189] introduced anticipatory classifier systems (ACSs), which formalised a type of LCSs with a feature of anticipation. ACS uses rules in the form of condition-action-effect, as opposed to the classic condition-action form. Consequently, the system not only specifies what to do in a given situation, but also gives information about what will happen after a particular action is executed. In contrast to ACSs, the sUpervised Classifier Systems (UCSs) are designed to address single step problems such as classification and data mining, in which delayed reward does not have special advantages [27]. UCSs replace the RL component with supervised learning. It demonstrates that a best action map can yield effective generalisations and evolve compact knowledge representations.

An EDA models the generation of a population through sampling a probability distribution model and then selecting the fittest individuals to update this model with, in the following generation, the new population generated from the updated model. This algorithm is concerned with the interrelationships among genes, it models directly through a joint probability distribution (e.g., Bayesian network). For population evolution, it does not particularly apply genetic operators (e.g., crossover, mutation) to the actual population but realises it through evolution of the joint probability model [26].

DE models the evolutionary process by using the differences among individuals to construct candidate individuals, with a new population created through a selection process that compares each candidate individual with a randomly selected individual from the old population [155].

An EA models the process of evolution through its natural resemblance to a swarm of individuals searching for a certain target, not using a pre-planned group search procedure but reorganising as clues regarding the target are encountered. Its simulated evolutionary dynamics produce an adaptive, fitness-biased exploration of the search space and, when the evolutionary process is terminated, the results obtained from that search process (e.g., the best individual found) can be viewed as the search result.

For a problem involving complex non-linear component interactions, there are often two options: either simplify the problem to permit analytical solutions or develop effective computational search procedures for finding solutions to a non-linear complex problem. An EA can serve as a problem-independent paradigm for designing effective search procedures but several instantiation aspects must be designed when applying it to a specific problem, such as: (1) deciding what an individual in the population represents; (2) providing a means for computing the fitness of an individual; (3) deciding how children are generated from parents; (4) specifying population sizes and dynamics; (5) defining a termination criterion for stopping the evolutionary process; and (6) returning the search result. In order for these procedures to be effective, the design decisions must also reflect the properties of the particular class of problems to which they are being applied. EAs have been surprisingly effective in a wide range of problem areas, due mainly to their not making many assumptions about the underlying fitness landscape [66].

However, there are problems on which EAs tend to perform poorly; for example, a search domain constructed by two or more interacting sub-spaces with no intrinsic objective measure for measuring the fitness of each individual. For these kinds of problems, researchers have turned to a natural extension of EAs, co-evolutionary algorithms (CEAs), which offer great potential for this purpose and have become an

important area of research in the field of evolutionary computation.

2.3.2 Co-evolutionary Algorithms

In biological terminology, co-evolution is defined as a reciprocally induced evolutionary change between two or more species or populations [156]. Inspired by this natural process, in the computer science community, co-evolutionary algorithms (CEAs) usually break a problem down into a few sub-solutions, each of which possesses a population for searching for its own best form. A CEA differs from canonical EAs by subjectively determining fitness based on the interaction of an individual with other individuals [152]. In fact, it does not always use multiple populations since its essential feature is the use of an indirect fitness measure. Darwen and Yao [46] demonstrated that co-evolution can also be achieved between evolving individuals within a single population by means of a niching mechanism.

The CEA suggests a divide-and-conquer strategy when the problem under investigation is large and complex and its implementation has two basic levels depending on the types of modules being simultaneously evolved. In the case of single-level co-evolution, each evolving sub-population represents a sub-component of the problem to be solved while a two-level co-evolutionary process involves simultaneous optimisation of the system and modules in separate sub-populations.

Co-evolution can be classified as competitive or cooperative depending on the nature of the interactions. The former is often likened to a predator-prey model in which individuals or populations compete with one another, with the prey implementing potential complete solutions to a problem and the predators individual fitness cases, and has been found to show an arms race phenomenon [11, 162]. The two populations reciprocally drive each other to increased levels of performance and complexity, with the increased fitness of one implying a diminution in the fitness of the other. Such evolutionary pressure tends to produce new strategies in the populations in order for them to maintain their chances of survival.

However, such a co-evolutionary idea has some problems as, when implemented

in a naive way, a number of 'Pathologies', such as cyclic dynamics, loss of the fitness gradient and evolutionary forgetting, can occur. This prevents the algorithm from finding high-quality solutions. The cyclic dynamics depicts a situation in which the solutions found in a population repeat themselves in the evolutionary process. For example, suppose we co-evolve two systems against each other. If, in the first generation, population A finds that solution A1 performs well against the individuals in the current B population, then A will evolve individuals resembling A1. Next, if population B evolves solution B1 that performs well against A1, it will fill its population with B1. Next, if population A discovers A2, which resists B1, population B will then find B2 to counteract A2. Finally, if population A finds that A1 is its best strategy against B2, the evolution starts a loop of strategies. Recently, researchers have applied game theory concepts to better understand the dynamics and pathologies of such CEAs [157], and developed improved algorithms that can overcome problems such as cyclic dynamics.

In contrast, cooperative co-evolution decomposes a problem into sub-components with the aim of finding co-adaptive individuals that together form a complete problem solution [152]. Although each population contains sub-components of the complete solution and evolves separately, the fitness of an individual depends on its ability to collaborate with individuals from other species. This evolutionary pressure favours the development of cooperative strategies and individuals, and the strategies have been successfully applied to a number of applications, e.g., benchmark optimisation problems [151], string matching and NN design [134].

For instantiation, the cooperative CEA needs to specify three components: (1) a decomposition scheme for dividing the complex problem into sub-species; (2) a collaboration scheme in which the individuals from different species can be combined into a complete solution for evaluation; and (3) an evolutionary process in each subpopulation [36]. This algorithm was first introduced by Potter and De Jong [151] for function optimisation and then the authors extended its prototypical idea of decomposing a complex problem into multiple co-evolving species to several modelling principles: firstly, one species represents a sub-component of a potential solution

and complete solutions are obtained by assembling representative members of each of the species; secondly, credit assignment at the species level is defined in terms of the fitness values of the complete solutions in which the species members participate; thirdly, when required, the number of species in the system should evolve; and, finally, the evolution of each sub-population is controlled by an EA. This proposed cooperative CEA was applied on a function optimisation problem with internal evolution (on one of its sub-populations) using a GA (called the CCGA) and showed competitive performances compared with those of a standard EA. The authors also pointed out its potential for extension with other EAs to solving complex problems.

The CCGA was later developed into a general architecture for cooperative co-evolution [152], which introduced explicit notions of modularity for effectively applying EAs to increasingly complex problems. It models an ecosystem consisting of two or more species, which are genetically isolated, with individuals mating only with other members of their species, although the species interact with one another within a shared domain model and have a cooperative relationship. This architecture also introduced a mechanism that allows the emergence of new species for adaptation. Such a dynamic adaptation architecture improves performances by being able to scale up to large and complex problems that often challenge standard EAs.

Collaboration schemes in shared domains were studied in depth in the study conducted by Wiegand [201], with examining a variety of them and providing insights about how to select an appropriate one. This also relates to the concept of cross-population epistasis, that is, the presence of non-linear relationships among genes, which has been an important part of evolutionary computation research [66]. In cooperative co-evolution, partitioning the problem into components may separate related genes into different populations. This study showed that, when there is significant contradictory cross-population epistasis, the co-operative co-evolution design should use more sophisticated collaboration methods. In the case of static function optimisation, using an optimistic credit assignment method is typically a good choice.

The cooperative co-evolutionary approach has been shown to be a powerful

tool for solving complex problems and its performance advantages demonstrated in various studies. In this thesis, a cooperative co-evolutionary technique is adapted for the searching procedure for hypothesis generation in continuous domains.

2.4 Representation of Functional Associative Relation

2.4.1 Artificial Neural Network

The unparalleled intelligence demonstrated by the human brain has attracted scientists to explore its underlying mechanism for centuries. Neurons form the basic functional units of a brain and their biological function in its simplest form is to receive input and produce a response. The massive interconnections among neurons constructs the physical base for the memory, knowledge, skill, experience and thinking of humans' intelligent activities. Interested in simulating the basic functionality of neurons and investigating their consequent behaviours, Warren McCulloch and Walter Pitts [120] proposed the first artificial model of a neuron and, since then, its classification power has attracted the interest of scientists in exploring its potential. This field is commonly referred to as ANNs and, in the following section, a brief overview of it, including the basic structure of an ANN, its application advantage and evolution-based method, is presented.

2.4.1.1 ANN Units and Architecture

Neuron Models: an ANN is a network connecting a group of artificial neurons. The first artificial neuron model, that of McCulloch and Pitts [120], produces binary output, with its inputs connected to it by two types of connections, excitatory (positive weights) and inhibitory (negative weights), as shown in Figure 2.1. A neuron is associated with a threshold value (θ) and, if the net input to the neuron is greater than this threshold, the neuron is supposed to fire. However, as the

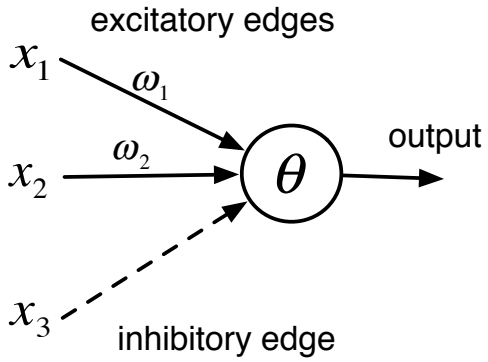


Figure 2.1: McCulloch and Pitts' artificial neuron model

inhibitory input absolutely vetoes the excitatory, if the neuron receives inhibitory input, its output will be fixed at 0 regardless of whether the sum of the excitatory input exceeds the threshold.

This scheme can be used to perform the Boolean logic function with single or multiple neurons. However, as such networks are essentially 'hard-wired' logic devices, they are too inflexible to apply to different systems and require manual designs of their weights and connections. The main importance of this study was that it showed that networks of neuron-like elements could do computations.

Donald Hebb later changed the view of artificial neuron modelling. His proposal, known as Hebb's rule, states that "When an axon of cell A is near enough to excite a cell B and repeatedly or persistently takes part in firing it, some growth process or metabolic change takes place in one or both cells such that A's efficiency, as one of the cells firing B, is increased." [76]. This rule is important as it also points out that the changing of the connection strength is one of the fundamental operations necessary for learning and memory, a perspective that is then reflected in the modelling of an artificial neuron.

The perceptron model [161] incorporates the basic structure and learning behaviour of the McCulloch-Pitts neuron. In essence, a perceptron is a function that maps weighted input values to output. More precisely, given the inputs of a perceptron are x_1 through x_n , its output is:

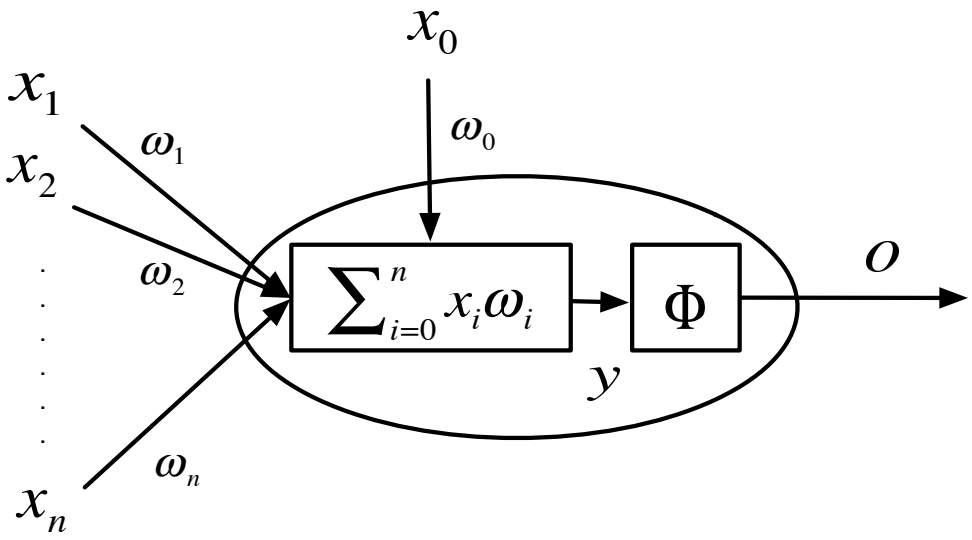


Figure 2.2: General perceptron

$$o = \Phi(\sum_{i=0}^{n} \omega_i x_i) \tag{2.2}$$

The perceptron unit computes a weighted sum of its input and then applies an activation function (Φ) to this sum to derive the output. This model also incorporates learning, which allows changes in the connection weight towards producing the intended output, as discussed in a later section. The step function, which is an active function in Equation 2.3, was used in the early stage of NN development and can represent most Boolean functions, but its structure and learning rule cannot deal with non-linear relations [127]. Therefore, non-linear activation functions and a second training scheme (i.e., the delta rule) were introduced into the family of perceptron activation functions that include the sigmoid, which is used in this thesis (Equation 2.4), piecewise linear and Gaussian.

$$\Phi(y) = \begin{cases} 1 & \text{if } y > 0, \\ -1 & \text{otherwise.} \end{cases} \tag{2.3}$$

$$\Phi(y) = \frac{1}{1 + e^{y}} \tag{2.4}$$

ANN Architecture: although a single perceptron unit can express a number of basic logical functions, as it becomes inflexible when used in more complex non-linear functions, its representation needs its units to be cascaded into networks [198]. The two main elements that affect the expressive power of an ANN are its architecture and weights. Its architecture refers to how it is constructed in terms of its number of layers, number of nodes (each node refers to a perceptron) in each layer, and how its nodes are connected.

An ANN is often represented in a directed graph with arrows pointing in the direction of the information flow. It can be a feedforward NN (FFNN), where there is no feedback connection, or a recurrent NN (RNN), which has feedback (often applied in temporal data analysis) or cyclic connections. Examples of such basic structures are shown in Figure 2.3. ANNs are usually arranged in layers, with a multilayer network having one more layer of units not connected directly to its output, which is called a hidden layer. The number of hidden nodes and their connections influence the overall computational power of an ANN.

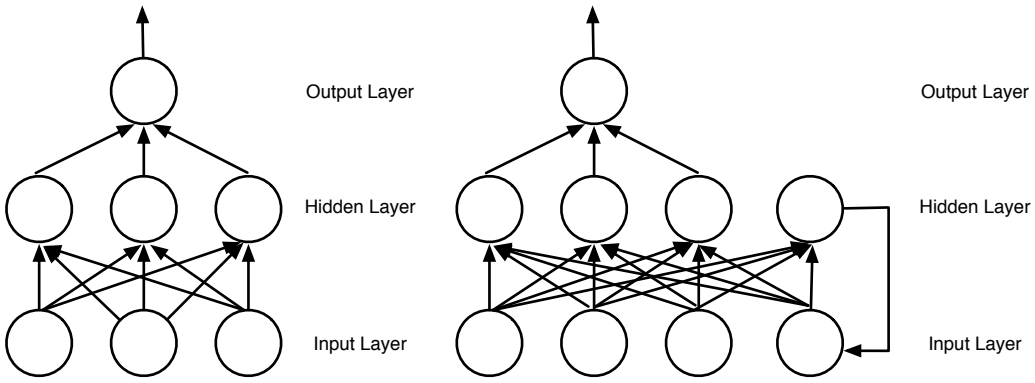
Besides the above basic ANN structures, another type of ANN is Boltzman machine (BM) [4]. A BM is a bidirectionally connected network of stochastic processing units, which can be interpreted as a neural network model. A BM can be used to learn the probability distribution from sample data. However, due to its complex structure, the learning process is difficult and time-consuming. Solutions were later developed by imposing restrictions on the network topology to simplify the learning problem, which lead to restricted Boltzmann machines (RBMs) [179]. RBMs consist of two types of unit: visible units and hidden units. The visible units constitute the first layer of the RBM and correspond to the components of an observation. The hidden units model dependencies between the components of observations. The connections are restricted to be between different types of units. No visible unit is connected to any other visible unit, and no hidden unit is connected to any other hidden unit. Unlike the ANN structure introduced above, the connections in RBM are undirected. The hidden units can be trained to represent the dependencies in the visible units [78]. The states of the visible units can be sampled from the hidden

units [91, 101, 106, 163].

RBMs can be stacked and trained to form deep belief networks (DBNs) [79]. The idea is that the hidden units extract relevant features from the observations. These features can serve as input to another RBM. By stacking RBMs in this way, one can learn features from features so that a high level representation can be achieved. A common DBN architecture has undirected and symmetric connections between the top two layers. The lower layers have top-down directed connections from layers above. The units in the lowest layer represent input data. Deep architectures have the advantage of automating the selection of an appropriate feature space where input instances have desirable properties for solving given problems. Therefore, training deep architectures has been a research focus [25, 109, 31, 130, 131]. An essential idea is to learn a hierarchy of features one level at a time by using the values of the latent variables as input data for training the next layer. It has been empirically shown that DBNs often yields better representations in terms of lower classification error [105], higher quality of the samples generated [168] and invariance properties of the learned features [67].

Another aspect of an ANN architecture is that the units can be fully or partially connected. The computational cost of a partially connected ANN is relatively less than that of a fully connected one when training a network in the same experimental setting. While a carefully designed partially connected ANN can be more precise than a fully connected one, how to select the nodal connections requires either prior knowledge or systematic experiments [108, 51]. Following the development of evolutionary and co-evolutionary computation, incorporating the former to achieve an optimal architecture for an ANN while training it has been explored [1, 134]. Studies show that such integration increases an ANN's representation accuracy and reduces the computational cost.

As the weights associated with each connection of an ANN determine the strength of the input to that node, they also determine the behaviour of the ANN when its architecture is selected. In general, the learning of a NN means changing its weight values so that it outputs expected values, a process discussed in the next



(a) Feedforward neural network (b) Recurrent neural network

Figure 2.3: Examples of basic ANN architectures

section.

2.4.1.2 Artificial Neural Network Learning

The learning of an ANN is achieved by presenting the ANN with a set of training data whereby it changes its weight so that its output approaches expected values. There are a few basic learning rules for this weight adjustment. The first, for an artificial neuron, is based on Hebbs description of how biological neurons learn [76]. It is referred to as the Hebbian learning rule, which states that, if two neurons on either side of a connection are simultaneously activated, the strength of that connection is increased as $\Delta\omega_{ij} = \eta \cdot x_{j \cdot out} \cdot x_{i \cdot in}$, where ω refers to a connection weight between two neurons (x_i and x_j). Although the Hebbian learning rule is biologically plausible (similar behaviour has been found in the hippocampus), it is unstable in functional representation as chance coincidences can build up a connection's strength.

Another type of learning rule is based on minimising some error function and measuring the difference between the expected and real outputs, that is, $\Delta\omega = \eta(y - \hat{y}) \cdot x_i$. It adopts a supervised learning scheme in which the outputs from the training set (y) guide the weight adjustment. In contrast, competitive learning, usually suitable for finding clusters within data, is an unsupervised learning scheme in which the output units compete to respond to a sub-set of the input data, with the

winning unit updating its weights using the learning rule, that is, $\Delta\omega_{ji} = \eta(x_i - \omega_{ji})$. A final type of learning is Boltzman learning in which a network is constructed by stochastic binary-state neuron-like units with defined energy measures. The value of the energy measure of each unit determines the probability of that unit being in an on or off state. Weight learning maximises the probability distribution of the training data representing the network, with the learning rule $\Delta\omega = \eta \cdot (p_{ij}^{+} - p_{ij}^{-})$, where p_{ij}^{+} and p_{ij}^{-} measure the associations between two units, one derived from each of the training and re-constructed training examples respectively, which push the network to approximate the distribution of the training examples.

Backpropagation (BP) the FFNN is a popular ANN architecture commonly used in different applications, which adopts the learning rule based on error correction, that is, the systematic adjustment of network weights using the BP approach [165]. This thesis applies this strategy in the process of hypothesis generation searching.

BP is built on the gradient descent concept. The following presents how a single perceptron weight is learnt using gradient descent and then BP is employed for network weight learning. The learning process of a perceptron is finding the weights that best fit the examples, which is usually transformed into minimising the predictive error between the example output (y) and the output calculated from the unit ($\hat{y}$) as:

$$E(\omega) = \frac{\mathbf{1}}{\mathbf{2}} \sum_{\mathbf{d} \in \mathbf{D}} (\mathbf{y_d} - \hat{\mathbf{y}}_\mathbf{d})^\mathbf{2} \tag{2.5}$$

where D is a set of training examples and E a function of ω as $\hat{\mathbf{y}}_d = \Phi(\omega, \mathbf{x})$. The gradient search minimises E by modifying ω towards the deepest descent at any given point of ω, with the direction of descent calculated by the derivative of E with respect to ω as:

$$\nabla E = [\frac{\partial E}{\partial \omega_0}, \frac{\partial E}{\partial \omega_1}, ..., \frac{\partial E}{\partial \omega_n}] \tag{2.6}$$

As, in order to achieve a decrease in E, each weight should be modified towards this direction, the modification rule for weights becomes:

$$\omega \leftarrow \omega + (-\eta \nabla E(\omega)) \tag{2.7}$$

For each individual weight (ω_i),

$$\omega_i \leftarrow \omega_i + (-\eta \frac{\partial E}{\partial \omega_i}) \tag{2.8}$$

where η is called the learning rate, which controls the weight-changing rate, with a small one resulting in a smooth trajectory but slower convergence. A large step size speeds up the learning but may cause instability (e.g., as the minima are missed) [75]. The selection of the error rate is often through a trial-and-error approach. This learning method requires the activation function to be differentiable, e.g., sigmoid, which has the very convenient property that its derivatives are easily expressed in terms of its output. When applying this gradient descent learning method to a multilayer network, the error term for the hidden units is not immediately available but, fortunately, we can backpropagate it from the output to the hidden layer. The adjustment of weights can be carried out after presentation of either each example (incremental learning) or the entire set of examples (batch learning). Algorithm 2 presents the incremental learning of a NN constructed with sigmoid perceptrons, which updates the weight after each example is presented to the network. In batch training, the weight error term ($\Delta_1\omega_{ij}, \Delta_2\omega_{ij}, \dots$) is computed for each example, with the overall weight update computed when all training examples have been put through the network (the end of an epoch), that is, $\Delta\omega_{ij} = \sum_{l=1}^{n} \Delta_l\omega_{ij}$, where n is the number of training examples.

As the goal of training is not to represent the training data but to be able to perform further tasks, after the training phase, ANN is applied to new tasks of the same kind, which is its generalisation aspect. A network that is not sufficiently complex can fail to fully detect the relationships in complicated datasets, which

Algorithm 2: Backpropagation algorithm for learning in feed-forward neural network using sigmoid activation function (adapted from [128])

Input : training examples ($\mathbf{X}$), each with input vector ($\mathbf{x}$) and output vector ($\mathbf{y}$), $\mathbf{X} = (\mathbf{x}, \mathbf{y})$; FFNN
Output: trained neural network

foreach *weight* (ω_{ij}) **do** initialise weights
$\omega_{ij} \leftarrow$ *a random number*
repeat
  foreach *example* ($\mathbf{x}, \mathbf{y}$) *in training example* **do** propagate input forward through network
  Calculate output (o_k) from each unit ;

  foreach *Network output unit (k)* **do** calculate its error term
  $\delta_k \leftarrow o_k(1 - o_k)(y_k - o_k)$

  foreach *hidden unit (h)* **do** calculate its error term;
  $\delta_j \leftarrow o_h(1 - o_h) \sum_{k \in outputs} \omega_{kh}\delta_k$

  Update each network weight (ω_{ji})
  $\omega_{ji} \leftarrow \omega_{ji} + \eta\delta_j x_{ji}$
until *stopping criteria satisfied*;

leads to under-fitting. An ANN that is too complex may fit the noise as well as the main underlying mechanism. This over-fitting is particularly dangerous because it can easily lead to predictions of unseen data far beyond the range of the training data. This generalisation problem can be seen as a bias/variance trade-off. The bias refers to the model fitting specific training data and the variance to it being sufficiently flexible to fit a variety of training sets [177].

Training with noise: one approach for alleviating over-fitting is to train with artificial noise in the training examples as, if we have two cases with similar inputs, the desired outputs will usually be similar. This means that we can take any training case and generate new ones by adding a small amount of noise to the input and then the output can be assumed to be the same as that before the addition of noise [9, 85].

Early stopping: the idea of the early stopping approach for handling over-fitting is to stop the training before it starts to over-learn the training data. It splits

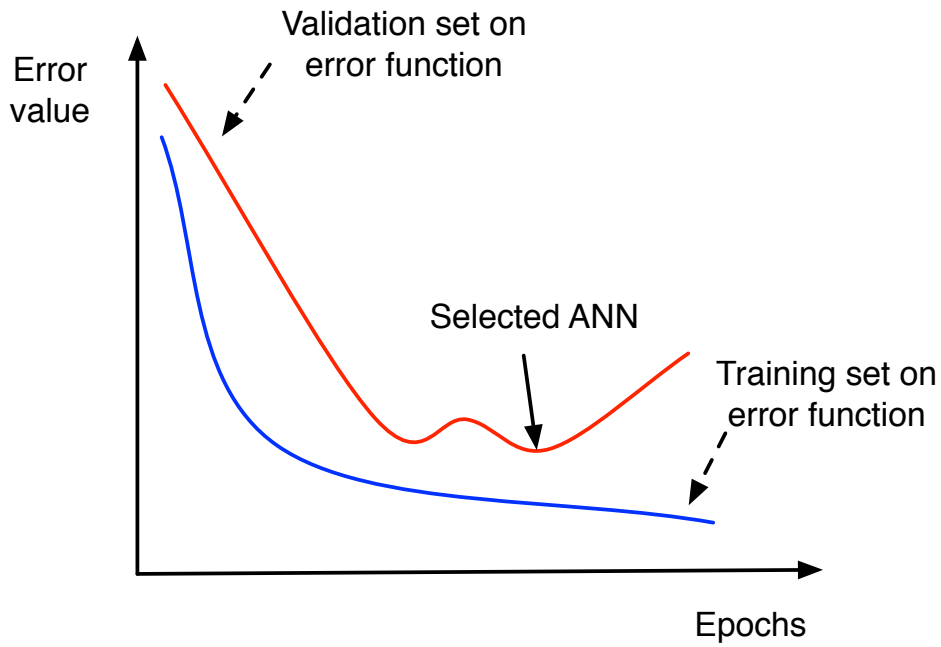


Figure 2.4: General early stopping scheme

the given data into two sets, training and validation, and uses the latter to measure the generalisation of the ANN. During the overall training phase, the error value on the training dataset decreases but, at some points, that on the validation data increases, which means that the ANN starts to memorise the training set. Therefore, in order to preserve generalisation, the training should stop. An early stopping approach has several advantages: it is fast (compared with cross-validation); and can be applied to an ANN with a large number of weights (thereby alleviating the difficulty of designing an ANN structure). In most training exercises, the validation curve is not very smooth and can rise and fall during iteration, as shown in Figure 2.4. The method most commonly used is to train the ANN to converge and select the ANN that has the lowest validation error [154]. In this thesis, early stopping is applied.

Weight decay: this adds a penalty to the error function used in BP with its common form being the sum of squared weights times a decay constant, as shown in Equation 2.9. As large weight values can cause excessive variations in an ANNs outputs if the output unit is not bounded [65], this penalty term penalises them and

causes them to converge to smaller absolute values. Other penalty terms, including weight elimination, as in Equation 2.10, are also used. Weight decay can require different decay constants for different types of weights (i.e., input to hidden, hidden to output) and adjusting the decay constant can be computationally expensive.

$$E_\omega = c \times \sum_i \omega_i^2 \tag{2.9}$$

$$E_\omega = \sum_i \frac{\omega_i^2}{\omega_i^2 + c^2} \tag{2.10}$$

2.4.1.3 Advantage of Approximating Functions

A FFNN can be viewed as a practical tool for approximating non-linear functions, which means that it can be used to capture hidden relationships of a non-linear nature as an alternative to commonly used statistical tool regression. This feature is supported by the universal approximation theorem first presented by Cybenko in 1989 [44], which states:

Let $\varphi(\cdot)$ be an arbitrary non-constant, bounded, and monotone-increasing continuous function. Let $X \subseteq R^m$, X is compact. The space of continuous functions on X is denoted by $C(X)$. Then $\forall f \in C(X)$, $\forall \varepsilon > 0 : \exists\ n \in N$, a_{ij}, b_i, $w_{i \in R}$, $i \in \{1, ..., n\}$, $j \in \{1, ..., m\}$, such that:

$$A_n(x_1, x_2, ..., x_m) = \sum_{i=1}^{n} w_i \varphi(\sum_{j=1}^{m} a_{ij} x_j + b_i) \tag{2.11}$$

can be considered an approximation of the function f:

$$\|A_n - f\| < \varepsilon \tag{2.12}$$

The compatibility of the universal approximation with FFNN is obvious: the activation function used in FFNN (e.g., sigmoid) can be a non-constant, bounded and

monotone-increasing function as $\varphi(\cdot)$; the FFNN can use m input units with n units in a hidden layer, where each hidden unit consists of a connection weight (a_{ij}) and bias (b_i); and the FFNN output can be a linear combination of the outputs from hidden units. The universal approximation theorem describes that the standard multilayer FFNN with a single hidden layer containing a finite number of hidden neurons and an arbitrary activation function are universal approximators in $C(R^m)$. Hornik emphases that it is not the specific choice of the activation function but, rather, the multilayer feedforward architecture itself, which gives a NN the potential to be a universal approximator [86]. Most non-continuous functions can also be approximated by a FFNN under Lusin's theorem that any finite and measurable function is continuous in most of its domain.

Barron states that, for a single-layer FFNN trained by BP, the total risk (R), the mean squared error between the target function and function estimated by the FFNN, is bounded by $O(\frac{C_f^2}{M}) + O(\frac{Mp}{N} logN)$, where C_f is the first absolute moment of the Fourier magnitude distribution of the target function (f), M the number of hidden units, p the number of FFNN inputs and N the number of training examples [18, 19, 75]. This analysis implies that the size of the training example does not need to be exponentially large to achieve a good approximation generalisation. As previously mentioned, in this thesis, early stopping is used to control generalisation.

The universal approximation theorem states that a function can be represented by a FFNN but does not specify how to determine this FFNN with the stated property. In addition, the functions being approximated are usually unknown and the number of hidden units cannot be set to be unlimited. In practice, the units in a single-layer FFNN tend to influence each other and result in unstable approximation performances when the function is complex. Often, in practice, a two-layer FFNN is used for complex function approximation [63, 37]. By using two hidden layers, the units in the first layer can respond to local features and partition the input space into regions while those in the second layer receive the output from the first layer and learn global features [75]. In this thesis, both single- and two-layer FFNNs are used for function approximation depending on the complexity of the underlying

relations hidden in the data.

2.4.1.4 Evolutionary Artificial Neural Network

Unlike a NN using the Boltzman learning rule, which inherently incorporates the feature that it can escape poor local optima to some extent (i.e., stochastically activated unit states), in general, NN training based on gradient descent will be trapped in local minima if no additional adaptive procedure is incorporated. This training only guarantees that the error function value towards the direction of a local minimum is reduced. Some common solutions to this problem include selecting proper initialisation weights through domain knowledge or trial-and-error and adding a perturbation to the weight or architecture during training. However, the combination of EAs and ANN provides a better remedy and also brings other appealing characteristics to ANN training and application, e.g., an adaptive ANN architecture and weights. The following section reviews applying a combination of EAs and ANN, which is often referred to as an evolutionary ANN (EANN).

Yao [208] described the combination of EAs and ANN on three levels of evolution, those of the weights, learning rule and architecture as shown in Figure 2.5, with the lower the evolution, the faster it is.

Evolution of ANN weights: applying an evolutionary approach for ANN weight training is relatively straightforward and includes representations of the ANN's weights and evolutionary schemes. Representing weights in the genotype can be either binary or real numbers [208]. A binary representation [200, 34, 187, 92] is a natural extension of a standard EA, where the weights of an ANN are converted into binary bits and concatenated to a genotype chromosome, and can adopt standard evolutionary operators. However, such a scheme often results in a large chromosome, which consequently affects the efficiency of the evolutionary process. A real-number representation of a chromosome has the advantage of accuracy of representation but needs new genetic operators or a different evolutionary strategy [54, 20, 132, 121]. Montana et al. [132] showed that, using specifically designed genetic operators, the

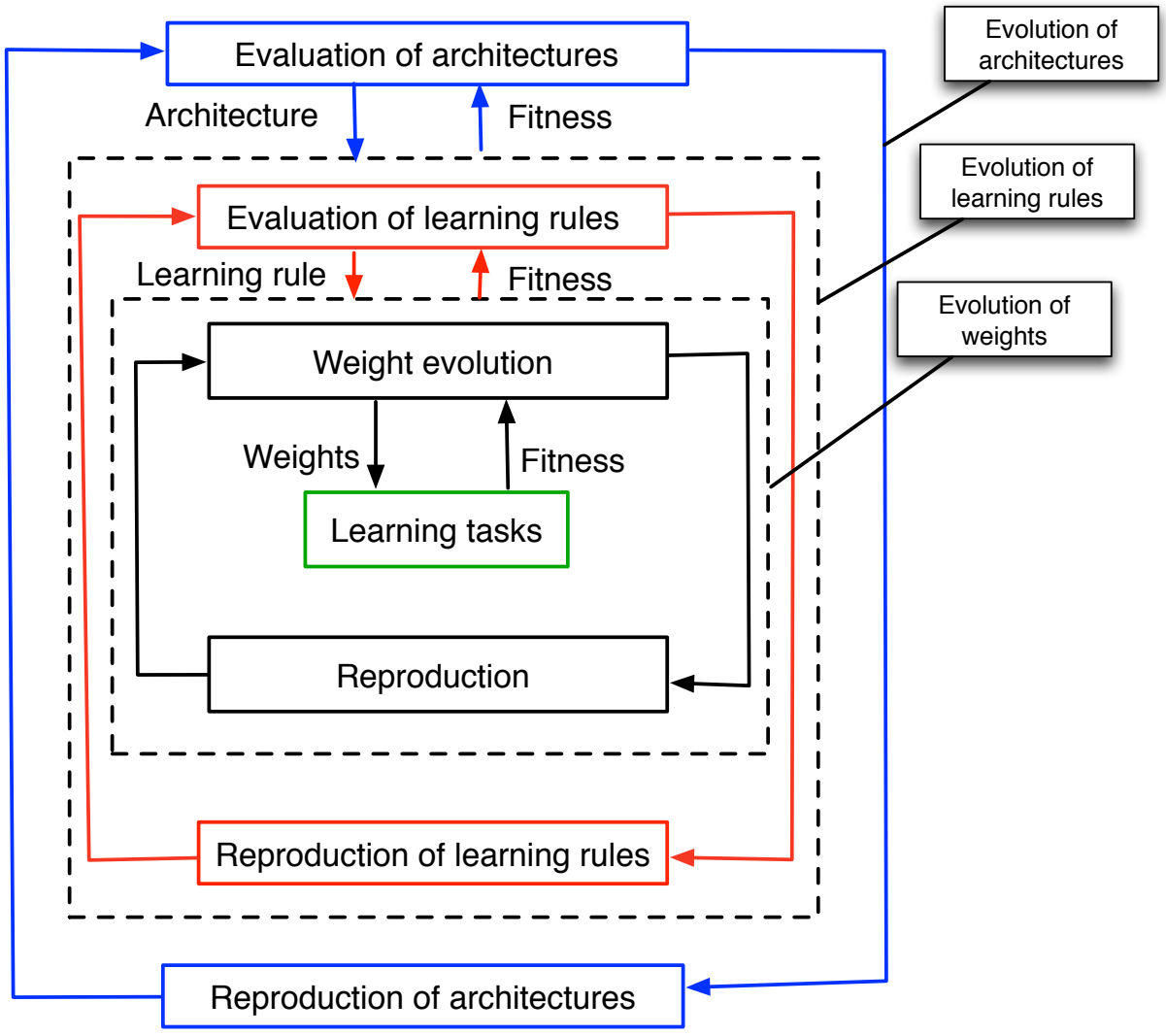


Figure 2.5: General framework for EANN (adapted from [Yao 1999])

EANN performs much faster than BP. Regarding different strategies, EP and ES have been used as the evolutionary component of an EANN due to their advantages for continuous domains [209].

A hybrid scheme that evolves an ANN with a local search is an efficient way of accelerating the training process. For a search space with multiple local minima, an EA can be used to search for the location of a basin where a global minimum may be within the reach of a local search so that the local search can identify it [22, 110]. This scheme is adopted in this thesis to identify the initial weight sets for an ANN in terms of function approximation.

Evolution of ANN structure: another concern regarding evolving ANN weights is that, although studies have shown that combining evolution with ANN training achieves better performance in terms of accuracy and less sensitivity to initial conditions [55, 56, 20], the design of an ANN still depends on either prior knowledge or experimenting with a number of different ANNs, which indicates the need to adaptively improve the ANNs' architecture. The evolution of an ANN architecture has two different chromosome design schemes: direct encoding (i.e., the chromosome encodes all information about the architecture) and indirect coding (i.e., the chromosome encodes the main parameters or provides a compact representation of the architecture).

In direct coding, the architecture is often encoded into a matrix, in which the entities represent connections between two units (on or off) in the ANN, which is then converted into a vector-like chromosome. In the evaluation phase, the chromosome is translated into an ANN, which is randomly initialised and trained on a given task, with the training error incorporated in the fitness measure. Because of the flexibility of an EA, the architecture of an ANN can be measured in multiple dimensions using complexity, statistical and information theory measures [53, 28]. Although such a coding scheme is very simple to implement and can generate a competitive ANN architecture [171], it can be affected by noise during evaluation; for example, as the initialisation procedure with random values introduces noise into the training, different initialisations can cause different evaluation results, thereby leading to

inaccurate evaluations. Fortunately, this problem can be alleviated by evolving weight information with the architectural matrix. One hybrid evolution approach is to replace a binary representation of the architecture with values representing the weights on corresponding connections [103, 30, 140, 118]. However, to use real-value representations to evolve both the architecture and connection weights, an issue is how to design the evolutionary operator. Pareto-front differential evolution is used by Abbass [1], and the proposed evolutionary scheme, which is enhanced with a local search (e.g., ANN architecture evolution plus BP training), is often referred to as memetic evolution. In order to preserve building blocks of potential solutions, one option is to adopt only mutation operators for evolution [166][209].

As this encoding scheme for the overall architecture often ends up in a large genotype representation and increases the computational time required for evolution, another approach, the indirect encoding scheme, has been used by many researchers [100, 73, 74, 48]. In it, only important parameters are used for genotype representation while other details are predefined by prior knowledge. However, although this coding produces a compact genotype, the evolution might not be sufficiently flexible to find an ANN with good generation [100, 129, 124]. NeuroEvolution of augmenting topologies (NEAT) proposed by Stanley et al. evolves both the weights and the structure of an ANN [188]. It employs crossover of different topologies, structural innovation protection, and incremental structure growing to increase the efficiency of the evolutionary ANN. Although it has the advantage of autonomously determining both the weights and the structure of an ANN for a particular learning task, this is achieved through the evolutionary process. In the context of FARM, as we have introduced in the first chapter, FAR searching itself is an evolutionary process. If for each potential FAR, another internal evolutionary process is introduced, the computational complexity will become exponential. Therefore, in this thesis, NEAT is not considered for FAR evaluation.

Evolution of learning rules: a last perspective concerning ANN training is the learning rule. For different ANN architectures, different learning rules will affect their learning performances. Therefore, it is of interest to researchers to adopt

a learning rule appropriate to the particular ANN architecture and given tasks. Such research starts with evolving the BP parameters (e.g., learning rate and momentum) [22, 73] and then progresses to evolving the learning rule itself (i.e., how the weights are updated). The challenge lies in designing the representation of the general learning rule, which is usually achieved by assuming that the updating of weights is based on the local variables (e.g., states of the connected units and previous connection weights), with the learning rule the same for all connections. Chalmers [35] designed a learning rule representation based on forming a linear relationship using local variables, which shows that the evolution produces the commonly used delta rule. Similar studies have been conducted [24, 23, 59, 21] and Nolfi et al. and Parasi et al. [139, 143] emphasised the importance of environmental diversity whereby a variety of architectures and learning tasks is available for evolution.

2.5 Causal Hypothesis: Causal Models

2.5.1 Rubin Causal Model

The Rubin causal model, also known as the 'Neyman-Rubin causal model', is a family of approaches to the statistical analysis of cause and effect based on the potential outcome framework. This framework refers to an experimental design principle that every unit has different potential outcomes depending on its *assignment* to a condition. In the terminology of this model, *treatments* are variables that are conceptually manipulable and *units* the objects to which these treatments are assigned. *Responses* are any variables the values of which may have been affected by their treatments and concomitants any variables the values of which are unaffected. A causal study under the potential outcome framework aims to find the relative effect of treatments on the responses of selected units with given values of concomitants.

To explain this through an example, if a study aims to provide evidence for the effect of multivitamin supplement tablets on reducing levels of cardiovascular disease risk, it uses two treatments, taking multivitamin tablets (treatment a) and

taking placebos (treatment b). For each unit (patient) in the study, after a certain time period (δt), there are two potential outcomes. If it received treatment a, the cardiovascular disease risk level would be Y_{ua} and, if it received treatment b, it would be Y_{ub}, with the difference between Y_{ua} and Y_{ub} due to exposure to different treatments. In addition, the difference between Y_{ua} and Y_{ub} can tell us how much the level of cardiovascular disease risk for unit u would change if treatment a were used instead of treatment b.

This is the strategy used in our causal hypothesis generation, with the reverse engineering scenario that allows for the values of the variables to be manipulable. In studies in uncontrolled situations, such as those involving humans, we cannot assign two treatments to one unit at the same time. The suggested solution is to create two groups of units, with one receiving treatment a and the other treatment b. However, since units are not exactly the same as individuals, differences in variables other than the treatment may affect responses. The strategies studied include randomised experiments [137], matching [164], blocking and stratification [164].

2.5.2 Automatic Causal Modelling

While the major means of causal investigation is based on the potential outcome framework, there are obvious practical and ethical considerations that limit the application of randomised experiments in many instances, particularly those involving human beings. Several techniques for representing causal relationships and inferring them from purely observational data, which rely on the relationship between causation and probability, have been developed.

A causal model in this branch of study consists of two parts, a directed acyclic graph (DAG) and a distribution over a set variables (X). Each x_i is expressible in the form of $x_i = f_i(pa_i, u_i)$, $i = 1, 2, ..., n$ and, in its general form, f_i is not committed to a certain function but assumed to be in the form of a deterministic, functional equation [147]. The main reason for preferring a deterministic form is that it is a more general representation as every stochastic model can be emulated by many functional

relations (with stochastic inputs) but not the other way around. Commonly used causal models in the literature are structural equation models and causal Bayesian networks. By incorporating a graph into a causal model, the directionality of the underlying process can be expressed by a prototype , where f is represented by a linear equation [205].

The connection between causality and probabilities is established by the conditional dependence among the variables and is built upon certain assumptions, the Causal Markov Condition (CMC) and Faithfulness [147]. The CMC in a DAG is that, given a set of variables (V), each variable in V is independent of the variables that are not its parents or descendants, given its parents. V is said to be causally sufficient if, and only if, there is no variable (C) not in V that is a direct cause of more than one variable in V. The CMC places a constraint on the probability distribution of the variables so that their joint probability distribution satisfies it. The Faithfulness assumption states that, for a causally sufficient set of variables (V), every true conditional independence relationship in the density over V is curtailed by the local directed CMC for the causal DAG. This assumption means that the causal structure fully determines the independence and dependence among the variables under investigation, which, if there is external perturbation, stay the same.

To identify the above causal model, there are two basic categories of algorithm for learning a causal network without intervention: score- and constraint-based. The score-based approach generally defines a scoring function for each network structure, which represents how well it fits the data, with the goal being to find the highest-scoring one, which, in general, is a NP-hard problem. Cooper and Herskovits [41] proposed a Bayesian scoring metric (log-likelihood) and a heuristic search algorithm called K2 for learning a network structure when the data is fully observable. As the log-likelihood metric itself favours graphs with many edges, later, a Bayesian information criterion [172] with an additional penalisation to favour graphs with fewer parameters was applied. Another metric based on a similar idea is the minimum description length [159]. However, the possible number of graphs grows exponentially with increasing numbers of nodes [160]. The score-based approach often applies a

Markov process to deal with this curse of dimensionality [114] [42]. Friedman et al. proposed the sparse candidate algorithm for reducing searching complexity by focusing on a relatively small number of candidate parents for each variable [60]. The optimal re-insertion [133] constrains the search on a candidate's parent set and corresponding child set. The greedy search algorithm [38] searches for DAGs of only an equivalent class to further constrain the search space, and is guaranteed to return an optimal structure if there is a faithful DAG.

Constraint-based approaches look for the constraints, e.g., conditional independence, in data and return a Markov equivalent class. When testing for conditional independence, the SGS algorithm introduced by Spirtes et al. [183] tests every possible conditioning set. It was later developed into a PC algorithm where the testing of conditional independence is reduced to the variables connected by directed or undirected paths to the variable under test [182]. It is faster than the SGS algorithm but can produce errors in removing arcs although Li and Wang showed how to control the false positive rate while using it [112]. Inductive causation (IC), introduced by Pearl and Verma, is a variation of the SGS algorithm, which starts by generating an undirected graph based on the dependencies between variables, as opposed to using a completely undirected graph as in SGS, and takes into account latent variables [146] [195]. The general procedure for the IC algorithm and its orientation is as follows [147].

1. For each pair of variable a and b in V, search for a set S_{ab} such that $a \perp\!\!\!\perp b \mid S_{ab}$ holds in $\hat{P}$. Construct an undirected graph G, such that vertices a and b are connected with an edge if and only if no set S_{ab} can be found.

2. For each pair of nonadjacent variables a and b with a common neighbour c, check if $c \in S_{ab}$.
   If it is, then continue;
   If it is not, then add arrowheads pointing at c (i.e. $a \rightarrow c \leftarrow b$).

3. In the partially directed graph that results, orient as many of the undirected edges as possible subject to two conditions: (i)orientation should not create a new $v-$structure; and (ii) the orientation should not create a directed cycle.

The Step 3 of above algorithm can be systematised in following ways [147]:

Rule 1 : Orient $b - c$ into $b \rightarrow c$ whenever there is an arrow $a \rightarrow b$ such that a and c are nonadjacent.

Rule 2 : Orient $a - b$ into $a \rightarrow b$ whenever there is a chain $a \rightarrow b \rightarrow c$

Rule 3 : Orient $a-b$ into $a \rightarrow b$ whenever there are two chains $a-c \rightarrow b$ and $a-d \rightarrow b$ such that c and d are nonadjacent.

Rule 4 : Orient $a - b$ into $a \rightarrow b$ whenever there are two chains $a - c \rightarrow d$ and $c \rightarrow d \rightarrow b$ such that c and b are nonadjacent and a and d are adjacent.

IC is a general framework summarised from previous work on causal search models [183][146] [195]. As input, its algorithm takes a stable probability distribution ($\hat{P}$) generated by an underlying autonomous mechanism that can be represented by a DAG, and outputs a pattern that represents the equivalence class of the underlying mechanism. The idea at its simplest level is that a certain pattern of dependence among variables implies causal relations, that is, when the independent relations (both conditional and unconditional) are eliminated from variables, the remaining links imply causal relations. The constraint-based algorithm outputs a set of Markov-equivalent causal models but does not show how much better one model is than another as it has only one test criterion, conditional dependence. In addition, due to a constraint-based approach's multiple testing, it is prone to over-fitting if not specifically controlled.

2.6 Hypothesis Generation Research

A hypothesis is traditionally constructed by researchers to formulate a potential knowledge of interest and, in hypothesis testing research, means a "specific proposition about the behaviour of a (biological or other) system, based on a logical reasoning that leads to an experimentally verifiable prediction that is either confirmed to be consistent with it or not" [98]. Constructing a hypothesis of this type requires the researcher to manually select the relevant factors and design experiments to test the hypothesis; for example, in the study in concerning marital happiness [29], the hypothesis states that marital happiness and a certain gene 5-HTTLPR are correlated, which is based on domain knowledge and the interest of the researcher. To test this hypothesis, experiments need to be manually designed and data precisely collected. Such hypothesis testing research adopts a reductionist strategy, also known as a 'bottom-up' approach, which asserts that, if we can break a system into its component parts and understand them and their interactions, we can intelligently reconstruct the system. Therefore, hypothesis testing research is a relatively slow and incremental process.

Sensing, data collection and data storage have advanced over the years, with scientists starting to piece together techniques for automating the process of knowledge discovery. Such work is usually motivated by the research requirement to alleviate the excessive workloads of human experts and speed up the knowledge discovery process, a trend that has given rise to hypothesis generation research. Hypothesis generation infers knowledge from data via various kinds of pattern recognition techniques and is not a new concept. Technological advances allow the handling of overwhelming amounts of data, thereby promoting the advantages of hypothesis generation which is a cross-disciplinary practice that different research fields apply differently according to their domain knowledge and research interests. A variety of hypothesis representations can be found in hypothesis generation approaches.

King et al. designed a robotic scientist [99] that can automatically identify the yeast metabolism pathway but has not yet been tested and validated. This

work was demanded by the fact that, despite being one of the best understood of organisms, the functions of about 30 % of the 6,000 genes of yeast (S. cerevisiae) are still unknown. However, since the biochemical equations of its basic metabolic pathway are known, how its biochemistry is related to genetics can be left to the machine to investigate.

The author designed a directed graph to encode the prior biological knowledge. Such a graph allows an algorithm to compute the phenotype of a particular knockout or to infer a missing reaction that could explain an observed phenotype. This domain knowledge assists in forming the basis of the hypothesis generation approach for inferring new knowledge. In the study carried out by Moss et al. [135], the researchers designed an ontology-driven system for monitoring real-time data to detect and explain anomalous patient responses to treatments in an intensive care unit. The hypothesis is represented by logic rules that express possible reasons for the detected anomalies. The knowledge about treatments and responses are first built in terms of ontology and then hypothesis generation is executed by following the strategies human clinicians use for reasoning.

The hypothesis representation can take the form of a probability model [199]. Such hypothesis generation is to produce a set of possible matches between extracted image features (R) and pre-calculated model features (M). The hypothesis is presented as a Markov random field variable ($X_{M,R}$), which indicates that region R in the image arose from object region M. Similarly, in the study of image processing in [39], the representation of a hypothesis is the probability of two points arising from the same model.

The hypothesis can also be represented by graph models and refers to the different phenomena in biological signalling networks, e.g., certain reactants lead to certain products [167]. Based on a basic biological signalling network, an hypothesis can be generated automatically by solving the problems of the constrained downstream and minimum knockout. Graph models are also used for hypothesis representation of causal structures, as reviewed in Section 2.5.

It can be seen that hypothesis generation is a cross-disciplinary practice. There is a variety of forms of representation for hypotheses depending on the specific research fields and research knowledge. Hypothesis generation provides an inductive paradigm that automatically finds interesting patterns in data as potential knowledge. Conventionally, representations reflect the specific domain knowledge and interest of the researchers, that is, the researchers have a clear idea of how to use the data or the knowledge they expect to extract from them. Guided by such intentions, an hypothesis representation is selected and a generation strategy developed. However, in a situation for which knowledge about the objective system and how to use the data is not available in advance, this system is different from systems in the conventional sense. Traditionally, a system is defined as a group of components interacting together to achieve an overall objective or a number of objectives known as the system-level objectives [90]. This thesis is concerned with systems that can be measured by a set of variables but their objectives or underlying structures are unknown; for example, sensor networks, large data problems, intelligent environments and cyber security. For a large data problem [116] for which data are recorded on a daily basis but not according to specific research concerns, one aspect is that people are interested in analysing the data. However, if no specific domain knowledge is provided in advance, how to use the data is not known. Therefore, hypothesis generation potentially provides a paradigm for establishing interpretable knowledge from such situations.

In an intelligent environment, sensors record activity data from the environment and pre-defined tasks can be performed by adapting these data to human activities through machine learning techniques [122]. Although such techniques can be designed manually when we know the common activities in typical scenarios, e.g., offices and lecture rooms, we need new ways for an intelligent environment agent to build up an hypothesis about how to understand and adapt to non-standard scenarios as a priori knowledge about what the observational data describes is not available.

The cyber security scenario has similar characteristics. Due to the constant

evolution of hacking activities, previous knowledge about abnormal activities in the log data can be outdated. How to use log data to actively acquire an updated insight into a system remains an interesting direction for which agents that can generate new hypotheses from data could be an advantage [174, 175].

In situations with the above features (e.g., systems measured by a set of variables and domain knowledge not well established), one type of knowledge that can provide initial insights into the underlying system is that regarding causal structures. Identifying a causal structure that controls the dynamics of a system can build up the knowledge base for further use, e.g., visualisation, control strategy. Existing principles and models for studying causation are reviewed in Section 2.5, and automatic causal modelling techniques can potentially be used for the above purpose. However, for the systems considered above, the assumptions set for automatic causal modelling may not be satisfied, which raises the need to define such a generalised hypothesis generation problem and its corresponding approaches. This thesis focuses on formulating a generalised hypothesis generation problem and developing its solutions.

2.7 Chapter Summary

This chapter reviewed the literature concerning general hypothesis generation as well as the techniques related to different aspects of causal hypothesis generation; that is, ARM, causal modelling, ANNs and heuristic searches. These approaches provide principles and techniques for representing potential associative relations, automatically mining associative relations and undertaking causal modelling. However, their advantages for a generalised hypothesis generation problem have not been widely considered. Likewise, no systematic approaches exist for developing or evaluating computational methods for interrogating an unknown system to gain new insights. These issues are addressed in the following chapters.

Chapter 3

Hypothesis Generation for Continuous Domains: Problem Definition and Solution Design

The previous chapter reviewed the techniques related to hypothesis generation approaches and discussed the problems associated with their adaptation to continuous domains. This chapter integrates the advantages of ARM and the counterfactual causal modelling criteria, with the aim of formally defining the problem of hypothesis generation for continuous domains and distinguishing it from other similar problems investigated in the literature. Based on this definition, the design of solutions to the problem is also discussed.

This chapter is organised as follows. Section 3.1 defines the problem of hypothesis generation for continuous domains and decomposes it into two sub-problems: the associative hypothesis generation problem (AHGP) and causal hypothesis generation problem (CHGP). Section 3.2 discusses the issues that need to be addressed when designing solutions to the AHGP and describes how they are considered. Section 3.3 presents an experimental causal search approach based on an artificial agent architecture as a solution to the CHGP. Finally, the conclusions drawn from this study are presented in Section 3.4.

3.1 Hypothesis Generation For Continuous Domains: Problem Definition and Comparisons with Similar Problems

One of the aims of this thesis is defining the problem of hypothesis generation in continuous domains, with the context being situations in which data can be collected about an unknown system. The unknown system is measured by a set of variables. It is a system with conventional definition (A group of components interacting together to achieve an overall objective or a number of objectives, known as system level objectives [90]). However, limited *a priori* knowledge is available for characterising the structure and dynamics of system. Hypothesis generation in such a situation can provide a paradigm for gaining new insights into the relevant data and the system.

The ultimate goal of investigating a system often boils down to studying the causal structure that dominates it, which provides us with the knowledge to develop further control and application strategies. In general, as a hypothesis generation approach, automatic causal modelling [183, 147] deals with a similar problem, as discussed in Chapter 2. However, the algorithms developed for automatic causal modelling are based on a set of assumptions, which place constraints on the underling system and observed variables. The context considered in this thesis does not strictly satisfy these assumptions, the characteristics of the underlying system are unknown. In addition, for continuous domains, automatic causal modelling adds additional assumptions about variables; for example, that they follow the same distribution while relationships among them are assumed to be linear. In contrast, the hypothesis generation problem defined in this thesis does not assume specific structures and characteristics of the variables. The proposed problem of hypothesis generation in continuous domains is defined as follows.

Autonomous Hypothesis Generation Problem in Continuous Domains

Input: observational dataset ($\mathbf{X}$) measured from unknown system (E). $\mathbf{X}$ has

p instances, each of which (X) a vector comprised of n variables and $X = [x_1, x_2, ..., x_i, ..., x_n]$, where x_i continuous variable.

Output: causal graph (G) describing underlying interrelations among variables in X, with each node in G a variable (x_i); edges in G with arrowheads represent direct causal relation between two variables and if arrowhead from variable x_i to x_j, x_i is a direct cause of x_j.

In the case of human knowledge discovery, causation discovery is a progressive process, with the understanding of the causal law behind a system beginning with the observation of associations, which leads to an inquiry into causal relations. Similarly, the hypothesis generation problem above can be decomposed into two progressive sub-problems: the AHGP and CHGP. As shown in Figure 1.1, the first problem is to find associative patterns, which provide evidence of potential causal relations. However, without *a priori* knowledge about the system, it is possible that there are no specific causal relations between its variables and, if so, it is not necessary to proceed to causal hypothesis generation. Also, associative hypotheses can potentially reduce the number of variables that needs to be examined when forming causal hypotheses and, because their generation procedures exclude irrelevant variables, the CHGP can take advantage of the output from the AHGP to form a causal hypothesis in a reduced variable space.

3.1.1 Definition for the Associative Hypothesis Generation Problem

The AHGP, the first sub-problem of the general hypothesis generation problem, is defined as follows: X is a vector of variables x_i; and the associative hypothesis generation procedure outputs several associative variable sets, that is, the variables within each set are associated with each other in a certain way.

Sub-problem 1: Associative Hypothesis Generation Problem (AHGP)

Input: observational dataset ($\mathbf{X}$) with p instances; each instance (X) vector comprised of n variables and $X = [x_1, x_2, ..., x_i, ..., x_n]$, where x_i continuous variable.

Output: set ($\mathbf{F} = \{f_1, f_2, ..., f_m\}$) of associative hypotheses; m number of hypotheses generated from observational data ($\mathbf{X}$). f_j a single associative hypothesis taking the form, $X_A \Rightarrow X_B$, $X_A \cap X_B = \emptyset$, $X_A, X_B \subset X$.

The above definition of the AHGP is similar to the general definition of the ARM problem. The difference is that the AHGP aims to extract associations by examining variables whereas the dominant ARM techniques for continuous variables firstly convert the values of the variables involved into intervals and then extract associations from them. However, discretising the variables presents difficulties for obtaining evidence as to how individual variables are related to other variables. The AHGP does not convert variables into intervals, but defines associative relations directly among variables.

3.1.2 Definition of the Causal Hypothesis Generation Problem Definition

The second sub-problem is the CHGP, the input for which is a set of associative hypotheses ($\mathbf{F}$=$\{f_1, f_2, ..., f_m\}$), with the hypothesis generation task to retrieve the underlying causal relations among the variables based on the evidence provided by $\mathbf{F}$.

Sub-problem 2: Causal Hypothesis Generation Problem (CHGP)

Input: set ($\mathbf{F} = \{f_1, f_2, ..., f_j, ..., f_m\}$) of associative hypotheses and m number of associative hypotheses.

Output: causal graph (G) describing underlying interrelations among variables in X; edges in G with arrowheads represent direct causal relation between two variables and if arrowhead from variable x_i to x_j, x_i is a direct cause of x_j.

The final causal hypothesis representation takes the same form as that used in automatic causal modelling techniques in the literature, but the problem definition takes different inputs and is set in a different context. Assumptions about the presence of causal relations within the objective system are not predefined. For example, the patterns among variables are not limited to the consequences of the causal relations among them. The patterns can be due to associations or coincidences, that is, the underlying system measured is not necessarily comprised of only inter-connected components. This allows for a situation in which the underlying system contains independent components that do not form a system in the conventional sense. The relaxed assumptions about the underlying system in this thesis distinguish the proposed CHGP definition from that for automatic causal modelling in the literature [183, 147].

3.2 Associative Hypothesis Generation: Functional Association Rule Mining

The previous section defined the problem of hypothesis generation for continuous domains and decomposed it into two sub-problems. This section describes these sub-problems and the issues that need to be addressed when solving them.

In this thesis, we identify three issues that need to be addressed when designing algorithms for solving the AHGP: the representation of the associative hypothesis, the strategy for generating associative hypotheses, and the strategy for evaluating associative hypotheses.

There is a variety of representations of associative relations ranging from condi-

tional probability [183], correlation and regressions among variables of interest [96] to association rules [6]. This thesis focuses on representing them in terms of the last form. Although approaches for extracting association rules can automatically group multiple interrelated variables together, a typical association rule definition does not reflect such information for continuous variables but, in general, converts them into intervals, which it treats as categorical data. However, as the AHGP defined above requires establishing relations among variables rather than intervals, in this thesis, instead of discretising the continuous variables into intervals, an alternative association rule form, termed FAR (functional association rule), is proposed.

FAR is a specific representational form of the associative hypothesis for continuous variables. It takes the form $f(X_A) \Rightarrow X_B$ $(X_A, X_B \subset X, X_A \cap X_B = \emptyset)$, which is interpreted as the variations of the variables in X_B that can be predicted by the variations of the variables in X_A. However, although it can establish associative relations among variables, it has the problem that it cannot enumerate all the functional expressions. Therefore, it is imperative to either constrain the associative relations of interest to fixed forms or select a general form that can represent all possible relations. In this thesis, both are considered, with the fixed relation form assumed to be linear and the ANN adapted to represent arbitrary functional relations.

The second issue to be addressed for associative hypothesis generation is the evaluation criteria, which are required to confirm whether a generated FAR is supported by the observational data. Such identification implies applying a machine learning technique to fit a FAR to given observational data and, if the data support it, preserving that FAR as a valid hypothesis for upper-level hypothesis generation. In this thesis, a predictive accuracy measure is used to perform the validation, that is, if the accuracy of the FAR predicting values of the variables on its right-hand side (RHS) exceeds a certain threshold, this rule is considered a valid associative hypothesis.

The last issue to be addressed is the process of generation itself, which is usually automated as a search procedure. The following two sections present details of the

designs of the search strategies. Both of the two FAR representation forms are discussed, fixed function form and arbitrary function representation.

3.2.1 Linear Functional Association Rule Mining

3.2.1.1 Linear Functional Association Rule Representation

The first associative hypothesis representation considered in this thesis is fixed function form representation, i.e., the associative relations are assumed to be linear. The corresponding functional association rule is then called linear functional association rule (LFAR). The proposed LFAR form only designates the variables involved in an associative relation. It does not represent how they are associated. Therefore, it needs another representation to express the specific relations among these variables.

This thesis proposes to use the regression model to represent such linear relations constructed by a LFAR. Although, the ultimate goal of this thesis is identifying arbitrary associative relations, the LFAR is studied as a starting point of the hypothesis generation approach. In addition, LFAR can be applied to conventional association rule mining tasks as an alternative association rule form. It is necessary to conduct an initial study on extracting LFARs from data as an association rule mining approach.

3.2.1.2 Search Strategy for Linear Functional Association Rule Mining

For simplicity, the search for the LFAR problem is termed LFARM, the objective of which is to find as many LFARs as possible hidden in the observational data. The search space is defined by all possible LFARs among the variables as $f(X_A) \Rightarrow X_B$, $X_A, X_B \in X$, $|X_B| = 1$. As an initial study on functional association rule mining, the number of the RHS variable of a LFAR is constrained to 1. The search space has the following features:

1. infinitely large since the number of possible combinations is large enough to be considered unbounded;

2. deceptive since similar combinations can perform quite differently; and

3. multimodal since we are expecting there to be multiple validation rules existing in the data.

For dealing with such a difficult, rugged, multimodal search space, evolutionary computational algorithms are often more efficient than other techniques [208]. An evolutionary algorithm (EA) works with a population of candidate solutions, which concurrently explore different parts of the search space, with a crossover operator working on different multiple genes at the same time, and it has the potential to preserve interrelated variables in its building blocks. EAs as heuristic search methods are known for their robustness and low sensitivity to noise, therefore they can be applied for designing the search approach for the LFARM problem. This design involves three main issues: (1) the encoding scheme for the LFARs; (2) the objective function design for guiding the evolution; and (3) the EA for evolving the chromosomes.

***Encoding scheme*:** there are two schemes for encoding a set of rules into chromosomes: Pittsburgh [178] and Michigan [82]. The former encodes a set of rules in one single chromosome, which has the advantage of evaluating the cooperative performance of a set of rules on certain domain tasks. From this point of view, it is a direct extension of the EA for a supervised learning problem. In the latter approach, each individual encodes only a single rule, with each member competing with every other member for evolution priority. It is adapted in this thesis, with every rule representing a potential LFAR in its search space and no specific domain task requiring all the rules to participate together.

***Objective function*:** the evaluation of each individual involves the learning task of fitting the encoded LFAR to the observational data. In essence, a single LFAR is a regression model, which predicts the values of its RHS variables using a linear function constructed with its LHS variables. The learning task in the evaluation process is to estimate the parameters of the encoded linear function. As previously mentioned, predictive accuracy is used to evaluate a LFAR validation

level. Since the variables under study are continuous, the predictive accuracy used here is the correlation between the predicted and observed RHS values. The squared correlation co-efficient r^2 is used for computing this correlation as in Equation 3.1:

$$r = \frac{\sum_{i=1}^{k}(X_{Bi} - \bar{X}_B) \times (X'_{Bi} - \bar{X}'_B)}{\sqrt{\sum_{i=1}^{k}(X_{Bi} - \bar{X}_B)} \times \sqrt{\sum_{i=1}^{k}(X'_B - \bar{X}'_B)}} \tag{3.1}$$

where $\bar{X}'_B$ here refers to the predictive value of X_B from variables in X_A, k is the number of observations. $\bar{X}_B$ and $\bar{X}'_B$ are the means of actual values of X_B and predicted values of X_B respectively.

***Search strategy*:** previous analysis of the search space feature of LFAR has shown the advantages of applying an EA to FARM. The general structure of the EA adapted to FARM is as follows, and the detailed implementation is introduced in Chapter 4.

1. collect observational data;
2. initialise a population of chromosomes, each of which encodes LFARs;
3. evaluate each individual and assigns it a fitness value;
4. produce a new generation of LFARs using evolutionary operators, i.e., crossover and mutation;
5. repeat the process from step 3 until the stopping criteria are met.

3.2.2 General Functional Association Rule Mining

3.2.2.1 General Functional Association Rule Representation

The LFARM approach presented above uses regression models to capture linear associations; nevertheless, one of the main interest of this thesis is to investigate the possibility of generating an associative hypothesis for arbitrary relations. In order to generalise the representational power of the FAR, a general FAR is proposed.

The representational form of this general FAR is the same as that of LFAR, that is $f(X_A) \Rightarrow X_B$, $X_A, X_B \subset X$, $X_A \cap X_B = \emptyset$, except that the function f is not constrained to a linear function but can be an arbitrary function.

The design of FARM algorithm needs to consider the following aspects. Firstly, unlike a linear relation that has a fixed representational form, a nonlinear relation can take any form. Therefore, a general representation of an arbitrary function needs to be designed which, in this thesis, is an adaptation of an artificial neural network (ANN). There have been concerns about applying ANNs to represent relations due to their inability to interpret the relations. In our FARM scenario, the essential interpretation we are looking for is to find which set of variables are associated, this is done using ANN predictive models. In addition, ANN based FARs also refine this information with a quantitative prediction. An example of a traditional AR is that milk and bread are often bought together. This rule specifies related variables, but does not include information about how much milk is bought when a certain number of bread is purchased. In comparison, ANN based FARs can append quantitative information to the rule. It builds up a functional relation among the variables through the ANN model.

Secondly, similar to that of LFARM, the evaluation of a FAR involves a machine learning practice that fits the FAR to the observational data for which back-propagation (BP) is used. Finally, the search scheme of FARM has different features from that of LFARM due to its ANN representation. Since the training of an ANN is itself a search problem, a few factors affect the performance of the ANN approximation (e.g., how its architecture is designed; how its initialisation of weights is conducted). These factors need to be taken into account when designing the mining approach for FAR.

3.2.2.2 Search Strategy for General Functional Association Rule Mining

Although the FAR search strategy using ANN as a general representational form has the potential to capture non-linear relations, the performance of such an approx-

imation is constrained by the ANN's architecture and initialisation [208]. For the best evaluation of an arbitrary FAR, an appropriate ANN, which can be designed by either applying domain knowledge or casting it as a search problem, should be assigned. This thesis considers the latter because its autonomous hypothesis generation approach assumes minimum *a priori* knowledge. Therefore, another layer, a search problem, is introduced on top of the FARM problem. For this FARM problem, on one hand, the FARs in the observational data are searched for interesting associations and, on the other hand, the most appropriate ANN for each FAR's representation and evaluation is also searched. The combination of potential FARs and their matching ANNs form the search space for the FARM problem, for which evolutionary computation again possesses advantages. This search space is:

1. infinitely large since the number of possible combinations is unbounded;
2. deceptive since similar combinations can have quite different FAR evaluation performance outcomes;
3. multimodal since we expect there to be multiple validation rules existing in the data; and
4. non-differentiable since changes in the ANN initialisation weights or FAR construction can have a discontinuous effect on evaluation performance.

The paradigm of the cooperative co-evolution architecture [152] is to decompose a complex problem into several sub-problems, each of which represents a partial solution and evolves in its independent sub-population. However, as the fitness of one individual is determined by the complete solution in which it participates with individuals from other sub-populations, the overall evolutionary process favours cooperative individuals. The search problem of FARM discussed above can be naturally mapped to this cooperative co-evolutionary prototype. Constructing a potential FAR is only a part of the solution to finding valid FARs in observational data while the other part is determining cooperative ANNs that best match the FARs. Based

on such concerns, the search strategy designed for the FARM problem in this thesis is a cooperative co-evolutionary FARM algorithm.

Both an ANN's architecture and initialisation affect its approximation performance on a potential function. In this thesis, only the proper ANN initialisation is evolved for cooperation with the FARs and also, for a FAR evaluation, the proper ANN architecture search can be designed in the cooperative co-evolutionary approach. However, incorporating an additional search sub-problem into the FARM problem significantly increases the complexity of the search. For a novel definition of the FARM problem, this thesis is firstly concerned with a concise design for the search solution.

A properly initialised ANN for a FAR evaluation is important, especially when the BP approach is used. As reviewed in Chapter 2, BP is a gradient descent technique in which the training can easily be trapped in a local optimum. For a potentially valid FAR, it is possible that, in its evaluation, the learning task could mean a multiple modal search space for the ANN training. If initialisation of the ANN can be located close to the slope of the global minimum, the BP may have a better chance of identifying a valid FAR. Therefore, the search for appropriate ANN initialisation is incorporated in the cooperative co-evolutionary solution. The general structure of its search algorithm is as follows, the detailed implementation is introduced in Chapter 5.

1. collect observational data;

2. encode FAR and ANN into chromosomes and initialise the parameters for their sub-populations;

3. apply a combination scheme to form a complete solution for the individuals in each population, evaluate each solution and return fitness values to the constituent individuals;

4. apply an evolutionary operator to the respective population to form the next generation;

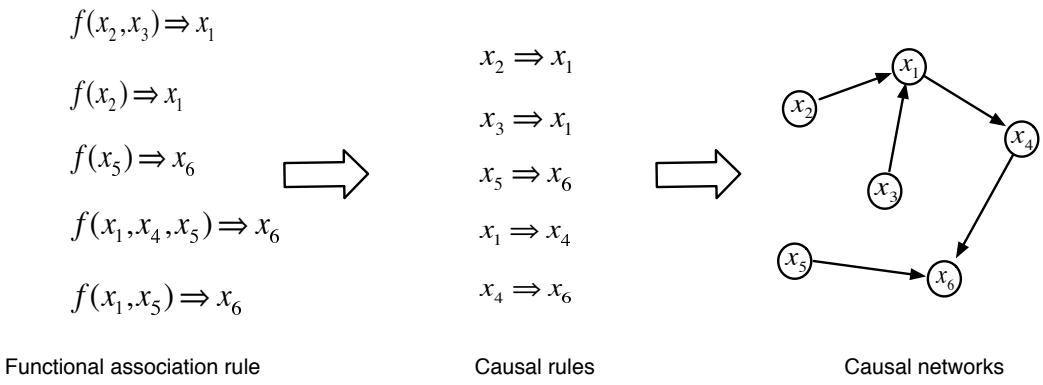


Figure 3.1: Visualisation of tasks for causal hypothesis generation

5. repeat step 3 until the stopping criteria are met.

3.3 Causal Hypothesis Generation: Experimental Causal Search

The CHGP defines the task of examining whether the input FARs contain valid causal relations, which are then integrated into a directed graph to generate the final causal hypothesis about the underlying system (as in Figure 3.1). To accomplish this, the main problem to be addressed is the criteria for identifying the causal relations.

Automatic causal modelling, as in the constraint-based causal search algorithms [182, 147], relies on conducting dependency tests to identify causal relations. When the variables under study are continuous, additional assumptions are made, i.e., that all follow the same statistical distribution and the interrelations among them are linear. In this thesis, such assumptions are not specifically set for the variables under study but, instead, the causal relations identification problem is approached from a systematic experimental perspective based on an agent architecture.

In hypothesis testing research, investigations on causal relations are generally conducted by applying the Rubin causal model (details of which are provided in Chapter 6). A causal effect is estimated by comparing the difference between the out-

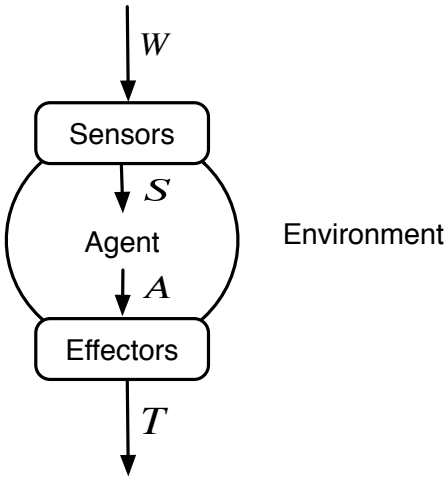


Figure 3.2: Agent in environment (adapted from [123])

comes for a supposed effect variable from two different experimental settings. In one experimental setting, the supposed causal variable is given a treatment/intervention, whereas in the other experimental setting, there is no treatment/intervention. Other variables involved in the supposed causal relations are adjusted to eliminate their potential influence on the effect variable. This basic principle of the causal effect inference incorporates temporal information which, in general, is considered essential for identifying potential causal relations [158]. It is also one of the most critical factors that people use for distinguishing causal relations from other types of associations.

Inspired by the above principle, the strategy planned for addressing the CHGP is to automatically apply a systematic manipulation using an agent [204]. An agent is considered as a system that uses sensors to monitor some subset (S) of the world (W), reasons about the sensed world and uses a set of actions (A) to trigger effectors that cause a subset of transitions (T) to occur, as shown in Figure 3.2. Such an architecture defines three fundamental characteristics of an agent as encompassed by the systematic manipulation approach for CHGP: (1) it obtains inputs through sensing, (2) it makes decisions and (3) it acts through effectors.

3.3.1 Sense

The sensed information an agent receives from its environment includes the observational data and FARs from previous processes. The observational data ($\mathbf{X}$) are collected from the unknown system, where no specific domain knowledge is available about its structure and dynamics. The FARs are the associative hypotheses ($\mathbf{F} = \{f_1, f_2, ..., f_m\}$).

3.3.2 Reason

In hypothesis testing research, the relevant variables required to be included for experiments are manually selected by the researchers based on their domain knowledge. The FARs mined from the observational data in the previous step can be considered automatic counterparts of this practice as a FAR of form $f(X_A) \Rightarrow X_B$ provides evidence of which variables can potentially serve as cause variables (X_A) or effect variables (X_B). Therefore, intervention experiments will be applied on the LHS of a FAR.

3.3.3 Action

The systematic interventions the experimental causal search algorithm applies on the objective system form the action part of the agent. At an abstract level, the action is first to adjust the system of interest to a state the same as, or similar to, a history state according to the observational record. Then, an intervention is applied to the supposed cause variable and, after a certain time step (Δt), the value of the supposed effect variable recorded. If a change in the effect variable after intervention is confirmed, the causal relation is established, otherwise the relation is merely an association. For example, the currently received FAR is $f(x_1, x_2) \Rightarrow x_3$ and, in the observational history, there is a record of the variables: $x_1 = x_{(1,c)}$, $x_2 = x_{(2,c)}$ and $x_3 = x_{(3,c)}$. Suppose that the current causal relation under intervention is $\{x_1, x_3\}$, and the values of the variables x_1, x_2 and x_3 are adjusted to the above record

($i.e., x_{(1,c)}, x_{(2,c)}, x_{(3,c)}$ respectively). Then a disturbance Δx is added to variable x_1 ($x_1 = x_{(1,c)} + \Delta x$). After a time step Δt, the value of the effect variable (x_3, $x_3 = x_{(3,\Delta t)}$) is compared with its value without disturbance ($x_{(3,c)}$). During this process, the value of the variable x_2 is kept unchanged ($x_{(2,c)}$) in order to eliminate its potential influence on x_3.

The typical cycle of the proposed experimental causal search process is:

1. collect observational data;
2. receive a FAR;
3. select a LHS variable, adjust the system to a history state, apply the intervention, control other LHS variables and record the change in the RHS variable;
4. repeat the last step a certain number of times on different history states of the system;
5. reason about the causal relation between the LHS and RHS variables under study;
6. if not all the LHS variables have been tested, return to step 3; or
7. if all the LHS variables have been tested, return to step 2.

The word 'experimental' is used to reflect the interaction of the agent with the given unknown system. By integrating the experimental causal search algorithm into an agent, the agent has the potential to autonomously acquire knowledge about its environment, as explored in Chapter 6.

3.4 Chapter Summary

This chapter proposed a generalised hypothesis generation problem in continuous domains, which was further decomposed into two sub-problems, the AHGP and CHGP, the formal definitions of which were also presented. It discussed appropriate

strategies for developing solutions to the two sub-problems; that is, a novel FAR form was proposed to represent the associative hypothesis and further refined into linear and general FARs, and then the AHGP was cast as a FARM problem. Two approaches based on evolutionary computation were proposed, with a prototypical experimental causal search algorithm presented for causal hypothesis generation. The following chapters present the detailed implementation of the algorithms described in this chapter, together with the subsequent experiments conducted on both synthetic and real-world data.

Chapter 4

Associative Hypothesis Generation: Linear Functional Association Rule Mining

The previous chapter defined a generalised hypothesis generation problem, which was then decomposed into two sub-problems, the associative hypothesis generation problem (AHGP) and causal hypothesis generation problem (CHGP), the definitions of which were also presented. For the AHGP, a functional association rule (FAR) was proposed as its representation. The generation strategy was cast into a heuristic search process, the preferred method for which was using evolutionary algorithms (EAs) as they are often very efficient for complex, multimodal and discontinuous search spaces. The previous chapter also described the strategies for developing linear FAR mining (LFARM) algorithms.

This chapter presents details of the implementations of three different EAs, a genetic algorithm (GA), population-based incremental learning (PBIL) and differential evolution (DE), adapted for the LFARM problem. To analyse the LFARs generated from the search process, two metrics, hypothesis complexity and variable perceptual selectivity, are proposed. Experiments are conducted on both a synthetic dataset and four real-world datasets. The remainder of this chapter is organised as

follows: Section 4.1 provides details of the methodology for LFARM, including the chromosome encoding scheme, objective function design and the three search algorithms for mining LFARs; Section 4.2 explains the proposed metrics for analysing the hypotheses generated; the experimental results are presented in Section 4.3; and the conclusions drawn discussed in Section 4.4.

4.1 Methodology for Linear Functional Association Rule Mining

A LFAR can be viewed as an alternative form of the quantitative association rule. As discussed in Chapter 2, the general quantitative ARM usually applies a strategy of converting continuous data into intervals whereby the data form can be adapted to the support and confidence framework of the classic ARM. However, as a LFAR describes an associative relation in terms of functions for which the downward closure property in ARM does not hold [7], the idea of generating a complete set of LFARs has to be abandoned. However, instead, we can adopt heuristic search approaches in which the algorithms return as many valid LFARs as possible. This section presents the algorithms designed for the LFARM problem and includes: (1) the scheme for encoding a LFAR into a chromosome; (2) the evaluation strategy and objective function design for assessing the individuals; (3) the procedure for extracting valid LFARs; and (4) implementations of the three EA-based LFARM approaches, the general process of which is illustrated in Figure 4.1. The termination criteria used is whether the specified number of generations for evolution is reached.

4.1.1 Linear Functional Association Rule Representation

As defined in Section 3.1.1, Chapter 3, the input observational data for FARM consist of n continuous variables. Accordingly, a chromosome for a LFAR is designed to be a binary vector of length n, with a gene in it referring to a bit and corresponding to one variable in an observed instance (X). The values 1 and 0 in a chromosome

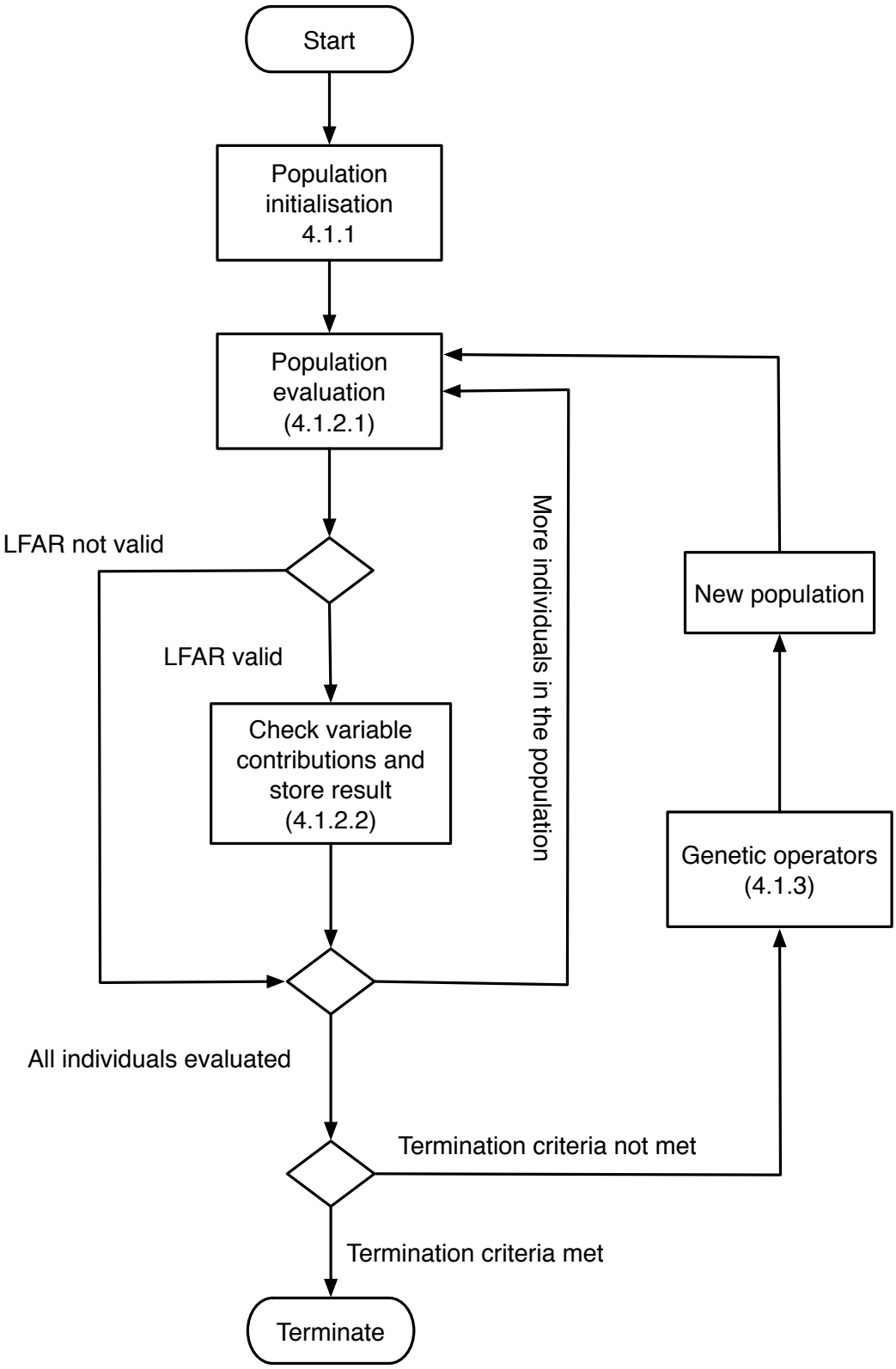


Figure 4.1: General process for evolutionary algorithm-based LFARM (numbering refers to subsections of this chapter)

Table 4.1: Examples of individual representation of LFAR

Chromosome	LFARs	Chromosome	LFARs
11100000	$f(x_4, x_5, x_6, x_7, x_8) \Rightarrow x_1$	11001100	$f(x_3, x_4, x_7, x_8) \Rightarrow x_1$
	$f(x_4, x_5, x_6, x_7, x_8) \Rightarrow x_2$		$f(x_3, x_4, x_7, x_8) \Rightarrow x_2$
	$f(x_4, x_5, x_6, x_7, x_8) \Rightarrow x_3$		$f(x_3, x_4, x_7, x_8) \Rightarrow x_5$
			$f(x_3, x_4, x_7, x_8) \Rightarrow x_6$

represent its right-hand side (RHS) and left-hand side (LHS) variables respectively. Examples of such chromosomes are given in Table 4.1. As this encoding scheme incorporates several LFARs in one chromosome, it allows one chromosome to investigate multiple rules at the same time, which increases its chance of finding a valid LFAR. For initialisation, each gene in a chromosome with a probability 0.5 is assigned value 1, otherwise 0. If all genes in a chromosome are either 0s or 1s, then the chromosome's fitness is set to 0.

4.1.2 Evaluation Strategies

4.1.2.1 Objective Function

In the conventional ARM, the 'interestingness' of an association rule is determined by its support and confidence measures [7] and, for one LFAR, by how well it is supported by the observational data. In the last chapter, we stated that, in essence, a LFAR is a regression model in which the variables involved are automatically selected by the evolutionary process. In contrast, conventionally, variables involved in a regression model are manually selected by analysts [96]. A LFAR in the form of $f(X_A) \Rightarrow X_B$ specifies the variables involved in a linear relation, e.g., $f(x_1, x_2, x_3) \Rightarrow x_4$, and its analytical expression can be written as:

$$x_4 = b_0 + b_1 x_1 + b_2 x_2 + b_3 x_3 \tag{4.1}$$

The parameters, $\{b_0, b_1, b_2, b_3\}$, are estimated by a multiple regression procedure which fits the observational data ($\mathbf{X}$) to the equation [3] (Algorithm 3, Line 4). Then, the quality of the corresponding LFAR can be evaluated and, with the complete analytical expression, can be used to estimate the values of its RHS variables. The predictive accuracy (R^2) (as in Equation 3.1, Chapter 3) determines the interestingness of one LFAR (Algorithm 3, Line 5).

According to the above encoding scheme, each chromosome can include multiple LFARs, each of which has its own predictive accuracy value ($R^2_{r_{i,j}}$), where $r_{i,j}$ refers to the ith chromosome in a population and j the jth LFAR encoded in one chromosome. The objective function of the chromosome is then determined by the maximum prediction value among the LFARs as:

$$\vartheta_{r_i} = \max_j \ R^2_{r_{i,j}}; j = 1, 2, ..., n_{r_i} \tag{4.2}$$

n_{r_i} is the number of LFARs encompassed in one chromosome. This overall evaluation procedure for one chromosome r_i is given in Algorithm 3.

Algorithm 3: LFARM chromosome evaluation: *regressionEvaluation*(r_i, $\mathbf{X}$, h_r)

Input : chromosome (r_i), observational data ($\mathbf{X}$), predictive accuracy threshold (h_r)
Output: fitness value (ϑ_{r_i}), valid LFARs recorded ($\mathbf{F}$)

```
1  n_{r_i} ← number of FARs in r_i
2  for j ← 1 to n_{r_i} do
3  |  Decode jth LFAR from rule r_i, denote as LFAR_{i,j}
4  |  Apply multiple regression on LFAR_{i,j}, and estimate regression parameters
5  |  Calculate predictive accuracy (R^2_{r_{i,j}}) of current LFAR_{i,j}
6  |  if R^2_{r_{i,j}} ≥ h_r then
7  |  |  //Apply backward elimination variable selection on valid LFAR
   |  |  (Section 4.1.2.2)
8  |  |  LFAR' ← backwardElimination (LFAR_{i,j})
9  |  end
10 |  if LFAR' unique then store in F
11 end
12 ϑ_{r_i} ← max(R^2_{r_{i,j}}); j = 1, 2, ..., n_{r_i}
```

The search process for the LFARs has a unique characteristic. The valid LFARs,

as defined by their predictive accuracies (R^2) being beyond a threshold (h_r) (Algorithm 3, Line 6), are recorded during the evolutionary process in a set (**F**). Although a chromosome including a valid LFAR can be replaced by its offspring according to the evolutionary principle, as different LFARs can represent different relations, simply replacing a parent chromosome during evolution can cause a loss of valid FARs. Assume that each of two LFARs (LFAR_a and LFAR_b) in two chromosomes (r_a and r_b) has a predictive accuracy greater than h_r, and that LFAR_a is $f(x_1) \Rightarrow x_2$ and LFAR_b $f(x_3) \Rightarrow x_4$, with r_b the offspring of r_a. Then, a basic evolutionary process will discard r_a and replace it with r_b when entering the next generation. However, as LFAR_a represents a different relation from LFAR_b, discarding it could mean losing a valid linear associative relation. Therefore, it is necessary to record the valid LFARs encountered during the evolutionary process into a set (**F**) (Algorithm 3, Line 8).

4.1.2.2 Sequential Search Variable Selection

During the recording process, a LFAR identified as valid by the above fitness calculation may have independent variables that do not contribute significantly to its linear relation. According to Occams Razor or the principle of parsimony, a model should contain all that is necessary for its purpose but nothing more; for example, if a regression model with two independent variables is sufficient to explain its dependent variable, only these two independent variables should be used. Therefore, since the independent variables are already specified by the LFAR, each valid LFAR also goes through a variable selection process, which is performed by the backward elimination procedure [96].

This procedure begins with a regression equation and sequentially deletes the independent variables that do not significantly contribute to the relation. Suppose an identified LFAR is $f(x_1, x_2, x_3) \Rightarrow x_4$ with a predictive accuracy of R^2 greater than h_r for x_4. The variable selection procedure sequentially drops each independent variable to determine whether the rest of the independent variables can still predict the dependent variable with an accuracy beyond a certain threshold; for example, when the variable x_1 is being checked, the LFAR becomes $f(x_2, x_3) \Rightarrow x_4$. The

multiple regression is then used to re-estimate the LFAR regression parameters. If the re-calculated prediction accuracy is still greater than a slightly lower threshold ($h_r - \Delta a$) (Δa reduces the accuracy threshold for re-estimation), the discarded variable can be considered to not contribute significantly to the relations and be eliminated from the LFAR. However, if the re-evaluation result shows that the prediction accuracy drops below the threshold ($h_r - \Delta a$) after removing one variable, that variable is put back in the LFAR and the backward elimination process moves to the next variable.

4.1.3 Evolutionary Algorithm based Linear Functional Association Rule Mining Strategies

In this thesis, the LFARM problem is considered a search problem. The feature of such a search space as well as the advantages of applying EAs were discussed in Chapter 3. Three different EAs, GA, PBIL and DE, are adapted as novel approaches to solve the LFARM problem, as discussed in the following three respective sections. These algorithms are selected due to the different characteristics of their evolutionary processes. DE maintains a parallel temporary population during its evolution, which also takes part in the evaluation. This feature should give it the capability to find more LFARs than other algorithms using a single population. PBIL uses a probability vector to generate its population, which it adjusts to move towards the direction of the current best individual. Although its evolutionary operators are relatively simple, as constantly shifting its population towards only the best individual could constrain its coverage of the search space, it is expected to find fewer LFARs. The GA is used as a canonical EA to be compared with. These three algorithms are compared using two proposed metrics to determine the differences in their performances.

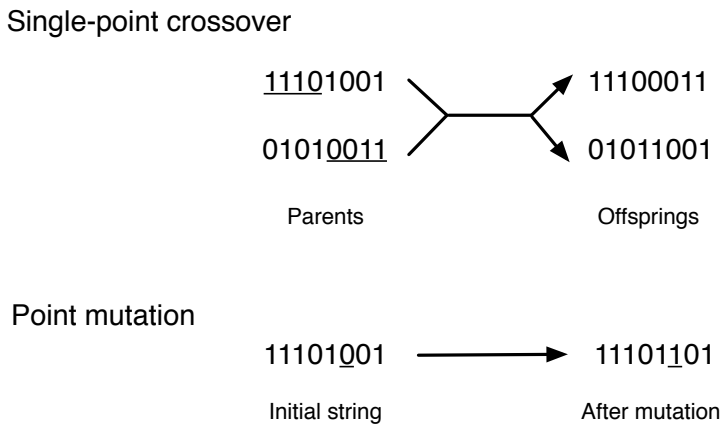


Figure 4.2: Single-point crossover operator and mutation operator

4.1.3.1 Genetic Algorithm based Linear Functional Association Rule Mining

The evolution in a GA is determined by a set of operators that recombine and mutate selected members of the current population. This GA-LFARM approach uses a single point crossover operator. It creates two offspring by exchanging a certain amount of genes between two parents specified by a gene position (as in Algorithm 7). Then, the mutation operator produces small random changes to the bit string by choosing a single bit at random and changing its value (as in Algorithm 6). These two operators are visualised in Figure 4.2. Roulette wheel selection is applied to select the parents for the generation of offspring [66] (as in Algorithm 5). Details of the implementation of the GA-LFARM algorithm are shown in Algorithm 4. The functions $rnd(lower, upper)$ and $rand(lower, upper)$ return respective integer and real values sampled from $[lower, upper]$ using a uniform distribution, where '$lower$' and '$upper$' refer to the lower and upper bounds of a range respectively. These two functions are also used in the other two EA-based LFARM approaches.

Algorithm 4: Genetic algorithm-based linear functional association rule mining: *GA-LFARM*(n_p, $\mathbf{X}$, n, R_m, R_c, h_r)

Input : population size (n_p), observational data ($\mathbf{X}$), number of observation variables (n), mutation rate (R_m), crossover rate (R_c), accuracy threshold (h_r)

Output: valid LFARs recorded ($\mathbf{F}$)

Create initial population: $R_p = \{r_1, r_2, ..., r_{n_p}\}$. **do**
  //Evaluate fitness of each chromosome and record valid LFARs
  for $i \leftarrow 1$ **to** n_p **do**
    $\vartheta_{r_i} \leftarrow regressionEvaluation(r_i, \mathbf{X}, h_r)$
  end
  $\Theta_p = \{\vartheta_{r_1}, \vartheta_{r_2}, ..., \vartheta_{r_{n_p}}\}$

  //Attach probability (p_i) to each chromosome individual for roulette wheel selection
  for $i \leftarrow 1$ **to** n_p **do**
    $p_i \leftarrow \frac{\theta_{r_i}}{\sum_{i=1}^{n_p} \vartheta_{r_i}}$
  end

  //Create temporary offspring population (supposing population size even number):
  for $i \leftarrow 1$ **to** $n_p/2$ **do**
    $parent^1 \leftarrow Selection(\Theta_p, R_p, n_p)$
    $parent^2 \leftarrow Selection(\Theta_p, R_p, n_p)$
    $\{child^1, child^2\} \leftarrow Crossover(parent^1, parent^2, R_c, R_m, n)$

    //Add two children into temporary population ($R'_p = \{r'_1, r'_2, ..., r'_{n_p}\}$):
    $r'_{2i} = child^1, r'_{2i+1} = child^2$
  end

  //Replace current population with temporary population
  for $i \leftarrow 1$ **to** n_p **do**
    $r_i = r'_i$
  end
while *Termination criteria is not met*;
access LFAR archive ($\mathbf{F}$), and return LFARs stored

Algorithm 5: Selection operator implemented in *GA-LFARM*: $Selection(R_p, n, \Theta_p)$

Input : population ($R_p = \{r_1, r_2, ..., r_{n_p}\}$), number of variables ($n$), set of fitness of each individual chromosome ($\Theta_p = \{\vartheta_{r_1}, \vartheta_{r_2}, ..., \vartheta_{r_{n_p}}\}$)
Output: Selected parent (*parent*)

$s = \sum_{i=1}^{n_p} \vartheta_{r_i}$
$selection_{sum} = s \times rand(0, 1)$
$partial_{sum} \leftarrow 0$
$j \leftarrow 0$
repeat
  $j \leftarrow j + 1$
  $partial_{sum} = partial_{sum} + \vartheta_{r_i}$
until $partial_{sum} \geq selection_{sum}$ *or* $j == n_p$;
Select jth individual for mating pool: $parent \leftarrow r_j$
Return *parent*

Algorithm 6: Mutation operator implemented in *GA-LFARM*: $Mutation(gene, R_m)$

Input : binary bit (*gene*), mutation rate (R_m)
Output: *gene*

if $rand(0, 1) < R_m$ **then** $gene \leftarrow \neg gene$
Return *gene*

Algorithm 7: Crossover operator implemented in *GA-LFARM*: $Crossover(parent^1, parent^2, R_c, R_m, n)$

Input : $Parent^1$, $Parent^2$, number of variables (n), crossover rate (R_c), mutation rate (R_m)
Output: $child^1, child^2$

if $rand(0,1) < R_c$ **then**
  $cross_{site} \leftarrow rnd(1, n)$
else
  $cross_{site} \leftarrow n$
end

$//i$ refers to gene position in chromosome
for $i \leftarrow 1$ **to** $cross_{site}$ **do**
  $child_i^1 \leftarrow Mutation(parent_i^1, R_m)$
  $child_i^2 \leftarrow Mutation(parent_i^2, R_m)$
end

$//i$ refers to gene position in chromosome
for $i \leftarrow cross_{site} + 1$ **to** n **do**
  $child_i^1 \leftarrow Mutation(parent_i^2, R_m)$
  $child_i^2 \leftarrow Mutation(parent_i^1, R_m)$
end
Return $child^1$, $child^2$

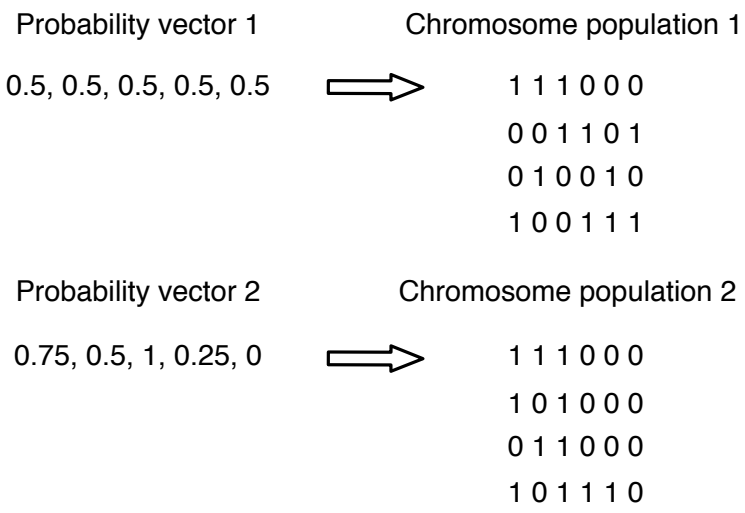


Figure 4.3: Examples of using probability vector (P) to generate population

4.1.3.2 Population based Incremental Learning for Linear Functional Association Rule Mining

PBIL belongs to the family of estimation of distribution algorithms (EDA) [17]. Its distinctive evolutionary process is that it maintains a probability vector (P) to evolve its populations. In a binary encoding chromosome, P specifies the probability of each bit position containing a value of 1 while the probability of that bit position containing a 0 can be derived by subtracting the probability specified in the vector from 1. This population generation process is visualised in Figure 4.3. The evolutionary process of PBIL adjusts P towards the best-fit individual in each generation, the population of which tends to be scattered around the regions represented by P. Then, the population is shifted towards the direction of the best individual in each generation through the updating of P. The probability update rule in Equation 4.3 shows that each element in the probability vector shifts towards one specific individual. This feature could cause the PBIL to converge faster but could also constrain its exploration of the search space to some extent. As a consequence, it is expected that it will identify fewer LFARs for the LFARM problem.

$$P_i = P_i \times (1.0 - l_r) + (l_r \times r^i_{best}) \tag{4.3}$$

r^i_{best} is the ith position in the best performing individual in the current population, towards which P moves and l_r a learning rate.

Unlike GA, most of the evolutionary operators of PBIL are not defined on the population but mainly occur directly on the probability vector, the adjustment of which corresponds to the selection and crossover operators in a GA. Regarding the mutation operator, there are two ways of defining it. The first is to perform a mutation on the probability vector, which is defined as a small probability of perturbation on each of the positions in it. The second is to perform a mutation on individuals in the current population, as conducted by the mutation operator used in a GA. In this thesis, the first mutation operator is used (Algorithm 8, Lines 15-18), with details of the implementation of the PBIL-LFARM approach shown in Algorithm 8.

Algorithm 8: Population-based incremental learning for linear functional association rule mining: $PBIL\text{-}LFARM(n_p, \mathbf{X}, n, R_s, R_m, h_r)$

Input : population size (n_p), observational data ($\mathbf{X}$), number of variables (n), mutation shift (R_s), mutation rate (R_m), accuracy threshold (h_r)

Output: valid LFARs recorded $\mathbf{F}$

1 $P \leftarrow$ initialised vector ($P_i \leftarrow 0.5$, $i = 1, 2, ..., n$) **do**
2 //Create the current population $i \leftarrow 1$
3 **while** $i \leq n_p$ **do**
4 Create chromosome (r_i) by sampling each probability value in P $\vartheta_{r_i} \leftarrow regressionEvaluation(r_i, \mathbf{X}, h_r)$ //Algorithm 3 $i \leftarrow i + 1$
5 **end**
6 Select best individual (r_{best}), $best = \arg\max_i \vartheta_{r_i}$ Update P towards r_{best} using Equation 4.3
7 //Apply mutation operator to P $i \leftarrow 1$ **while** $i \leq n$ **do**
8 $m \leftarrow rand(0, 1)$ **if** $m < R_m$ **then** $P_i \leftarrow P_i \times (1 - R_s) + rnd(0, 1) \times R_s$ $i \leftarrow i + 1$
9 **end**
10 **while** *Termination criteria not met*;
11 access LFARs recorded and return set $\mathbf{F}$

4.1.3.3 Differential Evolution based Linear Functional Association Rule Mining

DE is an EA initially designed to solve real-value problems using a population of floating-point encoded individuals [190]. It generates its offspring by firstly forming an intermediate trial individual from a set of selected parents using a mutation operator. The mutation operator perturbs one of the parent individuals (called the main parent) with a weighted difference derived from the other parents. The main parent can be either a randomly selected individual or the best individual found so far. The following equation defines how the perturbation is implemented using three randomly selected individuals and is known as the DE/rand/1 scheme [8].

$$r^t = r_{pa_1} + F_s \times (r_{pa_2} - r_{pa_3}) \tag{4.4}$$

F_s is known as a scale factor and is a real number that controls the rate, at which the population evolves. While there is no upper limit on it, in practice, its effective values are seldom greater than 1.0 (in Algorithm 9, Line 13, a Gaussian random number is used to determine its value, as adapted from [1]). r^t refers to the trial individual, and $r_{pa_1}, r_{pa_2}, r_{pa_3}$ are randomly selected individuals from the current population. There are also other variants of this perturbation procedure; for example, DE/best/1, where the main parent is the best-performing individual found so far and DE/rand-to-best/1, where perturbation is achieved from a pair of differences derived from four different individuals [8].

The trial individual (r^t) then goes through a crossover operator process by being mixed with another randomly selected parent individual (r_{pa}) as:

$$r^t_{,j} = \begin{cases} r_{pa,j} & rand(0,1) < R_c \text{ or } j == rand(1,n) \\ r^t_{,j} & otherwise \end{cases} \quad j = 1, 2, ..., n; \tag{4.5}$$

the index j is used to refer the genes in the individual chromosome. R_c is crossover

rate (Algorithm 9 Line 12-17). The function *rand* is defined in Section 4.1.3.1. Since the chromosomes used in the evolution is binary bit, the trial vector is then converted into binary vector (Algorithm 9 Line 14) as follows:

where the index j refers to the genes in the individual chromosome, R_c is the crossover rate (Algorithm 9, Lines 12-17) and the function 'rand' is defined in Section 4.1.3.1. Since the chromosomes used in the evolution are binary bits, the trial vector is then converted into a binary vector (Algorithm 9, Line 14) by:

$$r^t_{,j} = \begin{cases} 1 & if\ r^t_{,j} \geq 0.5 \\ 0 & otherwise \end{cases} \quad j = 1, 2, ..., n; \tag{4.6}$$

The offspring is eventually created by a selection operator, which compares the fitness values of r_{pa} and the trial individual (r^t) to determine which should proceed to the next generation (Algorithm 9, Lines 23-29) as:

$$r^{offspring} = \begin{cases} r^t & \vartheta_{r^t} \geq \vartheta_{r_{pa}}; \\ r_{pa} & otherwise; \end{cases} \tag{4.7}$$

It can be seen in the above selection process that this algorithm evaluates more individuals than the other two as it considers those in both the main and trial populations. Although some individuals in the trial population may not participate in evolution due to the selection operator, since they go through the evaluation process, any valid LFARs will be extracted. Due to these extra evaluations, DE-LFARM is expected to extract more FARs than those which evaluate only one population, and its implementation is shown in Algorithm 9.

Algorithm 9: Differential evolutionary algorithm-based linear functional association rule mining: *DE-LFARM*(n_p, $\mathbf{X}$, n, R_m, R_c, h_r)

Input : population size (n_p), observational data ($\mathbf{X}$), number of observation variables (n), mutation rate (R_m), crossover rate (R_c), accuracy threshold (h_r)

Output: valid LFARs recorded ($\mathbf{F}$)

Create initial population of LFARs: $R_p = \{r_1, r_2, ..., r_{n_p}\}$
do
  for $i \leftarrow 1$ **to** n_p **do**
    $\vartheta_{r_i} \leftarrow regressionEvaluation(r_i, \mathbf{X}, h_r)$ //Algorithm 3
  end

  //Generate trial population
  for $i \leftarrow 1$ **to** n_p **do**
    Select individual at random as main parent (r_{pa_1}) and two other individuals as supporting parents (r_{pa_2} and r_{pa_3})
    $j_{rand} = rand(1, n)$
    for $j \leftarrow 1$ **to** n **do**
      if $rand(0,1) < R_c$ *or* $j_{rand} == j$ **then**
        $r^t_{i,j} \leftarrow r_{pa_1,j} + Gaussian(0,1) \times (r_{pa_2,j} - r_{pa_3,j})$
        Convert $r^t_{i,j}$ into binary form
      else
        $r^t_{i,j} \leftarrow r_{pa_1,j}$
      end
    end
  end

  //Create new population:
  for $i \leftarrow 1$ **to** n_p **do**
    $\vartheta_{r^t_i} \leftarrow regressionEvaluation(r^t_i, \mathbf{X}, h_r)$
    if $(\vartheta_{r^t_i} > \vartheta_{r_i})$ **then**
      $r_i \leftarrow r^t_i$
    else
      $r_i \leftarrow r_i$
    end
  end
while *Termination criteria not met*;
access LFARs recorded and return the set $\mathbf{F}$

4.2 Performance Metrics for Linear Functional Association Rule Analysis

To analyse the LFAR results, one basic target is to uncover as many valid and unique LFARs as possible. However, it is not the completeness of the underlying rules that is of specific interest. Since the downward closure property [7] does not hold for LFARs, it is not expected that a complete set of LFARs will be generated. Therefore, one measure of performance is the number of rules found by different algorithms for the same experimental settings. This metric characterises the extent of the natural coverage of the search space obtained by different algorithms.

Other than the numbers of LFARs obtained, a particular point of interest in the experiments is their characteristics derived from different algorithms, which is achieved using the two metrics of complexity and perceptual selectivity.

4.2.1 Complexity

One aspect regarding analysing LFARs is complexity and, according to the principle of parsimony, which requires a model to be precise and simple, this metric examines the qualities of the LFARs found. Complexity evaluates LFARs by measuring the numbers of LHS variables for each RHS variable by:

$$Complexity_{x_i} = \frac{\sum_{k=1}^{N_{x_i}} J_k}{N_{x_i}} \tag{4.8}$$

where x_i refers to a variable that appears as a RHS variable in a LFAR, N_{x_i} the number of LFARs found with x_i and J_k the number of LHS variables in the kth LFAR of the LFARs with x_i as the RHS variable. This metric characterises the qualities of the LFARs found by different algorithms from the perspective of the parsimony principle.

4.2.2 Perceptual Selectivity

Another aspect of the performances of the different algorithms is the dynamics of their searching procedures; specifically, whether they behave similarly when constructing LFARs. Such behaviour can be characterised by perceptual selectivity, which is a term used in artificial agent design initially for the purpose of reducing computational complexity [58]. It characterises an agent's capability to limit the set of sensory data to be attended at any one time. Similarly, in LFARM, perceptive selectivity can be used to capture the dynamic choices of variables, which form LFARs and is defined as a frequency measure by:

$$PerceptualSelectivity_{(x_j,x_i)} = \frac{N_{(x_j,x_i)}}{N_{x_i}} \tag{4.9}$$

where $PerceptualSelectivity_{(x_j,x_i)}$ is the frequency of x_j appearing as an LHS variable in LFARs where x_i acts as the RHS variable ($i \neq j$), $N_{(x_j,x_i)}$ the number of valid LFARs found so far with x_i the RHS variable and x_j the LHS variable at the same time, and N_{x_i} the number of LFARs found so far with x_i as the RHS variable. As this perceptual selectivity metric can be applied during the evolutionary process, it can visualise the dynamics of different algorithms when constructing FARs during their search processes.

4.3 Experiments on Linear Functional Association Rule Mining

The proposed methods and metrics are applied to four real-world datasets downloaded from the UC Irvine machine learning repository (UCI) [14] (1-12) and function approximation repository [**?**], that is, (1) breast cancer Wisconsin prognostic (BCW prognostic), (2) breast cancer Wisconsin diagnostic (BCW diagnostic), (3) breast cancer Wisconsin original (BCW original) (4) concrete (slump), (5) concrete (strength), (6) dermatology, (7) fertility, (8) housing (Housing), (9) sonar, (10)

Table 4.2: Data set summary [14]

Data ($\mathbf{X}$)	No. variables (n)	No. instances (p)
1 BCW prognostic	34	194
2 BCW diagnostic	31	569
3 BCW original	10	599
4 Concrete (Slump)	10	103
5 Concrete (strength)	9	1030
6 Dermatology	35	358
7 Fertility	10	100
8 Housing	14	506
9 Sonar	61	208
10 Stockprice	10	506
11 Wine	14	198
12 Yacht	7	308
13 Baskball	5	96
14 Body fat	18	252
15 Bolts	8	40
16 Pollution	16	60
17 Quake	4	2178
18 Sleep	8	51
19 Vine	4	52
20 Iris	5	150

stockprice (11) wine (Wine) (12) yacht, (13) baskball, (14) bodyfat, (15) bolts, (16) pollution, (17) quake, (18) sleep, (19) vine, and (20) iris. Instances with missing values are deleted from the original datasets, and a summary of the experimental datasets is given in Table 4.2. In these datasets, the variables representing the participant ID are removed. The treatment of missing data is done assuming that domain knowledge about the given dataset is unavailable. In particular, whether the missing data corresponds to dependent variables or independent variables is assumed to be unknown. Due to this assumption, the instances with missing data are

Table 4.3: Experiment parameters

Parameters	GA-LFARM	PBIL-LFARM	DE-LFARM
Population size (n_p)	100	100	100
Generation size (n_g)	100	100	100
Crossover rate (R_c)	0.7	0.7	0.7
Mutation shift (R_s)	n/a	0.02	n/a
Mutation rate (R_m)	0.02	n/a	n/a
R^2 threshold (h_r)	0.9	0.9	0.9

simply removed. Some of the datasets are discrete (e.g. BCW original), while others are continuous (e.g. sonar). In this chapter, the basic idea is to use regression to identify and represent hidden relations. Therefore, both discrete and continuous datasets can be handled using our FARM algorithm. Furthermore, continuous datasets need not be discretised (unlike in traditional methods).

It should be noted that the original data mining tasks of these datasets are not important in this chapter as we ignore the default tasks and use these datasets assuming no *a priori* knowledge of them. The parameters used in the experiments are summarised in Table 4.3, and each approach is run 30 times with different seeds.

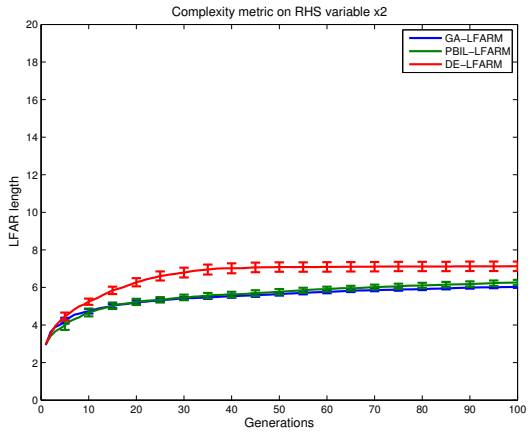
Table 4.4 shows the overall numbers of LFARs found by each approach. For each approach, these LFARs are stored in the set $\mathbf{F}$, in which each member is unique and has the predictive accuracy on the given dataset greater than h_r. In most datasets that returns LFARs, DE-LFARM finds more rules than the other two approaches (1, 2, 4, 6, 10, 11, 18). DE-LFARM is expected to return the most LFARs due to that it maintains a parallel trial population during its evolution. There are datasets, where three approaches converge to similar rules (e.g. dataset 4, 5, 11, 15, 16, 18). This can be attribute to that all possible linear relations in the data have been explored. Notably, there are datasets in which none of the three methods return any rules (e.g. 7, 8, 13, 17). It is known that there are hidden relations in the datasets, as these datasets have been commonly used for data mining tasks set

by domain experts. This result thus implies that assuming only linear relations is not sufficient for identifying complex hidden relations in the given datasets. This problem is further discussed in Chapter 5.

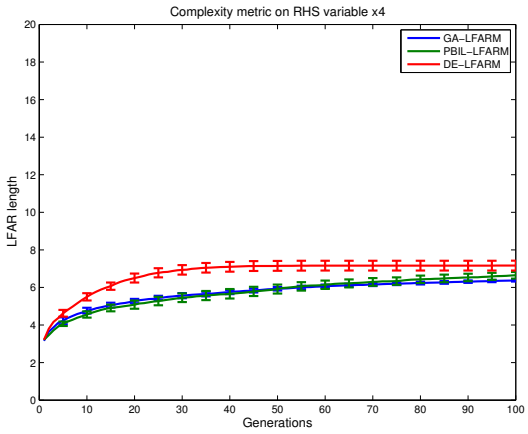
In Table 4.5, it can be seen that the PBIL-LFARM algorithm uses the least computational time to complete its evolution. Although DE-LFARM in general returns more LFARs, it takes significantly longer to process the data while, GA-LFARM performs moderately in comparison with the other two in terms of run time.

The two proposed metrics are also used to evaluate the performances of the three mining approaches. Figures 4.4 and 4.5 present the complexities of different LFARs using different RHS variables from the BCW prognostic dataset. The 95% confidence intervals are calculated and shown in order to compare the performance of the three algorithms. In general, the complexities of the LFARs identified by the three approaches gradually increase over generations, except for those variables that do not have their own LFARs (as in Figure 4.4 (c)(e)(f) and Figure 4.5(e)). In six of the eight plots, PBIL-LFARM has significantly longer LFAR lengths than the other two approaches. Of the remaining two plots, one shows that PBIL-LFARM has a shorter LFAR length than the other two approaches while the other indicates no evident difference. For GA-LFARM and DE-LFARM, except in Figure 4.5(f), there is no evidence of different performances measured by the complexity metric. Overall, there is no consistent evidence indicating significant differences in the complexity performances of the three approaches.

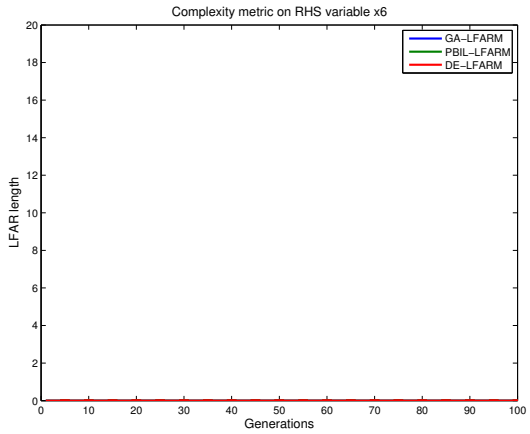
Figure 4.6 and 4.7 plots the perceptual selectivity metric analysis results for the three approaches using different datasets. This analysis is also executed along the generation dimension and we calculate the perceptual selectivities of different LHS variables and RHS variables in the LFARs. In Figure 4.6 (a) and (b), at the end of the evolution, PBIL-LFARM shows higher perceptual selectivity values for variables x_5 and x_6 but, in Figure 4.6 (f), a lower perceptual selectivity value for variable x_{10}. In the other plots, the differences are statistically ambiguous. In general, there is no consistent evidence indicating performance differences in terms of perceptual



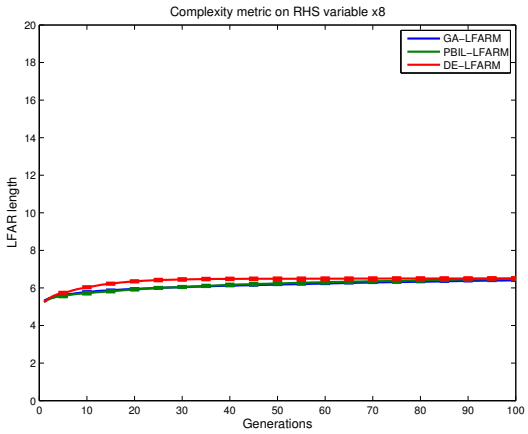
(a) complexity analysis of RHS x_2



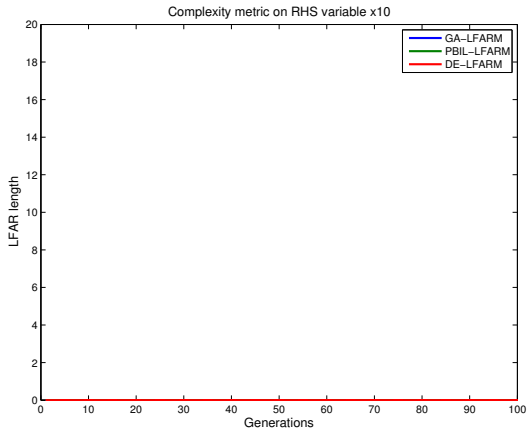
(b) complexity analysis of RHS x_4



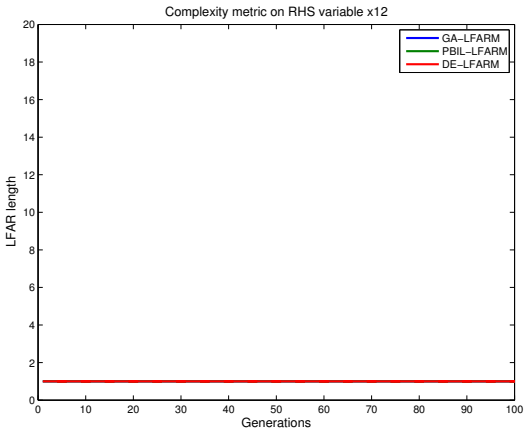
(c) complexity analysis of RHS x_6



(d) complexity analysis of RHS x_8

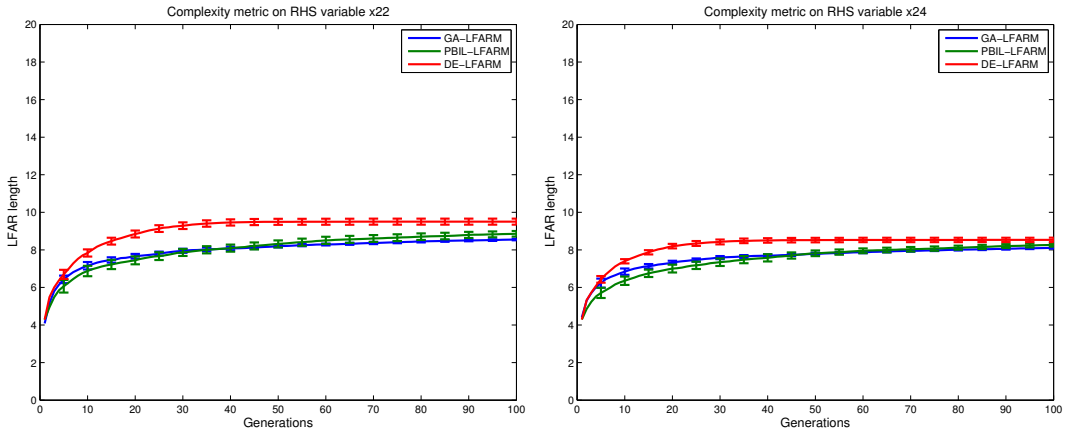


(e) complexity analysis of RHS x_{10}

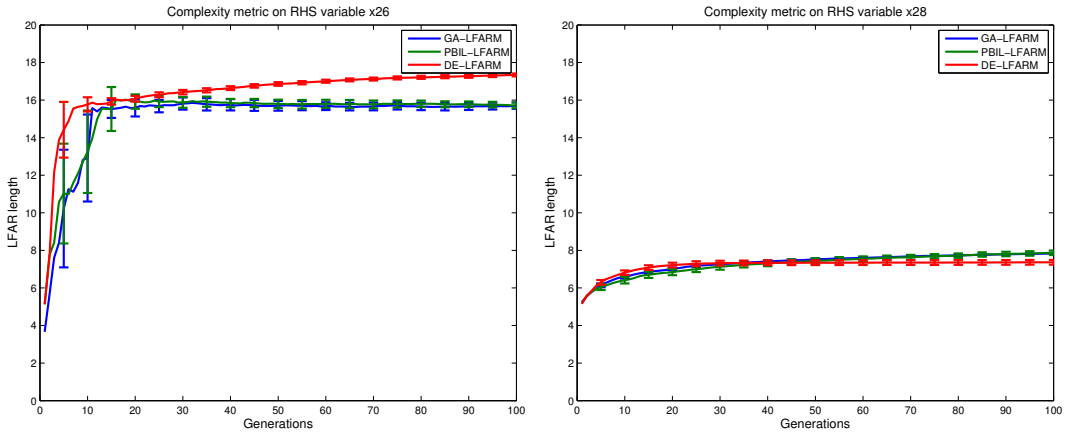


(f) complexity analysis of RHS x_{12}

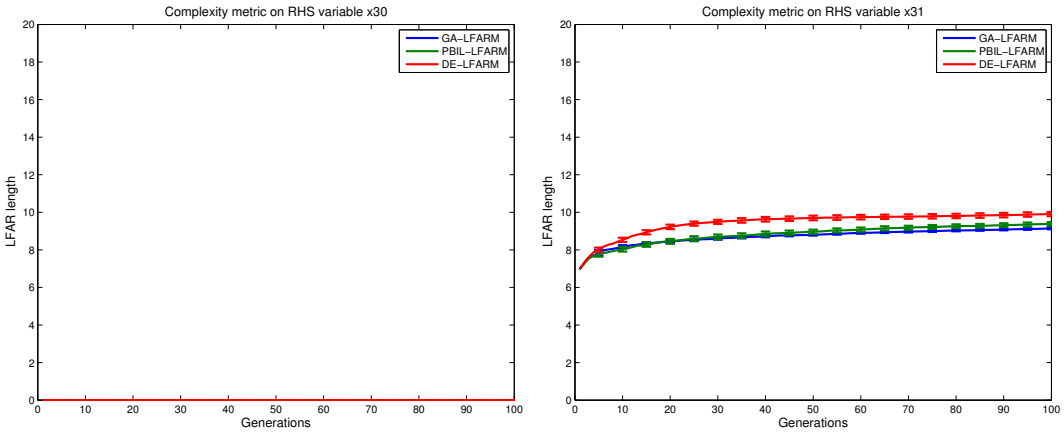
Figure 4.4: Average complexities of LFARs featured by different RHS variables: x_2, x_4, x_6, x_8, x_{10}, x_{12} (BCW prognostic)



(a) complexity analysis of RHS x_{22} (b) complexity analysis of RHS x_{24}



(c) complexity analysis of RHS x_{26} (d) complexity analysis of RHS x_{28}



(e) complexity analysis of RHS x_{30} (f) complexity analysis of RHS x_{31}

Figure 4.5: Average complexities of LFARs featured by different RHS variables: x_{22}, x_{24}, x_{26}, x_{28}, x_{30}, x_{31} (BCW prognostic)

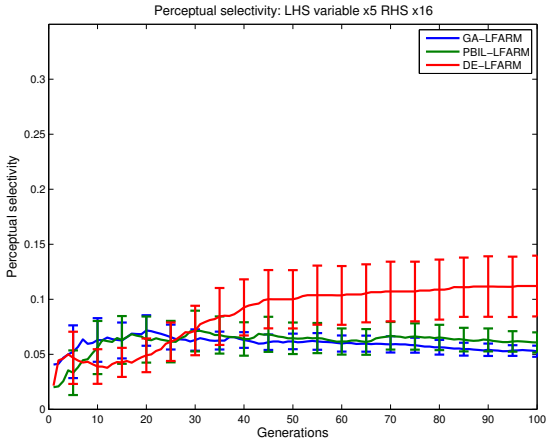
selectivity among the three approaches.

In summary, the complexity metric reflects the quality of the LFARs found from the perspective of the principle of parsimony. The experimental results do not provide any evidence of differences among the three algorithms. From the results for the other metric, perceptual selectivity, which reflects the dynamic aspects of the different approaches for constructing LFARs during their evolutions, the same conclusion can be drawn, that is, there are no consistent significant differences among the three approaches. Regarding the number of rules found by each algorithm, DE-LFARM can find most LFARs but at a higher computational cost. Although PBIL-LFARM has the fastest processing speed, it returns the fewest rules of the three while GA-LFARM performs moderately in terms of both processing speed and number of rules found. The experiment on the Housing dataset returns no LFARs while the default task suggests there is a certain relation(s) hidden in the dataset, which indicates the need to increase the representational power of the FARs.

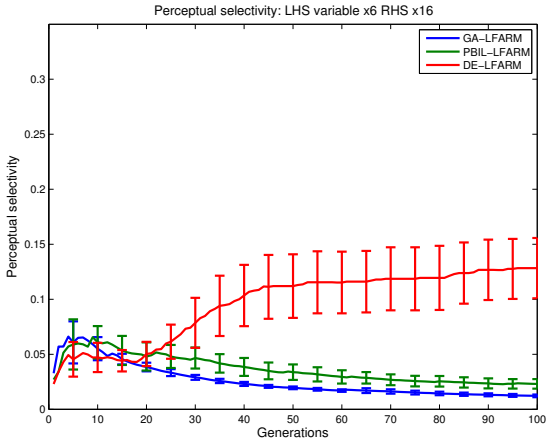
4.4 Chapter Summary

In this chapter, details of three EA-based LFARM approaches for solving the AHGP were presented. Both quantitative and qualitative comparisons of them were conducted through performing experiments on different datasets. PBIL-LFARM took the least amount of time for its evolutionary process but found significantly fewer LFARs using the same parameter settings as the other two approaches. DE-LFARM found the most LFARs but at a higher computational cost while GA-LFARM performed moderately in terms of both computational time and the number of LFARs found.

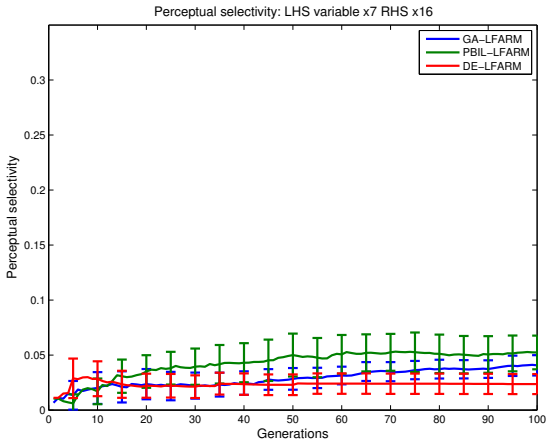
Applications of the two metrics, complexity and perceptual selectivity, showed that the performances of all three algorithms were not significantly different. Therefore, any of them could be selected depending on the computational time available and number of LFARs required. DE-LFARM could be selected when the target is to find the most LFARs and computational cost is not the main concern, while



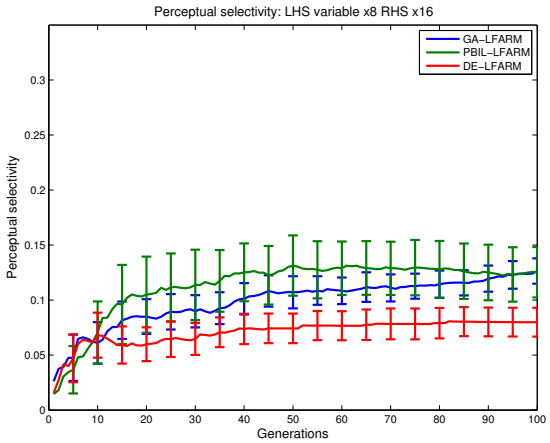
(a) perceptual selectivity of LHS x_5, with RHS x_{16}, in each generation



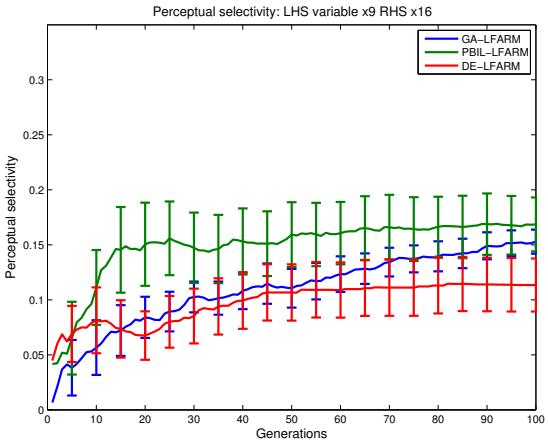
(b) perceptual selectivity of LHS x_6, with RHS x_{16}, in each generation



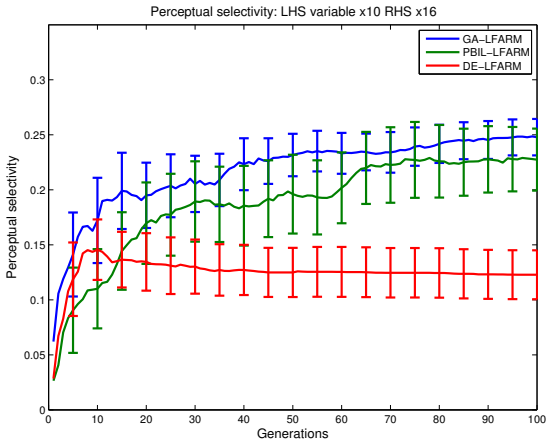
(c) perceptual selectivity of LHS x_7, with RHS x_{16}, in each generation



(d) perceptual selectivity of LHS x_8, with RHS x_{16}, in each generation

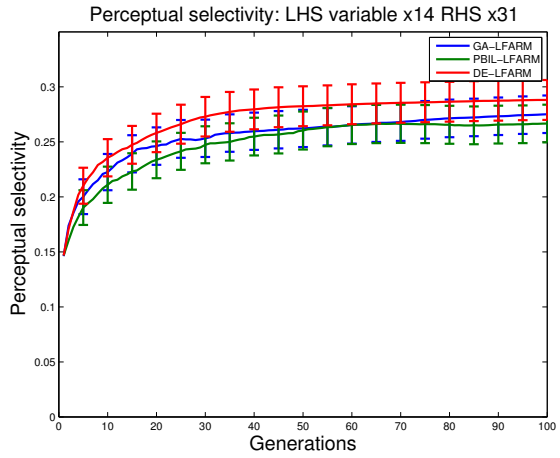


(e) perceptual selectivity of LHS x_9, with RHS x_{16}, in each generation

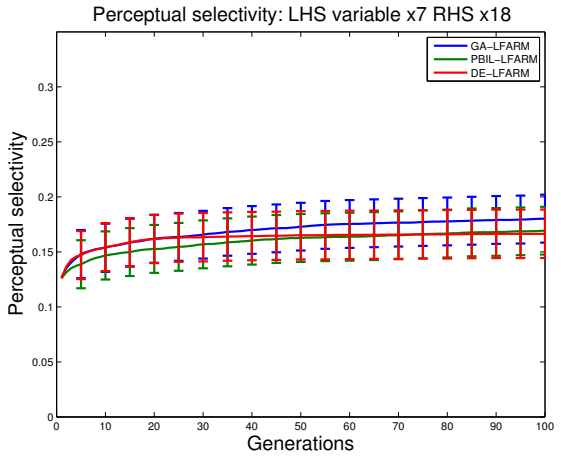


(f) perceptual selectivity of LHS x_{10}, with RHS x_{16}, in each generation

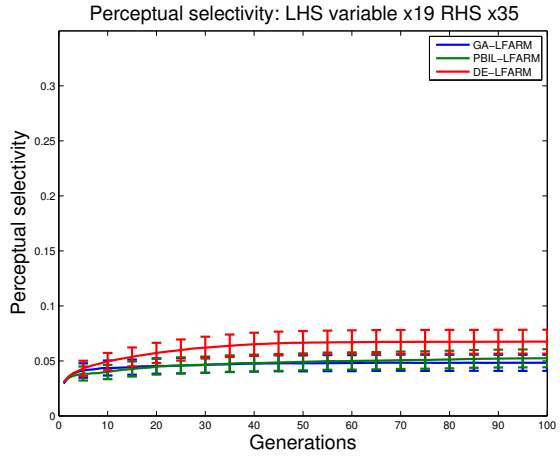
Figure 4.6: Different LHS variables measured by perceptual selectivity metric for LFARs with x_{16} as RHS variable (BCW diagnostic)



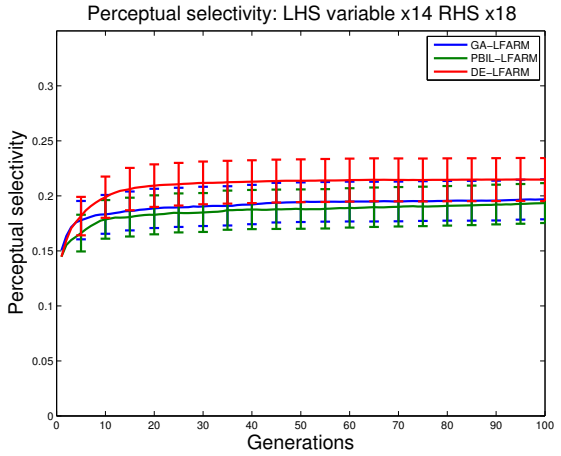
(a) perceptual selectivity of LHS x_{14}, with RHS x_{31}, BCW diagnostic



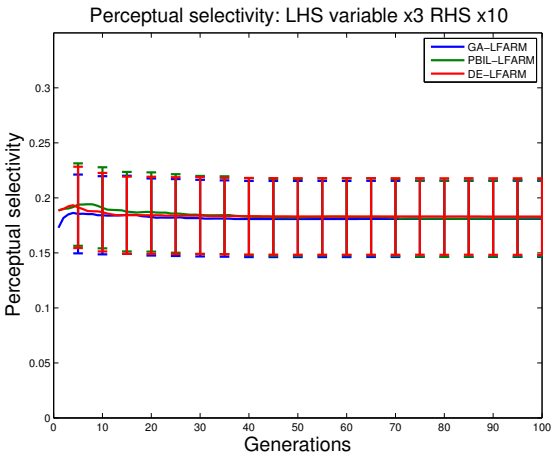
(b) perceptual selectivity of LHS x_7, with RHS x_{18}, Body fat



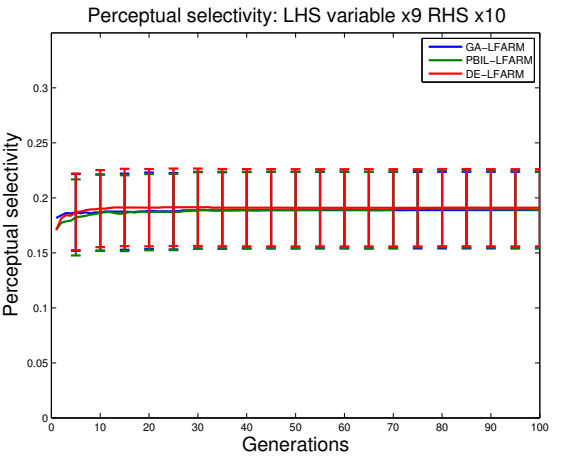
(c) perceptual selectivity of LHS x_{19}, with RHS x_{35}, Dermatology



(d) perceptual selectivity of LHS x_{14}, with RHS x_{18}, Body fat



(e) perceptual selectivity of LHS x_3, with RHS x_{10}, Stock price



(f) perceptual selectivity of LHS x_9, with RHS x_{10}, Stock price

Figure 4.7: Different LHS variables measured by perceptual selectivity metric for LFARs in different datasets

PBIL-LFARM has an advantage regarding computational time.

The experiments also exposed the disadvantage of LFARs for extracting complex predictive relations from data. Those on the Housing dataset did not return any LFARs although we know that there are certain associations in the data. Firstly, this emphasizes the need to investigate general representations of associative relations for the AHGP. In this chapter, we only searched for linear relations hidden in the data using LFARs. A general associative hypothesis representation and its specific search scheme are discussed in the next chapter. Secondly, there are other factors that potentially affect the performance of the LFARM algorithm, such as epistasis. In an underlying relation, some variables may belong to the same building block. During the evolutionary process, such building blocks should be preserved (not disturbed by the evolution operators) for valid relations to emerge. Our current algorithm design has not taken such a situation into consideration. However, for future work, this is a potential direction to pursue for improving the performance of LFARM algorithms.

Table 4.4: Average numbers of unique LFARs over 30 runs with 95% confidence interval

Data	GA-LFARM	PBIL-LFARM	DE-LFARM
1 BCW prognostic	6667.5±190.3	3658.9±100.3	10350.0±71.6
2 BCW diagnostic	10036.0±200.7	6361.9±158.6	15223.0±80.2
3 BCW original	0.0±0.0	0.0±0.0	0.0±0.0
4 Concrete (Slump)	41.7±0.2	41.2±0.2	42.0±0.0
5 Concrete (strength)	0.0±0.0	0.0±0.0	0.0±0.0
6 Dermatology	1192.0±51.9	1068.5±62.9	1813.5±24.7
7 Fertility	0.0±0.0	0.0±0.0	0.0±0.0
8 Housing	0.0±0.0	0.0±0.0	0.0±0.0
9 Sonar	14341.8±346.3	24865.9±421.4	20823.9±148.7
10 Stockprice	38.4±0.2	36.4±0.5	39.0±0.0
11 Wine	0.9±0.1	1.0±0.0	0.7±0.2
12 Yacht	3.0±0.0	3.0±0.0	3.0±0.0
13 Baskball	0.0±0.0	0.0±0.0	0.0±0.0
14 Body fat	1114.3±21.6	660.0±10.2	1433.0±7.7
15 Bolts	1.0±0.0	1.0±0.0	1.0±0.0
16 Pollution	2.0±0.0	2.0±0.0	2.0±0.0
17 Quake	0.0±0.0	0.0±0.0	0.0±0.0
18 Sleep	13.0±0.1	12.5±0.2	60.9±0.3
19 Vine	0.0±0.0	0.0±0.0	0.0±0.0
20 Iris	6.0±0.0	6.0±0.0	6.0±0.0

Table 4.5: Average run times (seconds) of three algorithms with 95% confidence interval

Data	GA-LFARM	PBIL-LFARM	DE-LFARM
1 BCW diagnostic	1627.4±46.1	1173.1±31.4	3918.8±48.6
2 BCW prognostic	1654.7±33.4	1192.8±24.4	4052.7±50.3
3 BCW original	29.2±0.4	15.6±0.3	63.0±0.3
4 Concrete (Slump)	41.0±1.3	41.6±0.4	60.5±0.3
5 Concrete (strength)	30.1±0.7	21.0±0.7	72.8±0.8
6 Dermatology	289.2±6.7	391.6±12.5	563.6±3.3
7 Fertility	14.5±0.2	10.2±0.2	44.3±0.2
8 Housing	33.9±0.5	17.5±0.2	78.7±0.4
9 Sonar	1935.1±54.3	4981.3±78.5	5427.6±140.5
10 Stockprice	52.6±1.1	69.5±0.3	97.9±0.4
11 Wine	25.2±0.3	27.9±0.2	63.6±0.2
12 Yacht	26.0±0.6	28.0±0.1	47.1±0.9
13 Baskball	8.4±0.2	6.6±0.2	20.8±0.3
14 Body fat	278.0±3.6	192.0±1.7	525.7±2.9
15 Bolts	14.3±1.6	25.2±0.2	38.0±0.2
16 Pollution	51.7±0.7	54.6±0.5	101.9±0.4
17 Quake	21.3±0.5	12.0±0.1	31.6±1.6
18 Sleep	35.8±0.4	32.5±0.2	13.0±0.0
19 Vine	7.3±0.1	6.0±0.1	16.1±0.1
20 Iris	24.2±0.4	21.3±0.1	42.4±0.1

Chapter 5

Associative Hypothesis Generation: General Functional Association Rule Mining

The last chapter presented three evolutionary algorithm (EA)-based linear functional association rule mining (LFARM) approaches as solutions to the associative hypothesis generation problem (AHGP), with two additional metrics, complexity and perceptual selectivity, used to analyse their performances. Although they showed different performances regarding the number of LFARs found and computational time, there was no evidence that they performed differently on the complexity and perceptual selectivity metrics. In addition, the experiments demonstrated the limitation of the LFAR representation as, although using linear functions as analytical expressions has the advantage of simplicity and interpretability, some complex hidden relations may not be captured.

In this chapter, a general FAR representation is presented, followed by a novel cooperative co-evolution based algorithm for mining FARs. Experiments on a set of synthetic and real-world datasets are conducted to assess the performance of the proposed algorithm for solving the AHGP. Besides providing inputs for the causal hypothesis generation problem (CHGP), a FAR is an alternative definition

for a general quantitative association rule. It can be implemented independently for quantitative ARM tasks and, to illustrate its performance on solely these tasks, we compare the proposed general FAR mining (FARM) algorithm with other closely related ones in terms of the metrics often used in ARM algorithms (e.g., rule size, predictive accuracy) and it is shown to be competitive. Also, in order to match the flexibility of the general association rule form, we extend the FAR representation to allow multiple RHS variables.

The remainder of this chapter is organised as follows: Section 5.1 outlines the design strategies for the FARM approach; Section 5.2 details implementation of the cooperative co-evolutionary FARM (CCFARM) algorithm; in Section 5.3, experiments on sets of synthetic and real-world datasets and a comparison with other EA-based quantitative ARM algorithms are presented; and conclusions are drawn in Section 5.4.

5.1 Outline of the Functional Association Rule Mining Approach Design

Like the LFAR, a FAR is an alternative form of an association rule but has fewer constraints on the underlying relations. Hereafter, for simplicity, FAR is used to refer to the general FAR and the AHGP defined in Chapter 3 is restated as follows. Suppose a given observational dataset ($\mathbf{X}$) has p instances, each of which is a vector of the form $X = [x_1, x_2, ..., x_i, ..., x_n]$ (n is the number of variables) and each x_i is a continuous variable. The target is to generate a set ($\mathbf{F} = \{f_1, f_2, ..., f_j, ..., f_m\}$) of m associative hypotheses, each of which (f_j) is represented in terms of the proposed FAR, which takes the form:

$$f(X_A) \Rightarrow X_B \tag{5.1}$$

where $X_A, X_B \subset X$ and $X_A \cap X_B = \emptyset$. This rule is interpreted as meaning that

the values of the variables in B can be predicted by the values of the variables in A (e.g., $f(x_1, x_2, x_3) \Rightarrow x_4$). The mining task is to generate as many valid FARs as possible from the given dataset. Considering a FARM algorithm as a potential solution to the AHGP, we focus mainly on its right-hand side (RHS) having one variable ($|X_B| = 1$). Then, as input to the causal hypothesis generation, a FAR with $|X_B| = 1$ can allow the experimental causal search algorithm (solution for the CHGP) to focus on testing one response variable for each FAR, as presented in the next chapter. However, FARs inherently allow the RHS to have multiple variables ($|X_B| \geq 1$), a representation discussed in Section 5.3.5.

Chapter 3 provided a preliminary discussion of the problems that need to be addressed when designing solutions for FARM, including the representation, evaluation and search strategy for FARs. This section presents more details about the design of solution strategies. As noted in Chapter 4, the LFAR has an inherently analytical expression (i.e., a regression model). However, a FAR does not have a general analytical expression as, although its general functional relation can take any form, e.g., polynomial, exponential or sigmoid, it is not applicable for enumerating all functional forms. Therefore, an intermediate representation of any function is needed to conduct the mining task. The evaluation strategy is to specify a metric to measure how well a FAR is supported by the data and whether it is a valid hypothesis. Also, the search strategy should be updated to accommodate changes in the FAR form and metric.

For the intermediate function, the artificial neural network (ANN) is chosen to represent the arbitrary relations among continuous variables. By using it, we can bypass the problem of specifying a particular mathematical form for mapping from the left-hand side (LHS) to RHS of a FAR because, if the FAR does represent a hidden relation, it has the potential to be uncovered by the ANN's training process. As reviewed in Chapter 2, an ANN is a universal function approximator. An ANN with two layers can approximate any bounded continuous functions with arbitrary small error [43] . An ANN with three layers (two hidden layers and one output layer) can approximate any function to an arbitrary precision with mild assumptions

about the activation function (activation function should be continuous and non-linear) [44], [87]. Although the universal approximation theorem does not specify how to determine the structure of such an ANN, in practice, for relatively simple underlying relations, an ANN with one hidden layer is used. For relatively complex relations, ANNs with two hidden layers are adopted as discussed in Section 2.4.1.3. If a trained ANN can predict unseen data (e.g. a reserved test set) with a certain degree of accuracy, it indicates that a FAR is identified.

A common concern when using an ANN as a predictive model is its lack of interpretability, which is not totally eliminated when using it for FARs. FARs can be perceived on two levels, with one being that several variables are related; for example, customers who buy milk, also buy bread. This interpretability is still preserved in the mining algorithm, which, in addition, describes the relationship itself for which we adopt an ANN. As, in the previous example, the rule does not necessarily say how much bread is being bought as a function of how much milk is being bought, the ANN adds a predictive layer on top of the association layer.

The search strategy designed in this chapter is also based on EAs. In order to find valid FARs, which have high predictive accuracies on their RHSs, we apply the genetic algorithm (GA). Each individual in its population is designed to represent one potential FAR. Predictive accuracy is incorporated in the objective function in order for the evolutionary process to favour FARs with high ones. As discussed above, the evaluation of FARs is conducted by an ANN approximation, which, in this algorithm, is carried out by backpropagation (BP). As a gradient-based technique, BP can become stuck in a local minimum and may miss a potential interesting FAR if the weight initialisation of an ANN is not properly assigned. This indicates that a complete solution for FARM requires a potential FAR individual and a relatively matched ANN, which forms a cooperative structure. Based on this feature, we introduce a cooperative co-evolutionary search strategy for the FAR mining problem.

For this mining problem, as we expect multiple solutions (multiple FARs) to exist in the observational data, we do not want the solution to converge to one best solution in the last generation of evolution. Therefore, the archival procedure

introduced in [193] is adapted in the proposed mining approach, by storing all the FARs identified during the search, which serves as an elitist mechanism. In addition, it is also used as a comparison set in fitness assignments for the purpose of increasing coverage of the search space of the FARs. The implementation of the CCFARM algorithm is explained in the following section.

5.2 Cooperative Co-evolutionary Functional Association Rule Mining Algorithm

A CCEA involves several co-existing populations that together form a solution to a given problem. As the fitness of each individual depends on its collaboration with individuals from other populations, evolutionary pressure favours cooperative individuals [151], [134], [152]. The CCFARM algorithm proposed in this chapter evolves two sub-populations, FAR and ANN. A valid solution for the mining task is comprised of a potentially interesting FAR and an appropriately initialised ANN for the FAR. After training, the ANN can predict the RHSs of FARs on unseen test data, which exceed a certain threshold (h_r).

The collaboration of the two sub-populations is a complete mixing whereby an individual in one sub-population is paired with every individual in the other to determine its fitness. Although this scheme is computationally expensive, it is recommended when cross-population epistases are expected in the sub-populations. In cooperative co-evolution, a cross-population epistasis refers to the genes in one sub-population having non-linear relations with those in other sub-populations and, when they evolve separately, affecting the performance of the co-evolutionary algorithm. As the FAR and ANN sub-populations are dependent on each other, this complete mixing scheme is adopted for the mining algorithm.

The external archive used in the proposed mining algorithm not only stores the valid FARs during the evolutionary process but also serves as a base to increase the coverage of the search in its search space. The complete solutions that exceed

the accuracy threshold (h_r) are selected for the archive so that individuals in the sub-populations can focus on regions that are not sufficiently explored. A measure of distance is added into the objective function to serve this purpose. A flowchart of the CCFARM algorithm is shown in Figure 5.1 and a detailed explanation of the process provided in the following sections.

5.2.1 Functional Association Rule Representation

A FAR chromosome encodes a FAR in such a way that each gene encodes a variable. The first gene indicates, which variable is on the RHS of the FAR while the rest indicate those on its LHS. Specifically, given the dataset $X = [x_1, x_2, ..., x_i, ..., x_n]$, the FAR chromosome is of the form $r = [o, e_1, e_2, e_3, ..., e_i, ..., e_n]$, where o is an integer value indicating the RHS variable of the FAR and e_1 to e_n binary values indicating whether the corresponding variables are on the LHS. The above encoding scheme only encodes one FAR in one chromosome in contrast to that for the LFAR in the last chapter, an update that increases evaluation efficiency. The complete mixing strategy for the adapted CCEA requires each FAR chromosome to pair up with every individual. The FAR chromosome, which encodes one FAR in it evaluates only once with one ANN individual in the other sub-population.

An ANN chromosome encodes the weights of an ANN using either a binary or real representation. As real coding has the advantage of being compact and a natural representation, we choose real values for the ANNs weight coding.

5.2.2 Evaluation Strategy

As a valid FAR can be applied to predict the value of its RHS variable, its predictive accuracy on an unseen data set reflects its validity in a similar way to the confidence measure for the conventional association rule. Therefore, the objective function uses this predictive accuracy as its component for the mining process (the given dataset ($\mathbf{X}$) is split into a training set ($\mathbf{X}_{train}$) and a test set ($\mathbf{X}_{test}$)). The other component of the objective function is the distance measure, which is used to push the search

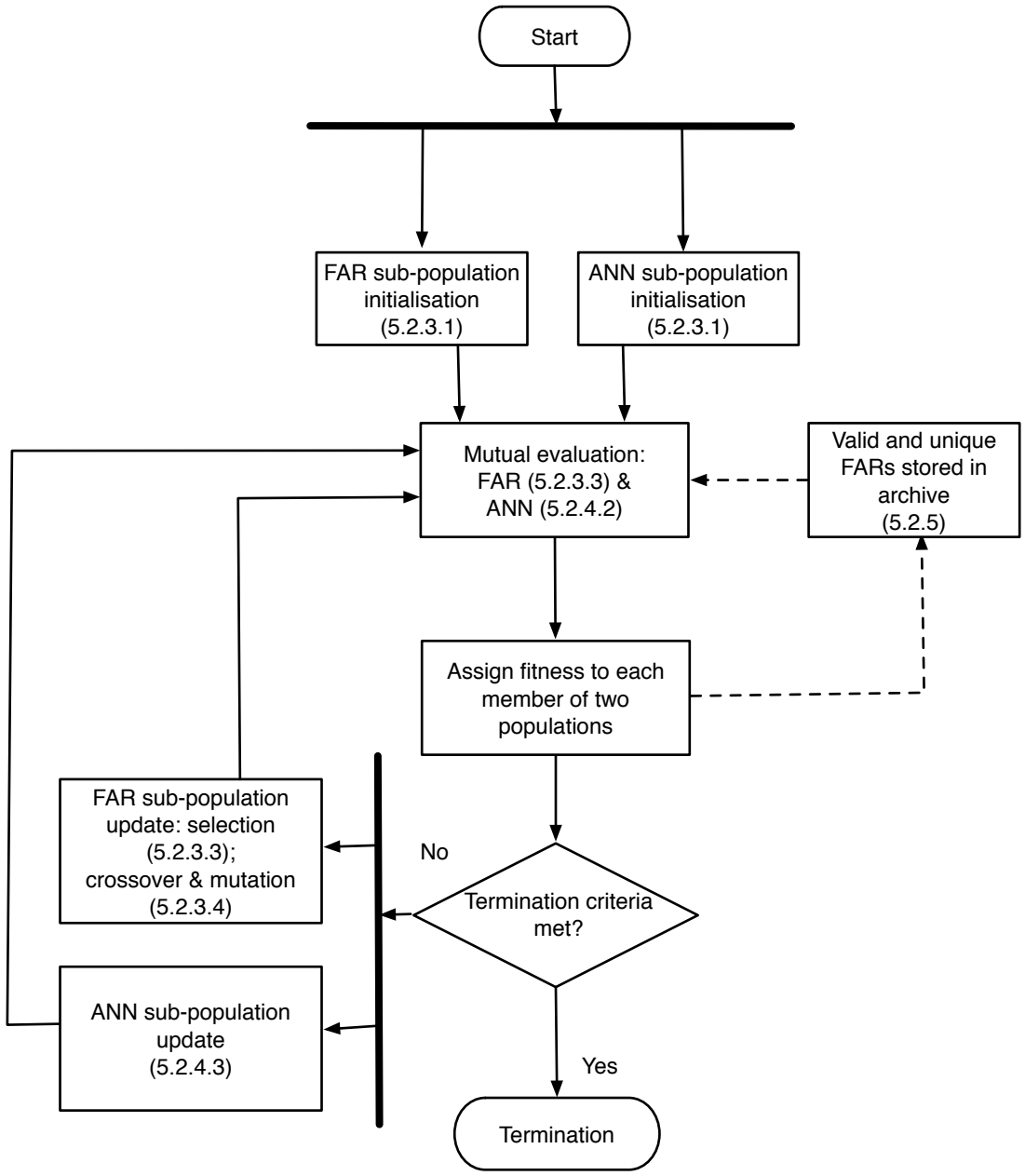


Figure 5.1: cooperative co-evolutionary functional association rule mining algorithm (numbers represent corresponding sections in this chapter and Algorithm 11 describes general process)

to expand in its search space. The FARs that are more valid and more different than others are favoured in the evolution, with the objective function for each FAR individual:

$$\vartheta_{r_i} = \frac{c_{r_i}}{(d_{r_i})^2} \tag{5.2}$$

where c_{r_i} indicates the best predictive accuracy of r_i. Suppose the FAR population is of size n_r and the ANN population of size n_a. Then, the predictive accuracy of r_i paired with an ANN (a_j) is represented by c_{ij} and, since each r_i is exposed to every ANN, is:

$$c_{r_i} = \max_j \ c_{ij};\ j = 1, 2, ..., n_a \tag{5.3}$$

The operator *max* influences the evolutionary process by preferring more valid FARs while d_{r_i} is a distance measure indicating how different a FAR is from other FARs in both the archive and its current population and is:

$$d_{r_i} = \frac{1}{1 + \delta_{r_i(pop)} + \delta_{r_i(arc)}} \tag{5.4}$$

where $\delta_{r_i(pop)}$ is the average hamming distance (h) between r_i and the rest of the FARs in the FAR sub-population, and $\delta_{r_i(arc)}$ the average hamming distance between r_i and the FARs stored in the archive. $\delta_{r_i(arc)}$ starts to affect fitness when the size of archive n_c exceeds a certain threshold (h_c) and:

$$\delta_{r_i(pop)} = \frac{1}{n_r - 1} \sum_{l=1, l \neq i}^{n_r} h(r_i, r_l) \tag{5.5}$$

$$\delta_{r_i(arc)} = \begin{cases} 0 & \text{If } n_c < h_c \\ \frac{1}{n_c} \sum_{l=1}^{n_c} h(r_i, r_l) & \text{If } n_c >= h_c \end{cases} \tag{5.6}$$

Figures 5.2 and 5.4 illustrate the process for calculating the predictive accuracy (c_{r_i}) and distance measure (d_{r_i}) for a FAR.

The objective function for an ANN individual (a_j) follows the same form and the ANN's accuracy measure is presented in Figure 5.3.

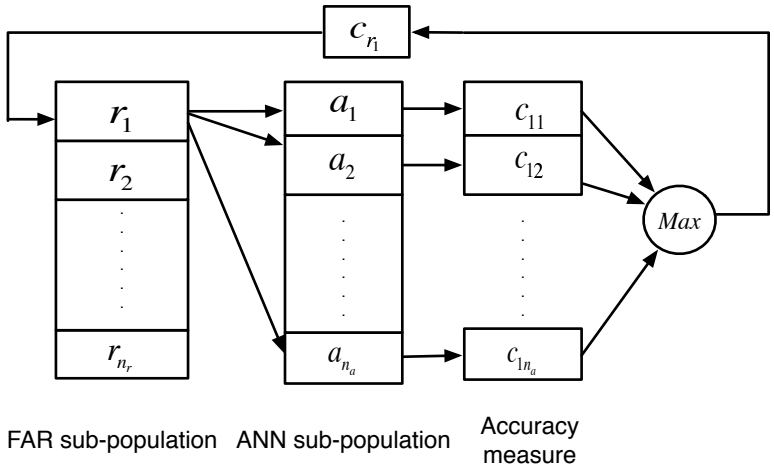


Figure 5.2: Visualisation of calculating accuracy (c_{r_1}) for FAR individual (r_1).

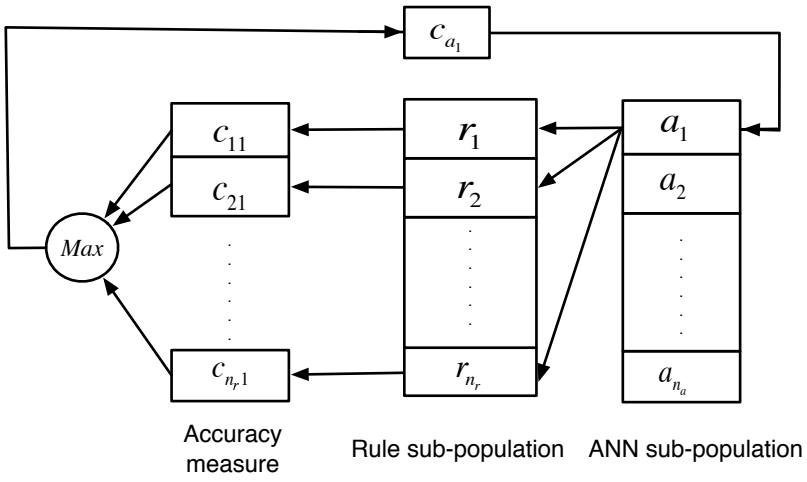


Figure 5.3: Visualisation of calculating accuracy (c_{a_1}) for ANN individual (a_1)

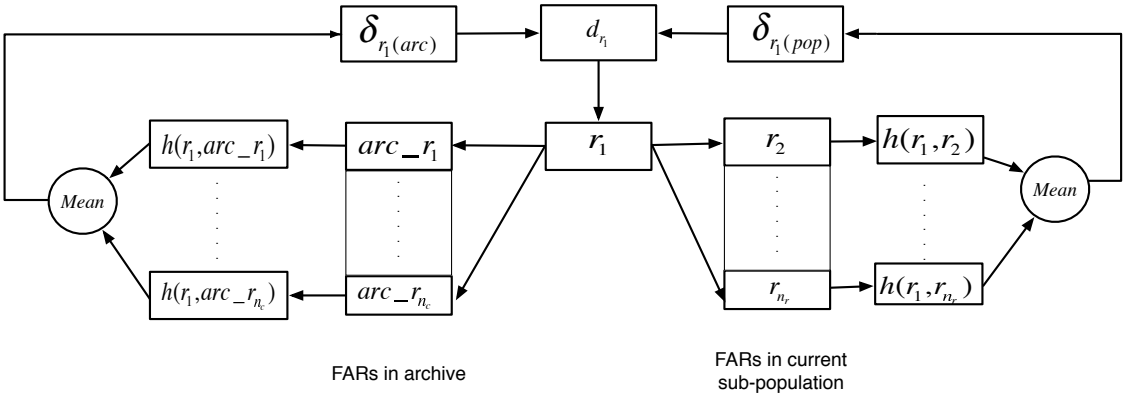


Figure 5.4: Calculation of distance (d_{r_1}) for FAR (r_1) (Notion arc_r_1 used to distinguish FARs stored in archive from those in current population)

$$\vartheta_{a_j} = \frac{c_{a_j}}{(d_{a_j})^2} \tag{5.7}$$

$$c_{a_j} = \max_i \; c_{ij};\; i = 1, 2, ..., n_r \tag{5.8}$$

The distance measure for a_j is adopted from the best-matching FAR (r_m) of a_j. The corresponding ANN of the FAR that has a high fitness for generating offspring also achieves a high fitness.

$$d_{a_j} = d_{r_m};\; m = \arg\max_i \; c_{ij};\; i = 1, 2, ..., n_r \tag{5.9}$$

5.2.3 Search Strategy: Evolution of the Functional Association Rule Sub-population

5.2.3.1 FAR Sub-population Initialisation

We first create a random initial population of FAR chromosomes, each of which is a vector of length $n + 1$, which represents a candidate FAR. For each individual chromosome, its first element, which indicates the RHS variable, is an integer generated

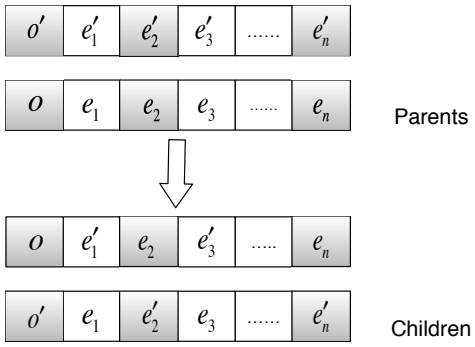


Figure 5.5: Example of using uniform crossover operator on FAR sub-population

between $[1, n]$ using a uniform distribution. The other elements (binary bits) are also generated using a uniform distribution and indicate whether the corresponding variable appears on the LHS of the FAR, with 0 meaning 'yes' and 1 'no'. The variable indicated by the first chromosome element appears on only the RHS.

5.2.3.2 Mutual Evaluation (FARs)

The FAR sub-population evaluation is conducted according to Equation 5.2-Equation 5.6.

5.2.3.3 FAR Sub-population Update (selection)

Binary tournament selection [126] is conducted to select individuals from the population for the generation of a mating pool. Then, individuals from the mating pool are used to create new offspring by applying crossover and mutation operators.

5.2.3.4 FAR Sub-population Update (Crossover and Mutation)

The crossover function uses a uniform crossover operator to exchange genes between two parent FARs, as shown in Figure 5.5. We apply two mutation operators on the offspring: for the integer gene (d), to re-select a value between $[1, n]$ to form a new RHS using uniform distribution; and, for the binary genes (e), to flip their values to 0 or 1.

5.2.4 Search Strategy: Evolution of the Artificial Neural Network Sub-population

5.2.4.1 ANN Sub-population Initialisation

For initialisation, a random population of ANNs, with each ANN a multi-layer feed-forward neural network (FFNN) with fixed numbers of layers and nodes, is created. For the input layer, the number of nodes is determined by the number of variables in the given dataset. When an ANN approximates a FAR, the input nodes that correspond to the LHS of the FAR are set to active and the other nodes inactive, which means that their input values are set to a constant 0. The output layer is restricted to one node as, according to our FAR construction, the RHS of a FAR has only one variable. For this investigation, the hidden layer is set manually, with the number of nodes a maximum of 10. The initial weights are assigned with random values between $[0, 1]$ using a uniform distribution. The activation function of the nodes in both the hidden and output layers is sigmoid.

5.2.4.2 Mutual Evaluation (ANNs)

The evaluation of each ANN (a_j) (where $j = 1, 2, ..., n_a$) is based on Equation 5.7 to Equation 5.9.

5.2.4.3 ANN Sub-population Update

Differential evolution (DE) [190], which is a versatile population-based optimiser over continuous domains, is implemented to evolve the ANN population. Since we encode the ANN chromosome using a real representation, DE is a compatible tool for ANN evolution [208, 89, 1]. In the last chapter, the classic DE algorithm was discussed in the context of the LFARM problem in Algorithm 9. In this chapter, the DE algorithm is adapted to evolve ANNs using a modification inspired by the method proposed by Abbass [1], due to its simplicity and competitive performance in function approximation. The pseudo-code for crossover and mutation is given in

Algorithm 10.

Algorithm 10: Generation of a trial individual for ANN evolution: $trialANN(R_c, R_m, \omega, n_\omega)$

Input : crossover rate (R_c), mutation rate (R_m), ANN weight vector (ω), number of weights in ANN (n_ω)
Output: trial ANN (a_c)

Select individual at random as main parent (a_{p1}), and two individuals (a_{p2} and a_{p3}) as supporting parents for generating trial individual (a_c);
$jr \leftarrow rnd(1, n_\omega)$
for $k \leftarrow 1$ **to** n_ω **do**
    $pr \leftarrow rand(0, 1)$
    if $pr < R_c$ *or* $k = jr$ **then**
        $\omega_k^{a_c} \leftarrow \omega_k^{a_{p1}} + Gaussian(0, 1) \times (\omega_k^{a_{p2}} - \omega_k^{a_{p3}})$
    else
        $\omega_k^{a_c} \leftarrow \omega_k^{a_{p1}}$
    end
    $pr \leftarrow rand(0, 1)$
    if $pr < R_m$ **then**
        $\omega_k^{a_c} = \omega_k^{a_c} + Gaussian(0, R_m)$
    end
end
Return generated trial ANN (a_c)

Using a uniform distribution, the function $rnd(1, n_\omega)$ in Algorithm 10 generates an integer number sampled from $[1, n_\omega]$ and $rand(0, 1)$ a real value sampled from $[0, 1]$ (as defined in Section 4.1.3.1, Chapter 4).

Also, each trial individual goes through the evaluation process defined in Equation 5.7 to Equation 5.9. The selection operator implemented in this study is adopted from [1], where the individual being compared is the main parent. It compares the fitnesses of the trial individual and its main parent, and selects the one that performs better for the next generation as:

$$a_s = \begin{cases} a_c & \text{If } \vartheta_{a_c} >= \vartheta_{a_{p1}} \\ a_{p1} & \text{Otherwise} \end{cases} \tag{5.10}$$

a_s represents the ANN individual selected for the next generation.

5.2.5 Global Archive

Similar to the cooperative co-evolutionary system, the archive is comprised of two related parts, a FAR archive and an ANN archive. The former stores the valid FARs found during the evolutionary process, with the criterion that the predictive accuracy value (c_r) of a FAR exceeds a predefined threshold (h_r), and the latter holds the ANNs that can represent the FARs in the FAR archive.

The FAR archive also serves as a base for pushing the FAR search to spread out in its search space by adapting the evolutionary search to focus on the unpopulated region. As the FARs in the archive represent the regions already explored in the search space, the current population only needs to search new regions, with the second FAR distance measure (d_r) used to reflect this feature. Thereby, those FARs that reside far away from explored regions are encouraged to evolve.

After the mutual evaluation of the two sub-populations, if the accuracy measure of a FAR exceeds a predefined threshold (h_r), that FAR and its corresponding ANN are placed in the archive, with feature selection applied. Sequential backward selection of each valid FAR is executed to eliminate redundant variables [192], with variables not contributing to the relation dropped and only unique FARs gaining admission to the archive.

The main steps of the above mining approach is presented in Algorithm 11 corresponding to Figure 5.1

Algorithm 11: Main steps in CCFARM algorithm

Input : dataset (**X**)
Output: FAR archive (**F**)

```
Population (FAR and ANN) initialisation;
while termination criteria not met do
    Mutual evaluation
    Archive valid solutions
    FAR new population selection, crossover, mutation
    ANN new population crossover, mutation, selection
end
Return F
```

5.2.6 Pruning

The main idea behind pruning is to perform an extraction procedure on the archive (**F**) to create a concise output set. The archive stores all the FARs identified during co-evolution that meet the accuracy measure and minimum variable involvement measure (feature selection). For each unique RHS variable in the FAR archive, we extract only the one with the best c_r value.

It is not practical to extract the complete set of FARs hidden in the data. Therefore, for the set of valid FARs found by the algorithm, the question of interest is how well the hypotheses represented by the FARs match the real underlying hidden relations. This investigation is executed on synthetic datasets where underlying associative relations can be ascertained from prior knowledge.

5.3 Experiments on Functional Association Rule Mining

In order to test the performance of CCFARM, a set of experiments is carried out to cover several objectives. Firstly, we run tests on synthetic datasets to check that CCFARM is able to find hidden relations among continuous variables. We also adjust the complexity of the hidden relations to test the changes in the FARs mined by CCFARM. The second set of experiments is conducted on real-world datasets from the UCI repository to test whether the CCFARM algorithm can identify default relations defined by the domain experts in them. In addition, an experiment that compares it with two state-of-the-art quantitative ARM algorithms in the literature is presented to demonstrate its performance. Finally, we discuss an alternative FAR form with an enhanced flexibility for its RHS variables.

Table 5.1: List of parameters used in experiments

Parameter	Symbol used in context	Value
FAR sub-population size	n_r	30
ANN sub-population size	n_a	14
No. generations	n_g	50
Crossover rate	R_c	0.8
Mutation rate	R_m	0.1
Accuracy threshold	h_r	0.95
Learning rate	l_r	0.1
No. of epochs	n_e	500
ANN structure (hidden nodes)	-	10

5.3.1 Synthetic Dataset Generation and Experiment Parameters

We use polynomial functions to program the hidden relations into the synthetic datasets. A synthetic dataset is formed by 20 continuous variables ($X = [x_1, x_2, ..., x_{20}]$) with a size of 500 instances after their values are randomly generated by a uniform distribution over the range $[0, 1]$. Then, the polynomial functions are written into the dataset through a two-step process: selecting the independent variables (IVs) from the first 10 variables and the dependent variable (DV) from the rest; and forming the polynomial function and replacing the values of the DV variable with those calculated from the function.

Suppose we want to write a polynomial function of the third order into one dataset. Firstly, a uniform distribution is used to randomly select 3 variables from x_1 to x_{10}. Assuming that they are x_{10}, x_7, and x_5, the corresponding function is then written as $x_{11} = 1 + x_{10} + x_7^2 + x_5^3$, with each instance of x_{11} consequently replaced by values calculated from this function. For multiple polynomial functions, we use sampling without replacement and, when the candidate sampling set for IVs is empty, we refill it with the original 10 variables (x_1 to x_{10}).

Therefore, the complexity of the hidden relations is reflected in two dimensions;

that is, the number of functions contained in one dataset and the highest order of them. To denote different datasets, we use a notation (Dp-q) to reflect these two dimensions, where D refers to a dataset, p to the number of functions and q to the highest order of these functions; for example, a dataset with two polynomial functions of the third order is denoted as D3-2. The highest number and order of functions hidden in a dataset of all 50 synthetic datasets are 10 and 5 respectively.

The proposed algorithm is applied to each of the 50 datasets with 20 seeds using the experimental parameters shown in Table 5.1. The parameter values for CCFARM and ANN are those commonly used in the literature [47]. As for the ANN structure, since the underlying relations are polynomial up to the fifth order, a relatively complex structure (10 hidden nodes) is chosen to capture the hidden non-linear relations. Experiments are carried out on an Oracle/Sun Cluster in which each node has two quad-core 2.93GHz Intel Nehalem CPUs, with 15 cpus used for each run and average run times of 60 to 120 mins. We can compute the time complexity of the cooperative co-evolutionary process as follows. Suppose two population sizes are n_r and n_a, and the complete mix interaction scheme between the two populations is run at $O(n_r \times n_a)$. Then, the time complexity is $O(n_g \times n_r \times n_a)$ for the core cooperative co-evolution algorithm.

In order to distinguish the hidden functions and FARs, we use the term 'function set' to refer to the set of hidden functions a dataset contains and 'FAR set' for the set of FARs after pruning. Table 5.2 shows an example of these two sets for D3-4.

5.3.2 Performance Metrics and Analysis of Experimental Results

To find FARs that match the original functions, as in the example shown in Table 5.2, the CCFARM can identify the original hidden functions in its FAR set. We are now interested in its strength when the complexity of the hidden relations increases. This is examined by a function-matching metric defined as the percentage of the hidden functions in the function set that has a matching FAR in the FAR set.

Table 5.2: Example of hidden functions and FARs extracted from dataset $D3$-4

Dataset	Function set	FAR set	Accuracy
$D3$-4	$x_{11} = 1 + x_{10} + x_7^2 + x_5^3$	$f(x_5, x_7, x_{10}) \Rightarrow x_{11}$*	0.996
	$x_{12} = 1 + x_4 + x_9^2 + x_3^3$	$f(x_3, x_4, x_9) \Rightarrow x_{12}$*	0.998
	$x_{13} = 1 + x_6 + x_8^2 + x_2^3$	$f(x_2, x_6, x_8) \Rightarrow x_{13}$*	0.998
	$x_{14} = 1 + x_1 + x_5^2 + x_4^3$	$f(x_1, x_4, x_5) \Rightarrow x_{14}$*	0.996
		$f(x_4, x_5, x_{14}) \Rightarrow x_1$	0.996
		$f(x_4, x_9, x_{12}) \Rightarrow x_3$	0.984
		$f(x_3, x_9, x_{12}) \Rightarrow x_4$	0.994
		$f(x_1, x_4, x_{14}) \Rightarrow x_5$	0.991
		$f(x_2, x_8, x_{13}) \Rightarrow x_6$	0.994
		$f(x_5, x_{10}, x_{11}) \Rightarrow x_7$	0.984
		$f(x_2, x_6, x_{13}) \Rightarrow x_8$	0.984
		$f(x_3, x_4, x_{12}) \Rightarrow x_9$	0.987
		$f(x_5, x_7, x_{11}) \Rightarrow x_{10}$	0.993

The matching criterion considers whether the variables in a FAR match the variables in a hidden function with IV/DV separation not considered. Since we are aware that the IVs and DV of the polynomial function are exchangeable in this experiment, our interest is in revealing under what situation the constituent components of the FARs deviate from those of the hidden functions.

Figure 5.6 shows the plot of this percentage from each dataset. The matching percentage indicates a decreasing trend when the number of hidden functions increases. Note that 'decreasing' does not mean that the mismatched FARs are invalid as all FARs in the FAR set have passed the accuracy check (the h_r threshold) but these increases suggest that multiple functions use the same variables as their IVs; for example, as shown in Figure 5.6, in dataset $D2$-8, where the matching percentage drops, 60% of the IV candidate set ($[x_1, x_2, ..., x_{10}]$) is sampled twice and, in dataset $D4$-8, 100% three times. This demonstrates that CCFARM is sensitive to the overlapping of hidden relations.

The sensitivity of CCFARM depends on the level of overlap in the functional relations. When the same variables are used in multiple functions, new functional relations emerge; for example, from $x_{12} = 1 + x_8 + x_{10}^2$ and $x_{20} = 1 + x_5 + x_{10}^2$, a new relation such as $x_{12} = x_8 + x_{20} - x_5$ could be formed, as shown in Table 5.3. As such relations are indistinguishable from the perspective of CCFARM, the matching percentages decrease at certain points.

Table 5.3: Example of discovered relations from dataset $D2$-10

Dataset	Hidden function	Relevant FAR	Accuracy
$D2$-10	$x_{12} = 1 + x_8 + x_{10}^2$	$f(x_5, x_8, x_{20}) \Rightarrow x_{12}$	0.998
	$x_{20} = 1 + x_5 + x_{10}^2$	$f(x_5, x_8, x_{12}) \Rightarrow x_{20}$	0.999

Essential relation metric, DV Matching: the functional relations hidden in the synthetic datasets are, in essence, formed by the DVs responding to the IVs values. We are now interested in whether, in experimental settings, such DVs can be identified as the RHSs of FARs. The results show that, in each experimental dataset, 100% of the DVs in the function sets appear as RHSs in the FAR set that suggest good DV matching.

The above two metrics show us the characteristics of the output. On one hand, the FAR set identifies the underlying relations, which, on the other, deviate from the original form in terms of constituent components when the hidden relations overlap with each other. The following metrics are designed to study the characteristics of the differences.

Difference metric, active variable ratio: the active variable ratio (ar) is the ratio of the number of variables appearing in a FAR set or function set to the number of all variables in a given dataset and is:

$$ar = \frac{n_{av}}{n} \tag{5.11}$$

where n_{av} is the number of variables in a given dataset appearing in a set (function

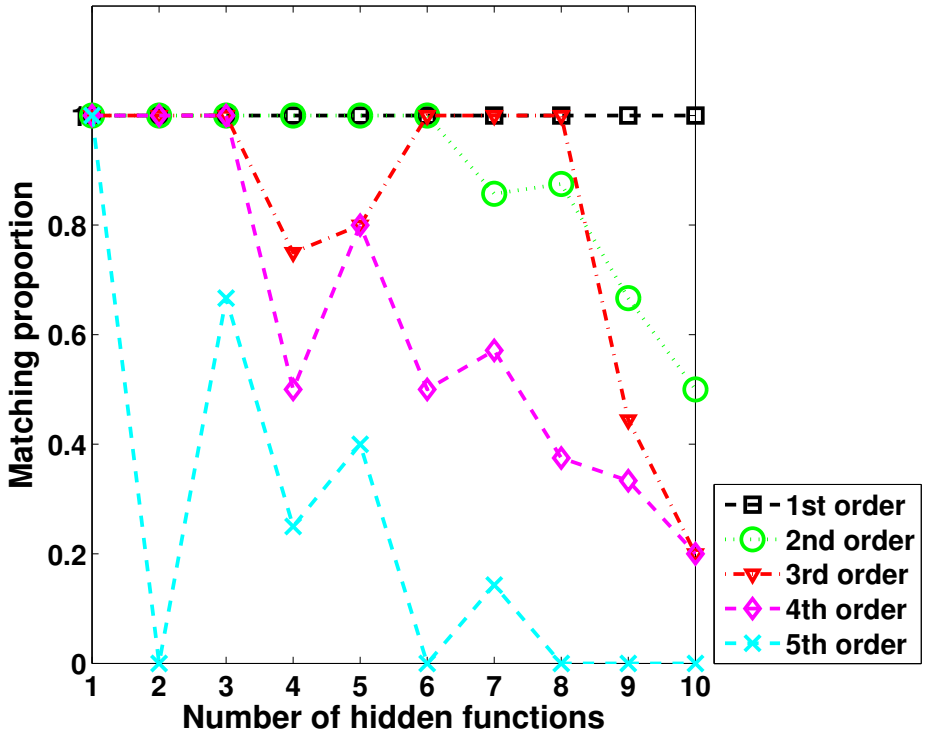


Figure 5.6: Matching results for underlying relations in each dataset

Table 5.4: Comparison of active variable ratio (hidden/found) for different datasets

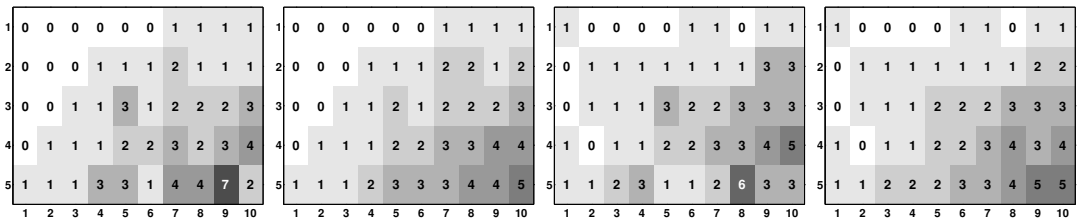
Dataset	q=1	q =2	q =3	q =4	q =5	q =6	q =7	q =8	q =9	q =10
D1- ($p = 1$)	0.10	0.20	0.30	0.40	0.50	0.60	0.70	0.80	0.90	1.00
D2- ($p = 2$)	0.15	0.30	0.45	0.60	0.75	0.80	0.85	0.90	0.95	1.00
D3- ($p = 3$)	0.20	0.40	0.60	0.70	0.75	0.80	0.85	0.90	0.95	1.00
D4- ($p = 4$)	0.25	0.50	0.65	0.70	0.75	0.80	0.85	0.90	0.95	1.00
D5- ($p = 5$)	0.30	0.60	0.65	0.70	0.75	0.80	0.85	0.90	0.95	1.00

or FAR) and n is the number of all variables in the same dataset; for example, if we write two functions ($x_{11} = 1+x_{10}+x_4^2$ and $x_{12} = 1+x_9+x_2^2$) in one dataset, the active ratio of the function set should be $6/20 = 0.3$. Table 5.4 shows the comparison of the active variable ratios calculated from each function set and corresponding FAR set, where there is only one value for each dataset because these ratios are exactly the same in the two sets. This table illustrates the consistency of the active variable ratio between the function and FAR sets. It implies that, of the 50 datasets on which experiments are conducted, the CCFARM is capable of avoiding inactive variables, which are those not used in hidden function generation and do not contribute to any hidden relation in a given dataset. Table 5.4 shows that they are also not included in the corresponding FAR set, which supports the correctness of the FAR set for mining hidden relations.

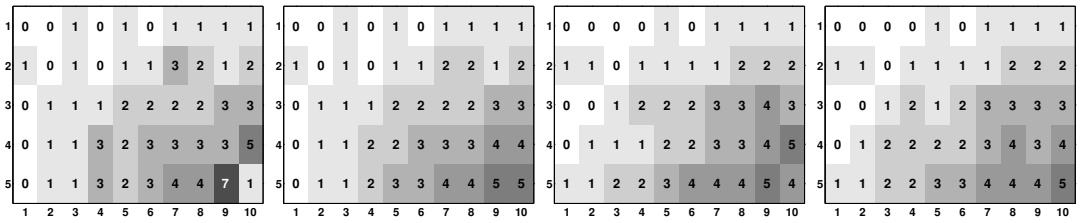
***Difference metric, frequency*:** as discussed above, the FARs in the FAR set could be different from the hidden functions in terms of their constituent variables due to the discovered functions. The frequency metric is a visualisation tool for reflecting this difference by counting the number of occurrences of every variable in the function or FAR set. Due to the IV/DV exchangeable feature of the polynomial functions, in the FAR set, if two FARs have the same variables, they are merged into one. We visualise the results using the greyscale figures shown in Figure 5.7 and 5.8, in which each plot represents a variable and its number of occurrences in either the function or FAR set.

A significant difference is observed for the case of DVs ($[x_{11}, x_{12}, ..., x_{20}]$), as shown in Figure 5.8. The DVs in FAR sets are observed to have higher frequencies than those in the function sets, particularly for the datasets with both higher number of hidden functions and more complex functions. This suggests that FARs are prone to including discovered functions.

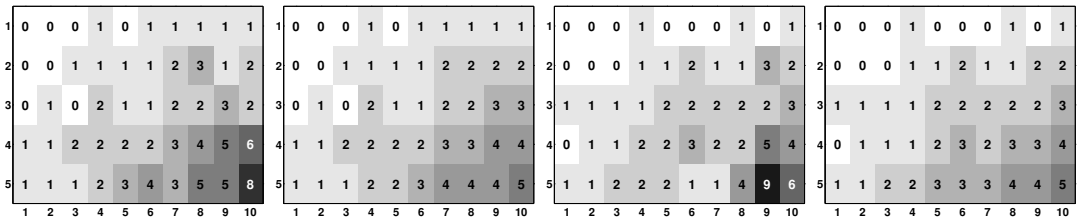
***Principal component visualisation*:** another visualisation tool for comparing the differences between the FARs and hidden functions is the principal component analysis (PCA), which we use to investigate whether a FAR and its corresponding function behave similarly. it is first applied to form the principal compo-



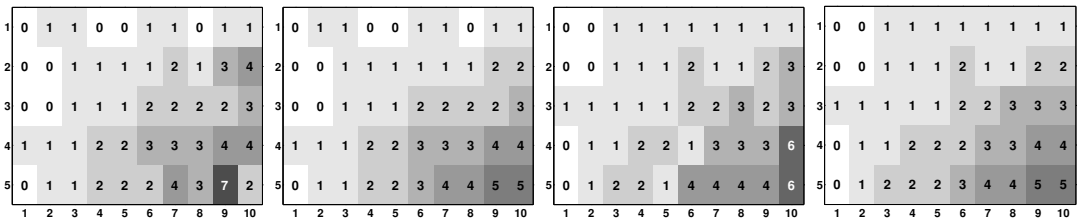
(a) variable x_1 in FAR sets (b) variable x_1 in function sets (c) variable x_2 in FAR sets (d) variable x_2 in function sets



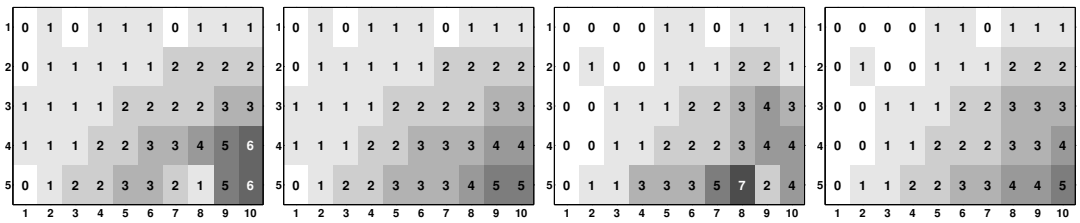
(e) variable x_3 in FAR sets (f) variable x_3 in function sets (g) variable x_4 in FAR sets (h) variable x_4 in function sets



(i) variable x_5 in FAR sets (j) variable x_5 in function sets (k) variable x_6 in FAR sets (l) variable x_6 in function sets

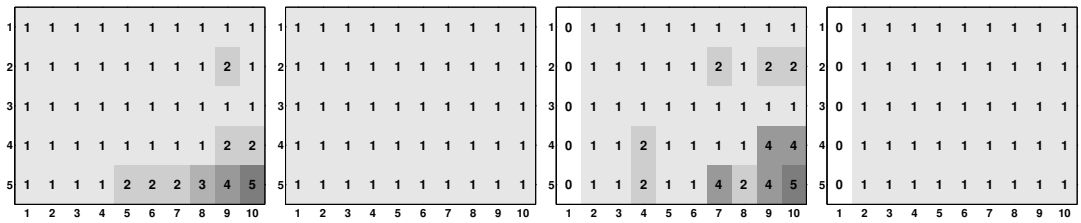


(m) variable x_7 in FAR sets (n) variable x_7 in function sets (o) variable x_8 in FAR sets (p) variable x_8 in function sets

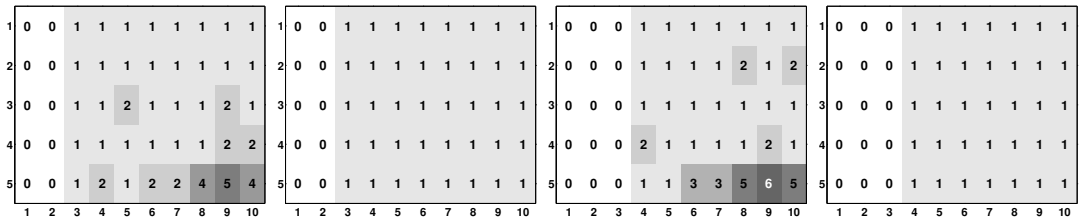


(q) variable x_9 in FAR sets (r) variable x_9 in function sets (s) variable x_{10} in FAR sets (t) variable x_{10} in function sets

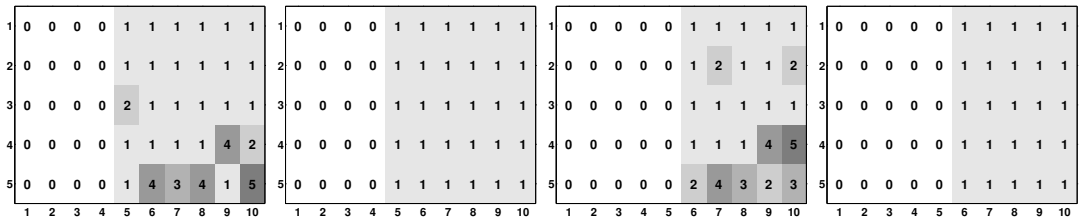
Figure 5.7: variable frequencies of FAR and function sets for variables x_1 to x_{10}. (each figure represents frequencies of one variable in 50 different dataset, with horizontal axis indicating the number of hidden functions and vertical axis their order)



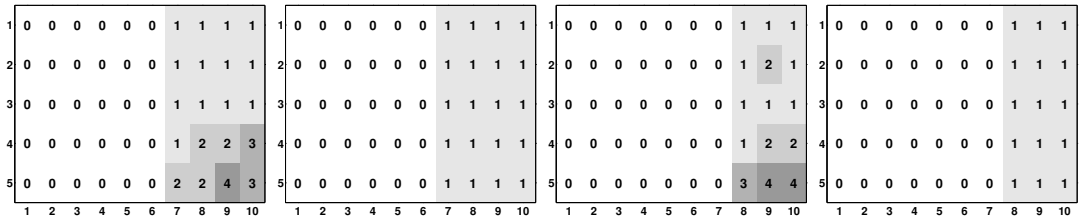
(a) variable x_{11} in FAR sets (b) variable x_{11} in function sets (c) variable x_{12} in FAR sets (d) variable x_{12} in function sets



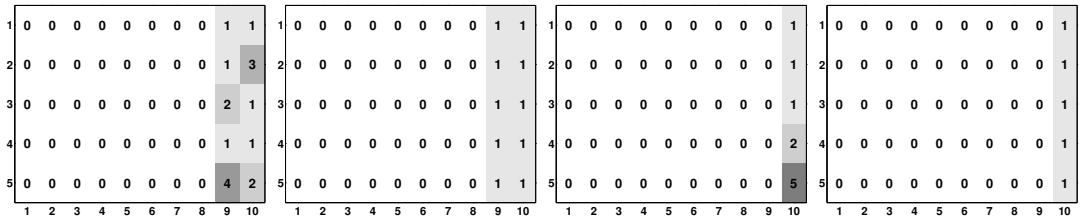
(e) variable x_{13} in FAR sets (f) variable x_{13} in function sets (g) variable x_{14} in FAR sets (h) variable x_{14} in function sets



(i) variable x_{15} in FAR sets (j) variable x_{15} in function sets (k) variable x_{16} in FAR sets (l) variable x_{16} in function sets



(m) variable x_{17} in FAR sets (n) variable x_{17} in function sets (o) variable x_{18} in FAR sets (p) variable x_{18} in function sets



(q) variable x_{19} in FAR sets (r) variable x_{19} in function sets (s) variable x_{20} in FAR sets (t) variable x_{20} in function sets

Figure 5.8: variable frequencies of FAR and function sets for variables x_{11} to x_{20}. (each figure represents frequencies of one variable in 50 different dataset, with horizontal axis indicating the number of hidden functions and vertical axis their order)

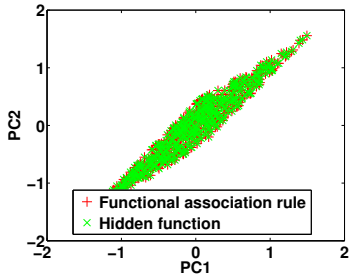
nent space and then instances of the variables involved in the hidden function or FAR are plotted against the first two principal components.

We select a few functions and their corresponding FARs that are affected by discovered relations, examples of which are shown in Figure 5.9 to Figure 5.12. In Figure 5.9(g), it can be seen that the PC plots of FARs and the hidden functions are very similar, but do not overlap exactly. This is due to the differences in constituent variables between FARs and the hidden functions as demonstrated in Table 5.3.

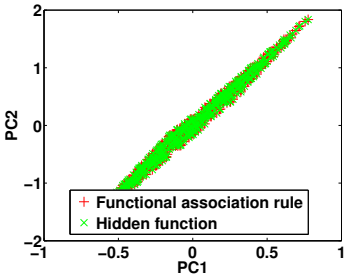
From the analysis of synthetic datasets, we can summarise that the FAR sets can focus on the relevant variables of hidden relations (as shown in the active variable ratio) and the FARs may not be in exactly the same form as the original hidden relations when the complexity increases (as shown in the function matching and frequency metrics). However, the FAR sets are capable of encompassing the essential relations in given datasets (as shown in DV identification) and uncovering original relations when the underlying relations do not overlap.

Table 5.5: List of parameters used in the real world data experiments

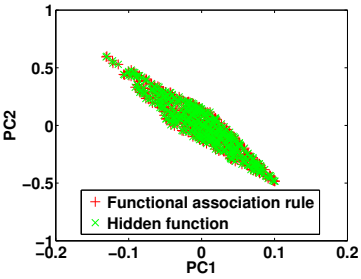
Parameters	Symbol used in this thesis	Value
Rule population size	n_r	30
ANN population size	n_a	14
Generation size	n_g	50
Crossover rate	R_c	0.8
Mutation rate	R_m	0.1
Accuracy threshold for real world data	h_r	0.8
Learning rate	lr	0.03
No. Epochs	-	100
No. nodes (hidden) for real world data	-	2/5/10



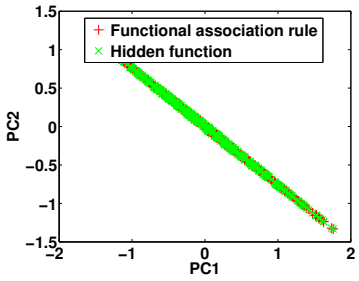
(a) first two PC plot of hidden function and FAR for DV x_{11} in dataset D2-8



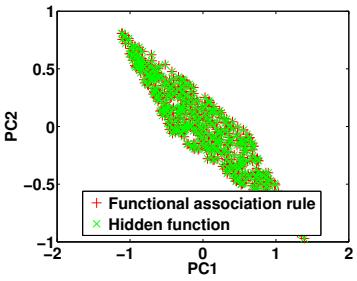
(b) first two PC plot of hidden function and FAR for DV x_{12} in dataset D2-8



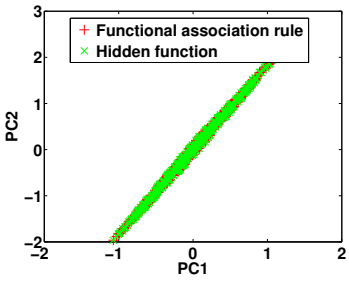
(c) first two PC plot of hidden function and FAR for DV x_{13} in dataset D2-8



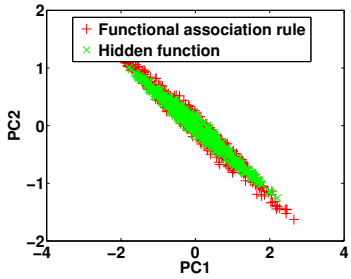
(d) first two PC plot of hidden function and FAR for DV x_{14} in dataset D2-8



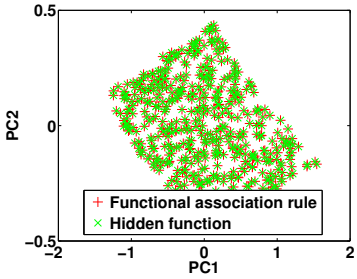
(e) first two PC plot of hidden function and FAR for DV x_{15} in dataset D2-8



(f) first two PC plot of hidden function and FAR for DV x_{16} in dataset D2-8

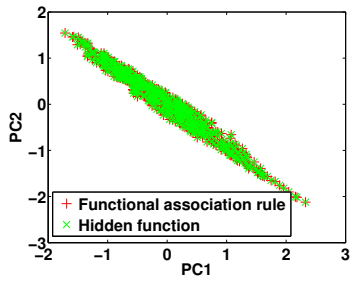


(g) first two PC plot of hidden function and FAR for DV x_{17} in dataset D2-8

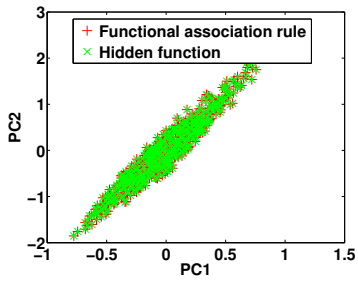


(h) first two PC plot of hidden function and FAR for DV x_{18} in dataset D2-8

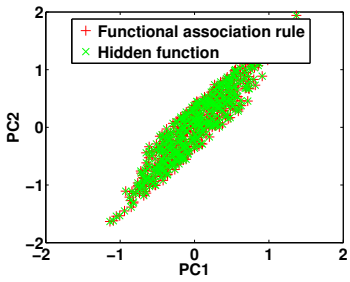
Figure 5.9: Visualisations of first two principal component (PC) plots of hidden functions and corresponding FARs for $D2$-8



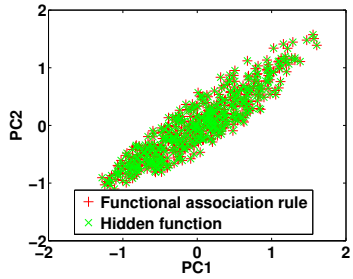
(a) First two PC plot of hidden function and FAR for DV x_{11} in dataset D3-7



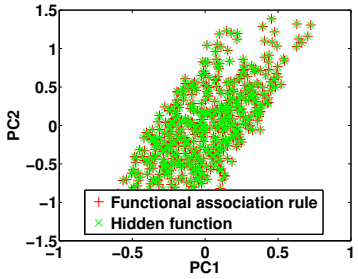
(b) First two PC plot of hidden function and FAR for DV x_{12} in dataset D3-7



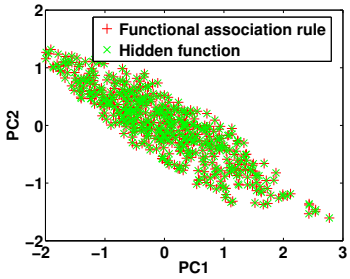
(c) First two PC plot of hidden function and FAR for DV x_{13} in dataset D3-7



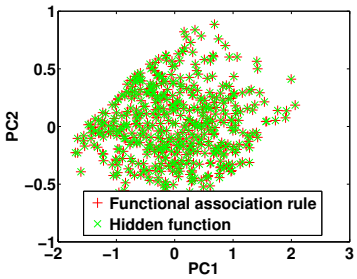
(d) First two PC plot of hidden function and FAR for DV x_{14} in dataset D3-7



(e) First two PC plot of hidden function and FAR for DV x_{15} in dataset D3-7

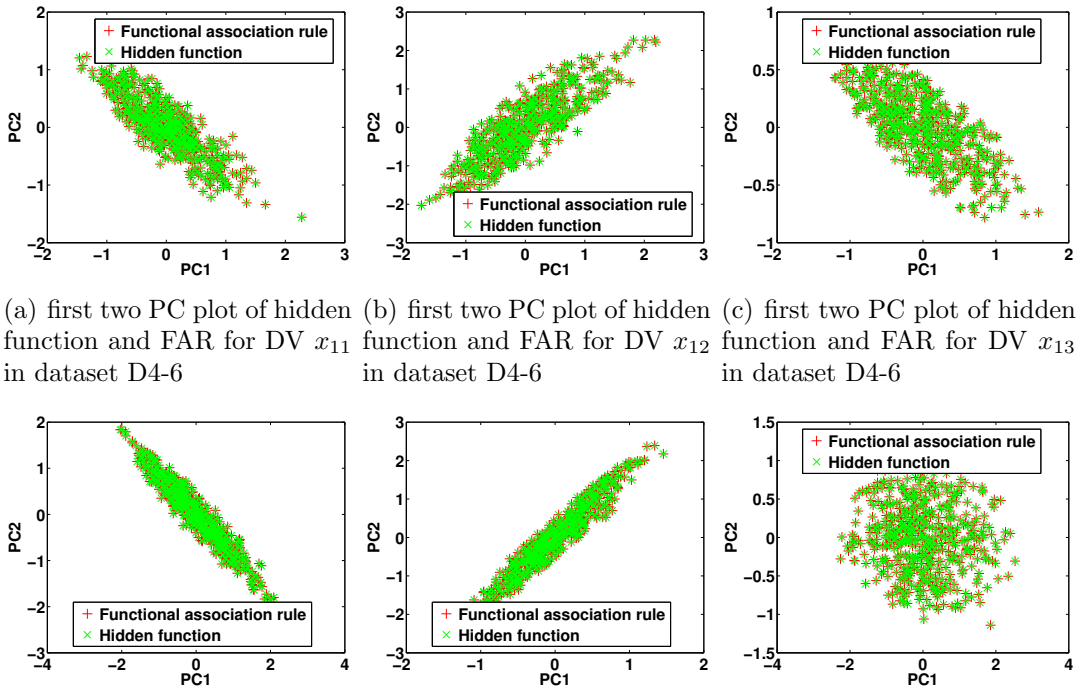


(f) First two PC plot of hidden function and FAR for DV x_{16} in dataset D3-7



(g) First two PC plot of hidden function and FAR for DV x_{17} in dataset D3-7

Figure 5.10: Visualisations of first two PC plots of hidden functions and corresponding FARs for $D3$-7



(a) first two PC plot of hidden function and FAR for DV x_{11} in dataset D4-6

(b) first two PC plot of hidden function and FAR for DV x_{12} in dataset D4-6

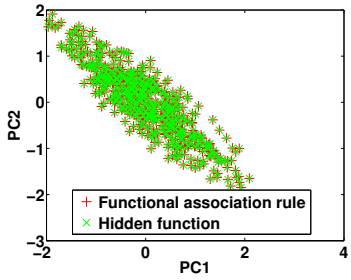
(c) first two PC plot of hidden function and FAR for DV x_{13} in dataset D4-6

(d) first two PC plot of hidden function and FAR for DV x_{14} in dataset D4-6

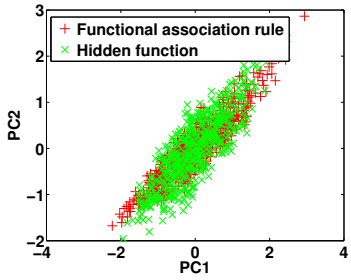
(e) first two PC plot of hidden function and FAR for DV x_{15} in dataset D4-6

(f) first two PC plot of hidden function and FAR for DV x_{16} in dataset D4-6

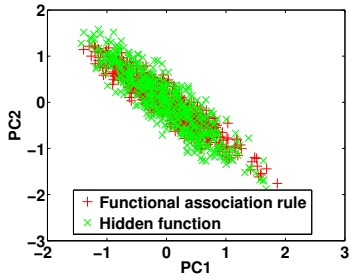
Figure 5.11: Visualisations of first two PC plots of hidden functions and corresponding FARs for $D4$-6



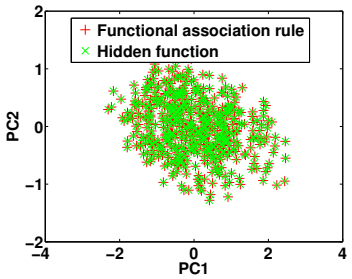
(a) first two PC plot of hidden function and FAR for DV x_{12} in dataset D5-5



(b) first two PC plot of hidden function and FAR for DV x_{13} in dataset D5-5



(c) first two PC plot of hidden function and FAR for DV x_{14} in dataset D5-5



(d) first two PC plot of hidden function and FAR for DV x_{15} in dataset D5-5

Figure 5.12: Visualisations of first two PC plots of hidden functions and corresponding FARs for $D5\text{-}5$

5.3.3 Experiments on Real-world Datasets

The original purpose of CCFARM is to identify hidden associative relations in given datasets and assist further investigations of causal relations. The experiments on the synthetic datasets contribute to our understanding of the final output from the mining algorithm. In summary, a FAR set represents a good duplication of a function set when the hidden functions' IVs do not greatly overlap; otherwise, it can comprise emerging relations from the hidden functions and provide more FARs than the function set. A FAR set is able to avoid inactive variables to the extent of the complexity that the experimental data possess.

In the following experiments, to check the performance of CCFARM, we apply it to 10 real-world datasets from the UCI data mining repository and Belkent University function approximation repository. The datasets are Breast Cancer Wisconsin (BCW) original and diagnostic, Sonar, Concrete, Body fat, Sleep, Vine yard, Pollution, Bolts and Stock price, the details of which are shown in Table 5.6. These datasets are selected essentially due to the fact that they contain continuous variables. In addition, they also have been used in experiments for two state-of-the-art ARM algorithms concerning continuous variables, which makes them suitable candidates for conducting comparison experiments. For the two BCW datasets, the variable representing participant ID is deleted, and hence the number of variables is one less than that stated in the repository website. In addition, the instances with missing data in these datasets are removed. This treatment for missing data is chosen according to our assumption in the AHGP definition. The AHGP assumes that the prior knowledge about the given dataset is not provided and, consequently, domain knowledge about the meaning and interrelations of the variables should not be considered in the algorithm. A range of other missing data treatment methods require analysis of the characteristics of missing data (e.g. whether the missing data corresponds to IVs or DVs). This is often conducted by human experts.

The parameters used in the experiments are shown in Table 5.5. The parameter values for CCFARM and ANN are the standard values used in the literature [1, 8].

Table 5.6: Parameters used in experiments on read datasets

Database	No. variables	No. instances	Source
Sleep	8	51	Bilkent
Vineyard	4	52	Bilkent
Pollution	16	60	Bilkent
Bolts	8	40	Bilkent
Body fat	18	252	Bilkent
Stock price	950	10	Bilkent
BCW(original)	10	683	UCI
BCW(diagnostic)	31	569	UCI
Concrete(strength)	9	1030	UCI
Sonar	61	208	UCI

The accuracy threshold is empirically selected for finding relatively more valid potential relations. Tenfold crossover validation is not applied due to its computational complexity. During the evolution process, the evaluation of a single rule is conducted using a collection of candidate neural networks. Training without ten-fold crossover could lead to bias. However, a FAR will be re-evaluated by the experimental causal search algorithm (introduced in the next chapter), which reduces the potential bias introduced here. Experiments on a number of different ANN structures are conducted in order to test the sensitivity of CCFARM to the ANN structures used.

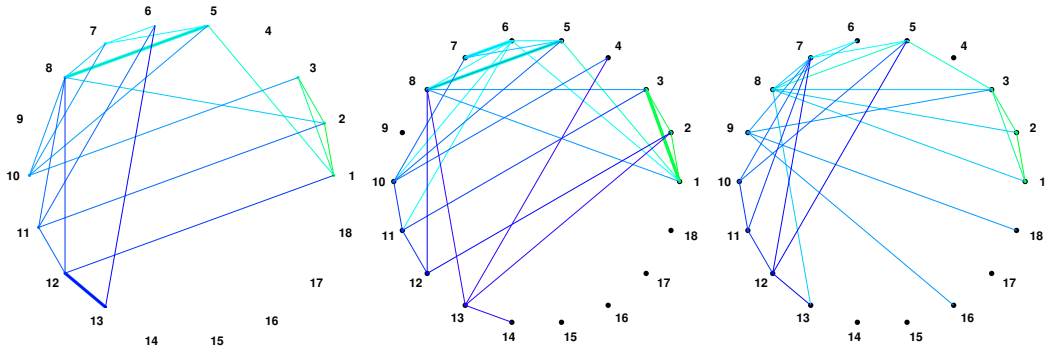
In contrast to the above synthetic dataset experiments, for these real world datasets, the only prior knowledge available is their default tasks defined by domain experts (i.e. predicting the values of the last variable in the dataset). These default regression/classification tasks specify the basic relations in the datasets. Our experiments are not focused on comparisons with previously reported results in literature, but on whether the CCFARM algorithm can identify these basic relations. We run the algorithm on these datasets with 30 different seeds, and the results are shown in Table 5.7. We checked whether the default task is identified in terms of (1) whether there is a FAR in the final output using the last variable as its RHS variable, and

Table 5.7: Percentage of the default task identification among 10 real world datasets

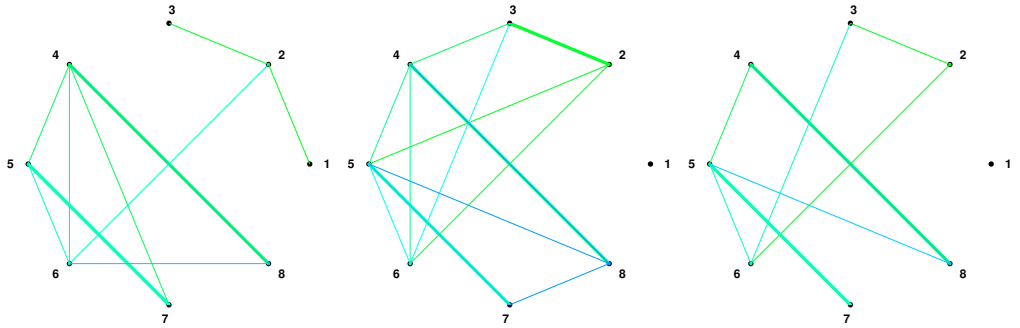
ANN structure		Percentage of identification
ANN with 2 hidden nodes	DV as RHS	50%
	DV included	80%
ANN with 5 hidden nodes	DV as RHS	50%
	DV included	60%
ANN with 10 hidden nodes	DV as RHS	50%
	DV included	80%

(2) whether the last variable is involved in a FAR. Different ANN structures do not affect the percentage of default task identified, if we only consider the first criterion. When applying the second criterion, the identification rate slightly increases. These results suggest that CCFARM is able to find basic relations in these datasets. However, there are some default tasks not identified, it can be attributed to the single accuracy threshold value (h_r) used across all the datasets. This implies a potential further work for the CCFARM algorithm, which is to investigate how to improve the adaptivity of the CCFARM when different datasets are given.

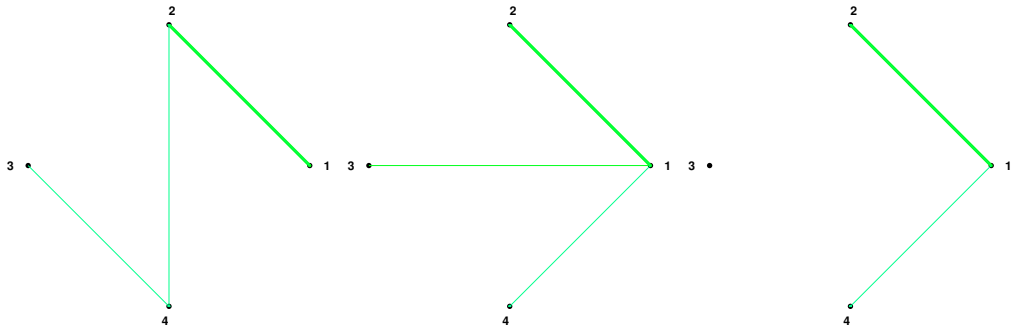
In addition to extracting a single default task, we integrate other FARs into network forms to visualise the overall interdependency among the variables based on all extracted relations from each dataset. These networks are plotted in Figure 5.13, 5.14, 5.15 and, in them, the variables in the same FAR are connected with links to one another. It is possible that multiple links exist between two variables; for example, in Figure 5.14 (I), two FARs ($f(x_3, x_10) \Rightarrow x_2$ and $f(x_2, x_6) \Rightarrow x_{10}$) are extracted from the corresponding archive and, when plotting them, there will be two links between x_2 and x_{10}. In this situation, these links are plotted with a strengthened width to indicate that both the connected nodes play roles on the RHS and include the other as a LHS in their FARs. Some of the networks are sparsely connected, e.g., Figure 5.15 (a) to (c), which means that there are limited basic functions in the datasets while others shows more complicated variable inter-



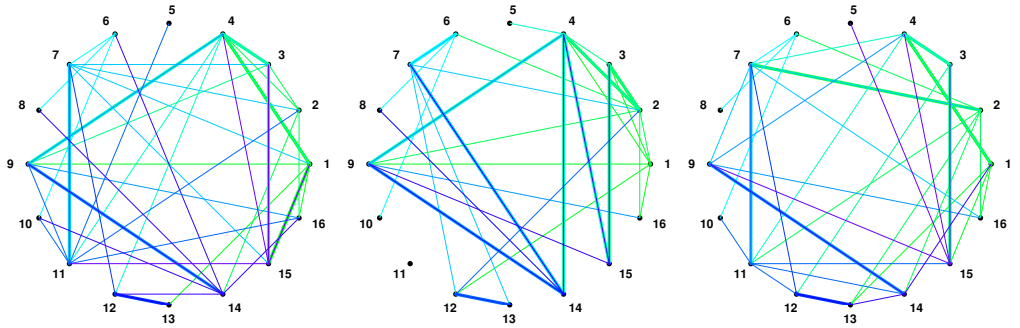
(a) Body fat FARs network (b) Body fat FARs network (c) Body fat FARs network



(d) Sleep FARs network (e) Sleep FARs network (f) Sleep FARs network

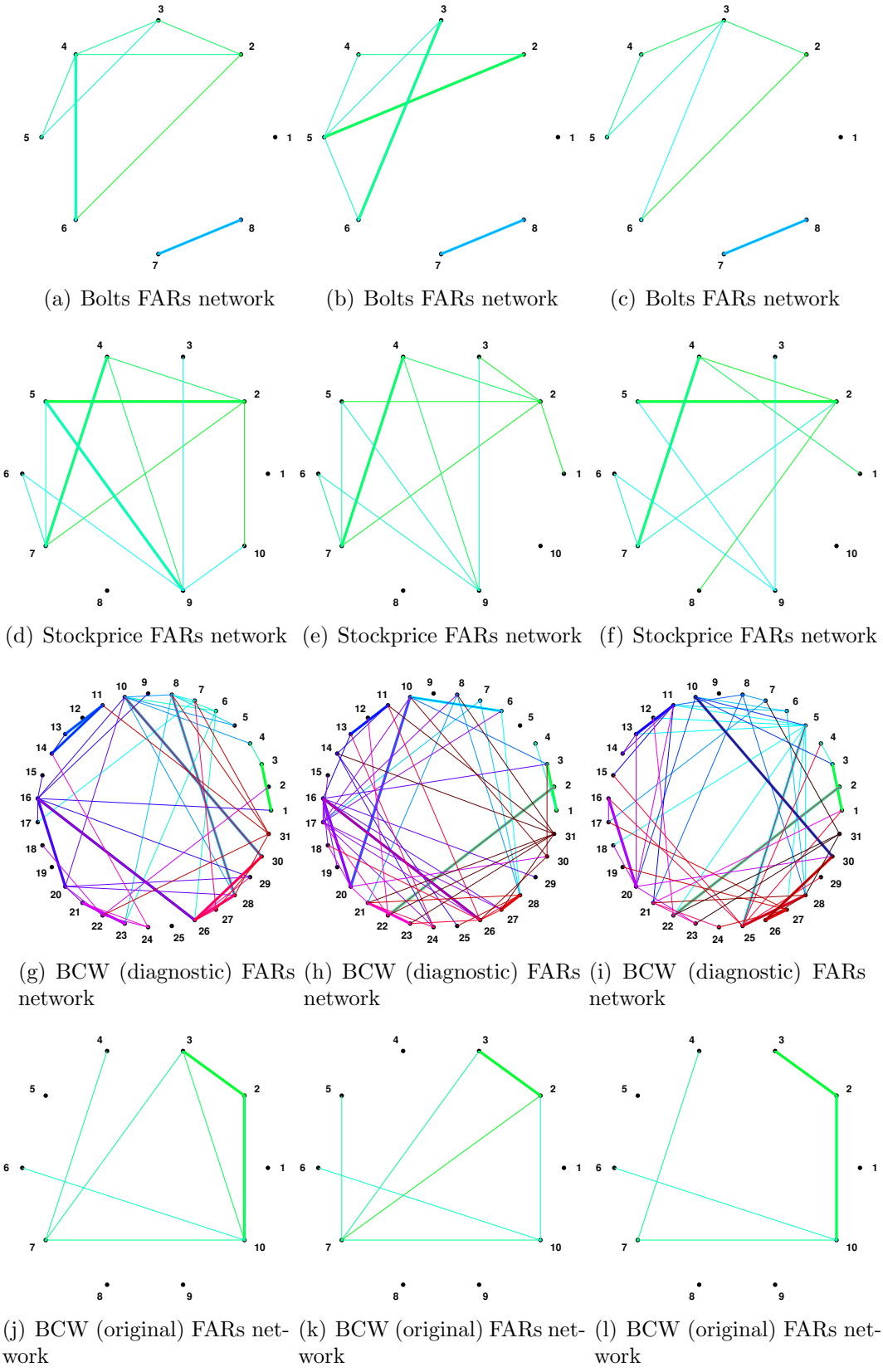


(g) Vine yard FARs network (h) Vine yard FARs network (i) Vine yard FARs network



(j) Pollution FARs network (k) Pollution FARs network (l) Pollution FARs network

Figure 5.13: Networks generated by FARs extracted from each archive (numbers refer to variables in respective datasets)



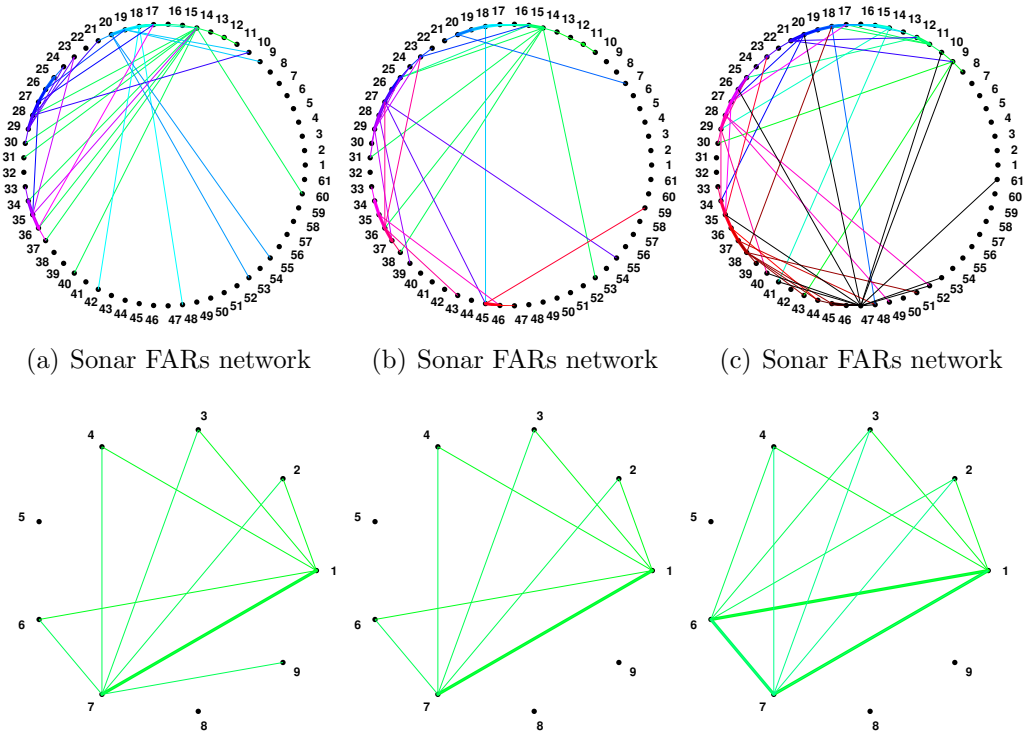
(a) Bolts FARs network (b) Bolts FARs network (c) Bolts FARs network

(d) Stockprice FARs network (e) Stockprice FARs network (f) Stockprice FARs network

(g) BCW (diagnostic) FARs network (h) BCW (diagnostic) FARs network (i) BCW (diagnostic) FARs network

(j) BCW (original) FARs network (k) BCW (original) FARs network (l) BCW (original) FARs network

Figure 5.14: Networks generated by FARs extracted from each archive (numbers refer to variables in respective datasets)



(a) Sonar FARs network (b) Sonar FARs network (c) Sonar FARs network

(d) Concrete FARs network (e) Concrete FARs network (f) Concrete FARs network

Figure 5.15: Networks generated by FARs extracted from each archive (numbers refer to variables in respective datasets)

actions. Such representations illustrate the hidden relations and interdependencies in a dataset. They present domain experts with preliminary knowledge about given data and reduce the amount of exploration work on data usually undertaken by human domain experts. In the literature, studies using the BCW (original) data have been reported with different algorithms. For example, Wilson [203] reported that if clump thickness (x_1) is 7 and uniformity of cell size (x_2) is 5 or above, then malignancy (x_{10}) is indicated; If bland chromatin (x_7) is 8 or greater, then malignancy is indicated (variable x_{10}); If uniformity of cell shape (x_3) is 8 or above and marginal adhesion (x_4) is not 1, then malignancy is indicated (x_{10}). There are clear consistencies with the network generated for BCW (original) (Figure 5.14 (j, k, l)), as the variable x_{10} (malignant or benign) is directly related to uniformity of cell size (x_2), bland chromatin (x_7), and uniformity of cell shape (x_3). This variable (x_{10}) is also indirectly related to marginal adhesion (x_4). The relation between clump thickness (x_1) and the class variable (x_{10}) is not found.

5.3.4 Comparison Experiments with other Evolutionary Computation based Quantitative Association Rule Mining Algorithms

FAR can not only be used as input to the CHGP but also independently for ARM tasks. In order to demonstrate the performance of the proposed CCFARM in this respect, we conduct experiments to compare it with two state-of-the-art continuous variable ARM techniques: GAR (Genetic ARM algorithm [119]); and MODENAR (multi-objective DE algorithm for mining numeric association rules [8]).

These two algorithms, which are based on evolutionary computation, aim to mine the interval-based association rules without the discretisation pre-processing, while CCFARM works on FARs. Although, CCFARM works on a novel ARM form, since it is also based on heuristic search and evolutionary computation, it is appropriate to compare it with the other two state-of-the-art ARM algorithms. The experiments are conducted on six public domain databases available from [69]: body

fat, bolts, pollution, sleep, stock price and vineyard, as they are the datasets used in the other two algorithms. The summary of these data sets is given in Table 5.6, and the parameters used in these experiments are shown in Table 5.5. The experiments are conducted with 30 random seeds.

Due to their different association rule forms, it is difficult to compare the accuracies of the FARs mined by different algorithms. However, as the confidence measure of conventional association rules can be seen as a predictive accuracy when used for prediction, we base the comparison on it. The accuracies of the FARs refer to their predictive accuracies on unseen test data, while those of the interval-based association rules refer to rule confidence, but both measure the strengths of the derived associations in terms of prediction. The accuracy results are shown in Table 5.8, in which it can be seen that the accuracies of the FARs extracted by CCFARM are relatively higher compared with those by MODENAR. Different ANN structures do not show constant impact on the performances of CCFARM regarding this metric, as in dataset Pollution, the highest accuracy is achieved when using ANN with 5 nodes, while in dataset Sleep, it is achieved when using ANN with 2 nodes.

The coverage metric shows the percentage of data covered by the derived FARs and other association rules. The CCFARM extracts FARs with a lower coverage compared with GAR and MODENAR as shown in Table 5.9. Standard deviation is shown for the purpose of comparing the performance of the algorithm proposed in this thesis with those reported in the literature. In four datasets (Sleep,Vine yard, Pollution and Bolts), the CCFARM has relatively higher coverage. Again, the different ANN structures do not impact the performance of the CCFARM regarding this metric of coverage.

The metric rule size shows the mean numbers of variables contained in the derived FARs. As shown in Table 5.10, the FARs from CCFARM are generally smaller than the association rules from the other two algorithms, which could be attributable to the feature selection process, where the LHS variables of the potentially interesting FARs are checked for their contributions to the FARs when entering the archive. Those variables that do not contribute significantly to the associative rela-

Table 5.8: Comparisons of the results with MODENAR on metric accuracy

Data set	Accuracy/Confidence (%) with standard deviation			
	CCFARM (2 nodes)	CCFARM (5 nodes)	CCFARM (10 nodes)	MODENAR
Body fat	**94±6.5**	**94±7.1**	**94±7.1**	62±3.2
Sleep	**86±5.8**	85±5.9	85±5.1	64±3.4
Vine yard	89±6.4	**90±5.6**	89±6.9	–
Pollution	88±6.0	**90±5.7**	88±7.3	67±2.7
Bolts	**100±0.1**	**100±0.0**	**100±0.1**	65±1.8
Stock price	**84±3.4**	**84±5.0**	83±4.9	56±1.9

tions are eliminated from the FARs. As for the ANN structure, it does not impact the performance of the CCFARM regarding this metric.

The comparisons of these metrics indicate that CCFARM is competitive with other interval-based ARM algorithms in terms of finding FARs with a relatively high predictive accuracy and smaller sizes, however, as for the data coverage, CCFARM does not show evident advantages. The ANN structure does not impact the performance of the CCFARM on any of the metrics.

5.3.5 Alternative Functional Association Rule Form

In the previous section, we introduced the coding scheme with only one RHS to demonstrate that the algorithm is capable of extracting hidden relations from continuous datasets. In this section, we provide another complementary coding scheme that allows for an enhanced flexibility of the FAR form, where the RHS can have multiple variables. Such a chromosome has the same number of genes as the instances (X) in a given dataset and is of the form $[e_1, e_2, ..., e_n]$, with each gene having three candidate values $\{0, 1, 2\}$. Value (0) is interpreted as the variable not included in the FAR encoded in this chromosome, the value (1) as the corresponding

Table 5.9: Comparisons of the results with GAR and MODENAR on coverage metric

Data set	Coverage (%)				
	CCFARM (2 nodes)	CCFARM (5 nodes)	CCFARM (10 nodes)	MODENAR	GAR
Body fat	61.1	72.2	77.8	86.0	**86.1**
Sleep	**100.0**	87.5	87.5	80.6	79.0
Vine yard	**100.0**	**100.0**	75.0	–	**100.0**
Pollution	**100.0**	93.8	**100.0**	95.0	95.0
Bolts	**87.5**	**87.5**	**87.5**	80.0	77.5
Stock price	80.0	80.0	90.0	**98.7**	**98.7**

Table 5.10: Comparisons of the results with GAR and MODENAR on rule size metric

Data set	Rule size				
	CCFARM (2 nodes)	CCFARM (5 nodes)	CCFARM (10 nodes)	GAR	MODENAR
Body fat	**3.3**	3.6	3.5	7.5	6.9
Sleep	3	3.1	**2.7**	4.2	4.2
Vine yard	2.3	2.3	**2**	3.0	–
Pollution	4.9	4.6	**4.5**	7.3	6.2
Bolts	2.6	2.6	**2.3**	5.2	5.2
Stock price	4.0	4.3	**3.4**	5.8	6.0

variable appearing in the LHS of the FAR and the last value (2) indicating that the variable is in the RHS of the FAR. The coding scheme for the ANN is the same as that introduced in Section 5.2.1 and the co-evolutionary process the same as that applied for a single RHS FAR.

We compare this alternative FAR form with its single RHS output counterpart on three datasets, with the results shown in Table 5.11. The accuracy measures in the first column are presented with standard deviations. By using the multiple output form, the average predictive accuracy is slightly lower than that of the single output form of FARs, which can be attributed to the predictive accuracy being averaged over multiple outputs. The multiple RHS has a higher rule size on the body fat and sleep datasets but a lower one on the vineyard dataset. Regarding coverage, the multiple RHS covers fewer variables than the single RHS on the vineyard dataset but more on the other two datasets.

5.4 Chapter Summary

In this chapter, we introduced a more general FAR form for variables in continuous domains, with the general functional relation represented by an ANN and a CCFARM for mining such FARs presented. The experiments on synthetic and real-world datasets showed that the proposed algorithm was able to identify similar basic relations to those hidden in the dataset and provide an insight into the underlying regularities of the dataset. Other than producing input for the CHGP, the proposed CCFARM could also be used to carry out the conventional quantitative ARM task. The comparison with other evolutionary computation-based quantitative association rules showed its competitive performance on mining association rules. The FARs were also extended to allow multiple output forms. Although an arbitrary output FAR was more general, it had a relatively lower predictive accuracy than a single RHS FAR.

The representational ability of the FARs encouraged us to base the CHGP on it. In particular, we preferred the single RHS form of a FAR as it more effectively

Table 5.11: Comparison of single RHS FAR and multiple RHS FAR (accuracy with 95% confidence interval)

Data		Accuracy(%)	Coverage	Size
Body fat	Single RHS	*93.7± 4.2	72.2	3.6
	Multiple RHS	79.7±0.2	100.0	7.4
Sleep	Single RHS	86.1±3.7	100.0	2.8
	Multiple RHS	83.8±1.6	87.5	3.9
Vine yard	Single RHS	89.2± 8.1	100.0	2.7
	Multiple RHS	88.1±4.9	100.0	2.6
Pollution	Single RHS	88.0± 3.6	100.0	4.4
	Multiple RHS	81.4±3.0	100.0	4.4
Bolts	Single RHS	*100.0± 0.0	87.5	2.6
	Multiple RHS	93.6±1.1	100.0	4.9
Stock price	Single RHS	86.1± 4.5	70.0	4.0
	Multiple RHS	83.8±3.0	60.0	4.0
BCW (original)	Single RHS	*90.2± 3.1	87.1	3.9
	Multiple RHS	77.6±0.1	100.0	11.0
BCW (diagnostic)	Single RHS	85.4± 4.9	60.0	3.0
	Multiple RHS	81.0±1.0	90.0	5.6
Concrete (strength)	Single RHS	91.4± 10.2	66.7	6.0
	Multiple RHS	87.4±8.9	77.8	6.3
Sonar	Single RHS	*87.3± 2.9	50.8	4.4
	Multiple RHS	–	–	–

reduces the search space for causal hypothesis generation, as presented in the next chapter.

Chapter 6

Causal Hypothesis Generation: Experimental Causal Search

The general associative hypothesis generation approach, CCFARM, introduced in the last chapter aims to identify associative relations from the observational data of a system. The associative relations of interest are not limited to linear relations. Non-linear associative relations among variables can be also potentially captured due to the combination of ANN and FAR. CCFARM is featured by a cooperative co-evolution strategy that searches both the FARs and their matching ANN in order to capture valid associative relations. The experiments show that the FARs mined by CCFARM can find the relations that are similar to the underlying associative relations of a system. The factor which leads CCFARM to perform differently regarding different datasets is identified as the overlapping of underlying relations which means that the constituent components of the FARs can be different from the real underlying relations. Except for serving as a solution to the associative hypothesis generation problem (AHGP), CCFARM can be applied to tasks of conventional continuous variable association rule mining (ARM). Comparisons of it and other evolutionary computation-based ARM algorithms show its competitive performance.

This chapter departs from the FARs that identify the associative relations from

observational data and presents the algorithm developed to solve the causal hypothesis generation problem (CHGP). In Chapter 3, we introduced a general strategy based on an agent architecture for designing solutions to the CHGP. In this chapter, details of its implementation are presented. The remainder of this chapter is organised as follows: Section 6.1 revisits the CHGP and relative causal models related to it; Section 6.2 introduces the experimental causal search algorithm based on an agent architecture; experiments on synthetic datasets are presented in Section 6.3, with both overlapping and hierarchical relations explored to test the performance of the proposed algorithm in terms of its metric error rate; in Section 6.4, a game environment within the context of retrieving causal relations is designed to explore the performance of the algorithm; and Section 6.5 discusses the conclusions drawn.

6.1 Causal Hypothesis Generation Revisited

The focus of this thesis is placed on an unknown system measured by a set of continuous variables, for which there is no prior knowledge about the system's underlying structure. The CHGP is a problem of retrieving the underlying causal relations potentially existing in a system and is defined in Chapter 3 as follows. Given a set of associative hypotheses ($\mathbf{F} = \{f_1, f_2, ..., f_j, ..., f_m\}$), m is the number of hypotheses and the goal is to retrieve the potential causal relations existing in $\mathbf{F}$. The causal hypothesis is represented by a graph (G), where an arrowhead from x_i to x_j specifies that x_i is a direct cause of x_j.

The hypothesis generation in this thesis focuses on causal relations because causation is often the central practical interest of different disciplines in terms of knowledge discovery. In medical science, the efficacy of a medicine, including its side effects, must be well studied. In epidemiology, the interest is often in the causes of diseases while social scientists look for the causes of human behaviour patterns. Retrieving causal relations from an unknown system can provide insights into the system and suggest a controlling strategy [2]; in order to change the state of an effect variable, one can apply intervention on the cause variable.

The basic question in a causal relation investigation is what forms a cause and how can it be determined. In the remainder of this section, a number of causal models are briefly discussed in the context of the above CHGP problem based on the information gained from FARs.

Hume [88] defined causality in terms of the induction of observed phenomena, arguing that causation is a metaphysical concept and, in practice, can only be adequately defined in terms of empirical regularity. He proposed three criteria for indicating causation: contiguity; succession; and constant conjunction. Mill [125] [83], who shared the same regularity view, proposed the following four rules regarding how to discover causation in practice. Suppose that L and M are two potential causes, and N a potential effect: (1) if M varies as N varies, M might be a cause of N; (2) the difference in N when M happens and when L happens indicates the cause; (3) the effect of L on N can be observed by taking the difference in N between, when L and M both happen and when only M happens; and (4) L and M are not causes of N if N does not change regardless of L and M happening. Rule (1) describes a situation which implies potential causation. In this thesis, the FARs generated from the previous AHGP serve such a purpose. Rule (2) is important as it outlines the empirical principle of discovering causation. However, it has been criticised on a variety of grounds; for example, as a causal relation is regular, the possibility of measurement error and uncertainty is precluded. Later development of the causality theory has enriched the rule of difference from a counterfactual perspective and established a systematic approach for designing experiments to conduct comparisons of different outcomes.

The counterfactual model defines causality in terms of comparisons of observable and unobservable events. Generally, a counterfactual is a conditional statement, where the first clause expresses something contrary to fact; for example, "If I had taken the medicine, my headache would have gone by now". In a counterfactual model, a *treatment* (T) is a variable that is manipulable and considered the potential cause of a certain response. A *unit* (u) refers to an object to which treatments are assigned while *concomitants* are any variables in u the values of which

are unaffected by the treatments. For any treatment applied, there are two potential outcomes from a unit, $Y_0(u)$ and $Y_1(u)$, which are the responses of u when treatment 0 ($T = 0$) and treatment 1 ($T = 1$) are applied respectively. Without any further assumption, the time when a treatment is applied is important since it is possible that treatments applied at different times cause different responses. Therefore, only one response can be observed. When $Y_0(u)$ is observed, $Y_1(u)$ becomes counterfactual and, when $Y_1(u)$ is observed, $Y_0(u)$ becomes counterfactual, with $\tau(u) = Y_1(u) - Y_0(u)$ defined as the effect of the treatment. However, this difference cannot be directly observed on the same unit and is called the fundamental problem of causal inference (FPCI). Counterfactual responses can be constructed by randomised experiments [137] in which treatments are randomly assigned to each unit, with each unit having an equal probability of receiving either the $T = 0$ or $T = 1$ treatment while assignments are independent of the concomitants.

If randomised treatment experiments cannot be conducted in some situations, in order to create a counterfactual group, a commonly used approach is matching [164]. Matching constructs paired units for comparison by selecting units of similar concomitant values which then form the counterfactual group for comparison. However as, in certain scientific settings, it is reasonable to assume that the FPCI does not apply, there are two assumptions for specifying such a situation. Temporary stability states that a response does not change if treatment times are slightly different. Causal transience describes a situation in which the response is not affected if the unit has been previously exposed to a different treatment. A third assumption is that the units are homogeneous with regard to treatments and responses. The experimental causal search algorithm uses these assumptions to form the solution to the CHGP.

A FAR specifies an associative relation among a set of variables and an instance of this set can be considered a unit in the counterfactual model. Suppose a FAR is $f(x_1, x_2) \Rightarrow x_3$. In order to investigate the potential causal relations in it, the above counterfactual causal experiment principles are adopted to design the algorithm for causal hypothesis generation. Since applying an intervention to the system

requires actions with the environment, the algorithm's design is based on an agent architecture.

There is another branch of causal modelling, automatic causal modelling, which aims to retrieve a causal structure from only observational data [183, 147]. The causal relations are defined on the assumption of faithfulness which states that the conditional independence relations presented in the data are due to only the underlying causal structure [183]. This family of approaches has the advantage of reducing the intervention of human experts. However as, for the system in which this thesis is interested, such an assumption is not satisfied, causal hypothesis generation is not approached from this perspective.

6.2 Experimental Causal Search based on Agent Architecture

The above counterfactual causality definition and its inference form the basis of the reasoning process of the proposed causal search algorithm. Since this reasoning process requires interactions with the objective system, we therefore design the causal search algorithm using an agent architecture. A single agent is used in the proposed causal search algorithm. Details of the design are presented in the following sections.

6.2.1 Sense

As defined by the CHGP, its input is the set of FARs derived from solving the AHGP ($\mathbf{F} = \{f_1, f_2, ..., f_m\}$) which comprises one part of the agent's sensed information while the information from the objective system (i.e. observational data $\mathbf{X}$) with which the agent interacts through its sensors forms the other part. The experimental causal search algorithm aims to form two groups of instances, where one group is the other's counterfactual group. The observational data ($\mathbf{X}$) provide a database from

which a number of instances can be selected to form one of the instance sets (D_s). The construction of its counterfactual instance set (D'_s) relies on the actions of the agent applying interventions to the potential causal variables. Such a construction is possible based on the temporary stability, causal transience and homogeneous assumptions introduced in the last section.

6.2.2 Reason

The reasoning part of the agent determines how to select the set of instances and actions to execute in order to construct the corresponding counterfactual set. Suppose a currently received FAR is $f(x_{p_1}, x_{p_2}, ..., x_{p_i}, ..., x_{p_k}) \Rightarrow x_q$, where $\{x_{p_1}, x_{p_2}, ..., x_{p_i}, ..., x_{p_k}\} \in X$, $x_q \in X$ and $\{x_{p_1}, x_{p_2}, ..., x_{p_i}, ..., x_{p_k}\} \cap x_q = \emptyset$. The variables involved in the current experiments are confined to those specified by this FAR, that is, $\{x_{p_1}, x_{p_2}, ..., x_{p_i}, ..., x_{p_k}, x_q\}$. The RHS variable ($x_q$) is considered as a response variable while the LHS variables ($\{x_{p_1}, x_{p_2}, ..., x_{p_i}, ..., x_{p_k}\}$) are examined one by one as potential cause variables. Suppose the potential cause variable under examination is x_{p_i} and the observational data ($\mathbf{X}$) are sorted according to its values (Algorithm 12, Line 3). Thirty instances are selected from the sorted data ($\mathbf{X}'_{\mathbf{x_{p_i}}}$) and form the set ($D_s$), which will be compared with its counterfactuals (Algorithm 12, Line 4).

In order to construct the counterfactual set (D'_s) of (D_s), the agent applies an intervention (Δx_{p_i}) to each instance of variable x_{p_i} in D_s, which is calculated as in Equation 6.1 (Algorithm 12, Line 5). In this equation, the maximum and minimum values of x_{p_i} are extracted from its instances in ($\mathbf{X}'_{\mathbf{x_{p_i}}}$) to form a pair of boundaries. Then this range is divided by 30, as 30 instances of x_{p_i} are selected from $\mathbf{X}'_{\mathbf{x_{p_i}}}$ to form D_s. This interval is further halved to form an intervention value (Δx_{p_i}). By doing so, when Δx_{p_i} is applied to x_{p_i} in D_s, the value of x_{p_i} does not exceed its original boundary. If after the intervention is applied to x_{p_i}, x_q shows a different state, then there is a potential causal relation between x_{p_i} and x_q.

$$\Delta x_{p_i} = \frac{(max(x_{p_i}) - min(x_{p_i}))/30}{2} \tag{6.1}$$

6.2.3 Action

The agent's actions are to create counterfactual instances for the corresponding instances in D_s. Given the current potential cause-effect variable pair $\{x_{p_i}, x_q\}$, for each jth instance in D_s ($D_{s,j}$), the agent adjusts the system to the same state as $D_{s,j}$ and applies the intervention (Δx) to $x_{p_{i,j}}$ (Algorithm 12, Line 10). However, as the other LHS variables might also have causal relations with the RHS variable, it is usually preferable to control them to eliminate their influences [164]. The other LHS variables are then adjusted back to the values in $D_{s,j}$ and this newly generated instance $D'_{s,j}$ forms the counterfactual for instance $D_{s,j}$ (Algorithm 12, Lines 8-13).

When all the counterfactual instances are constructed, for each one, the difference between the paired values of x_p in D_s and D'_s is calculated. In order to rule out the possible influence of random error, this difference is confirmed by the paired t-test (Algorithm 12, Line 15). If the test results show no difference between the x_q values in D_s and x'_q values in D'_s, then x_{p_i} is not the direct cause of x_q and the link between them is dropped. Otherwise, an arrowhead points from x_{p_i} to x_q to indicate that x_{p_i} is the direct cause (Algorithm 12, Lines 16-20). Figure 6.1 shows a visualisation of this process conducted on an example FAR of $f(x_1, x_2) \Rightarrow x_3$.

6.3 Experiments on Synthetic Datasets

We are interested in the accuracy of the overall algorithm, including CCFARM, for recovering the underlying mechanism. The experimental design involves two levels of investigation: separated hidden relations and chained hidden relations. The former is defined in terms of the dependent variables not being directly influenced by each other. The latter means that, among the underlying relations, the dependent variable of one relation can become the independent variable of another, settings which increase the complexity of the underlying relations. Visualisations of separated and chained hidden relations are presented in Figure 6.2 and Figure 6.3 respectively. The experiments are applied on synthetic datasets.

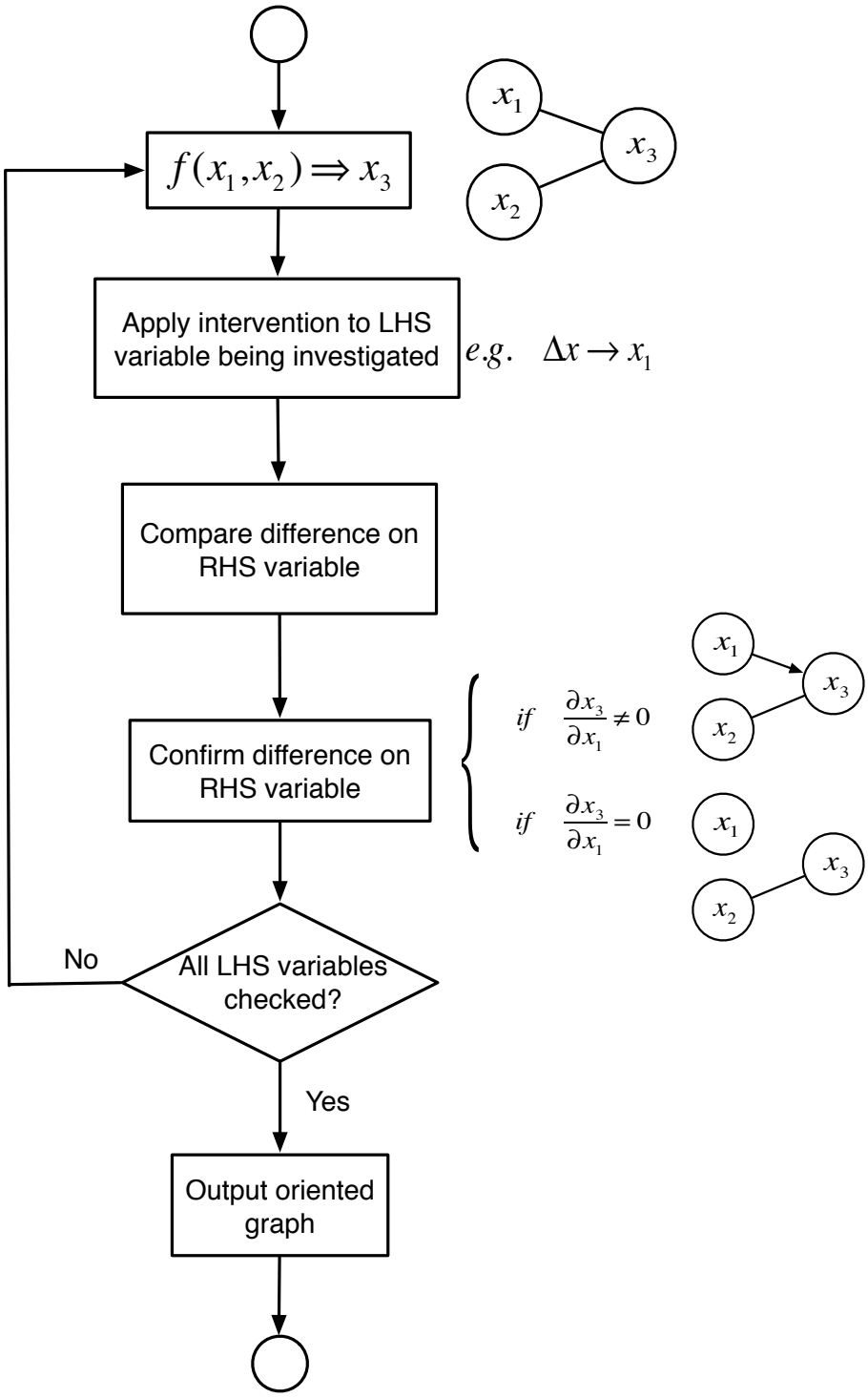


Figure 6.1: Process flow of experimental causal search for one FAR (process repeated for multiple FARs)

Algorithm 12: Experimental causal search algorithm based on agent architecture

Intput : FAR ($f(x_{p_1}, x_{p_2}, ..., x_{p_i}, ..., x_{p_k}) \Rightarrow x_q$), observational data ($\mathbf{X}$)
Output: causal relations between $\{x_{p_1}, x_{p_2}, ..., x_{p_i}, ..., x_{p_k}\}$ and $\{x_q\}$

```
1  i = 1
2  while i ≤ k do
3      Sort observational data (X) according to x_{p_i}
4      Select 30 instances from the sorted data using equal intervals to form a set
       (D_s).
5      Determine intervention value (Δx_{p_i}) according to Equation 6.1
6
7      //Creating D'_s through intervention:
8      for j ← 1 to 30 do
9          //j refers to the jth instance in D_s
10         x_{p_i,j} = x_{p_i,j} + Δx_{p_i}
11         Adjust other LHS variables to their values in D_{s,j}
12         Add this post-intervention instance to D'_s
13     end
14
15     Apply paired t-test to the values of x_q in D_s and D'_s
16     if average difference 0 then
17         Drop link between x_{p_i} and x_q
18     else
19         Orient link x_{p_i} → x_q
20     end
21     i = i + 1
22 end
```

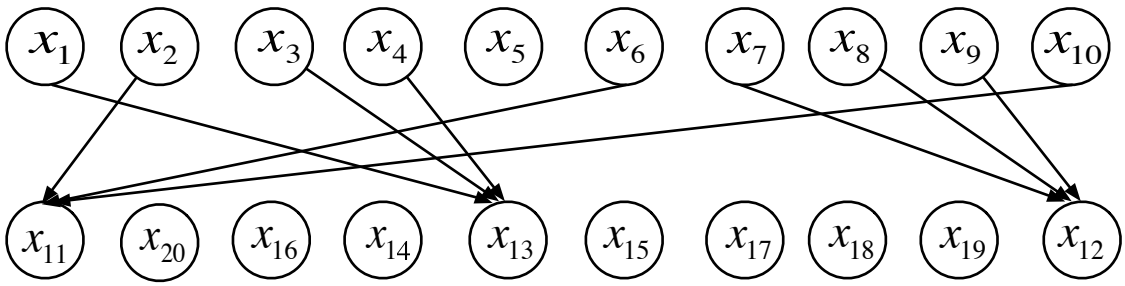


Figure 6.2: Example of separated hidden relations in D3-3

Table 6.1: List of parameters used in two synthetic data experiments

Parameter	Symbol used in context	Value
Rule population size	n_r	30
ANN population size	n_a	14
Generation size	n_g	50
Crossover rate	R_c	0.8
Mutation rate	R_m	0.1
Accuracy threshold	h_r	0.95
Learning rate	lr	0.1
No. epochs	n_e	500
No. nodes (hidden)	n_u	10

6.3.1 Experimental Design for Separated Hidden Relations

The synthetic data generation for separated hidden relations is the same as that for the dataset used in experiments with CCFARM. Each hidden relation is a polynomial function and each dataset has a different number of relations. Details of the data generation steps are provided in Chapter 5. Each dataset (D) is denoted as $Dp-q$, where p refers to the order of its hidden functions and q to its number of hidden functions.

In order to measure the accuracy of the final hypotheses derived from the causal search, an error rate metric is used. In the literature, when reconstructing a causal network from a synthetic dataset, the analysis is often conducted on either the number of links or orientation matching [182] [41], or the matching of the network [38]. Inspired by these analyses, the error rate is defined as the percentage of the links in the hidden relations not correctly identified by the algorithm:

$$error = m_h/n_h \tag{6.2}$$

where m_h refers to the number of links in the hidden relations not identified by the

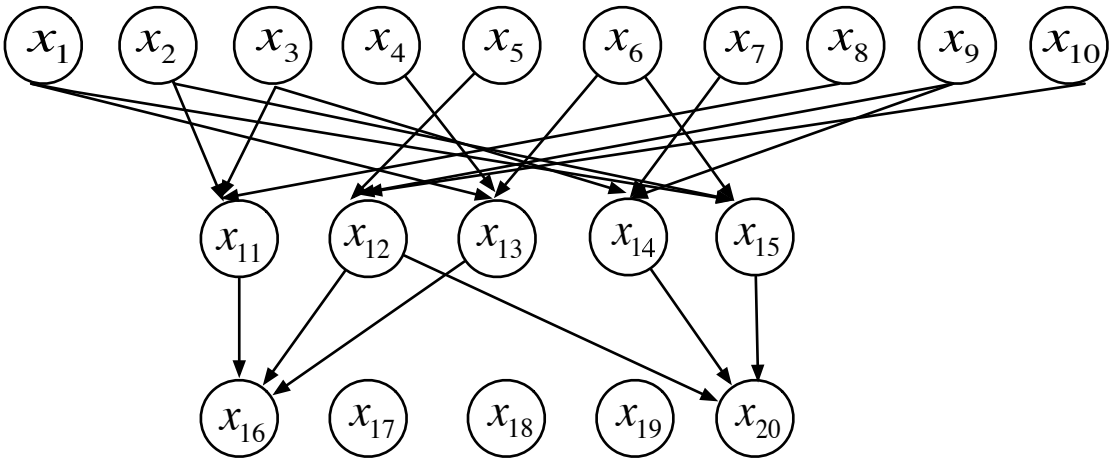


Figure 6.3: Example of chained hidden relations

search algorithm and n_h the number of overall links hidden in the dataset.

6.3.1.1 Searching Pruned Archive

In Chapter 5, we discussed the overlapping problem in the FARM. Due to the potential overlapping of the independent variables, greedy pruning may filter out the FARs that are potentially useful for retrieving the underlying mechanism. Table 6.2 shows the error rates incurred when rebuilding the causal network for each dataset from the pruned FAR set using the experimental causal search algorithm. The high error rates appearing in the right-hand bottom corner show that, when the number of hidden relations increases, using the pruned archive as a guide for causal search is unreliable. Therefore, we would suggest using the archive without pruning for a causal search. However, if the overall process is not for reverse engineering (e.g., building a causal network) but a predictive task (e.g., predicting the value of one variable), the pruned archive is still preferred as it selects the associative relations with the highest accuracies.

6.3.1.2 Searching Unpruned Archive

The following experiment is applied on the original output from the archive. Table 6.3 shows the error rates for the 50 synthetic datasets with separate hidden

Table 6.2: Error rates of causality orientation for pruned CCFARM output

Datasets	$q=1$	$q=2$	$q=3$	$q=4$	$q=5$	$q=6$	$q=7$	$q=8$	$q=9$	$q=10$
$D1$- $(p=1)$	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
$D2$- $(p=2)$	0.00	0.00	0.00	0.00	0.00	0.00	0.07	0.06	0.11	0.15
$D3$- $(p=3)$	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
$D4$- $(p=4)$	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.097	0.21	0.31
$D5$- $(p=5)$	0.00	0.00	0.00	0.00	0.28	0.7	0.34	0.48	0.53	0.72

Table 6.3: Error rates of causality orientation for original archive of CCFARM

Datasets	$q=1$	$q=2$	$q=3$	$q=4$	$q=5$	$q=6$	$q=7$	$q=8$	$q=9$	$q=10$
$D1$- $(p=1)$	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
$D2$- $(p=2)$	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
$D3$- $(p=3)$	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
$D4$- $(p=4)$	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.03	0.00	0.00
$D5$- $(p=5)$	0.00	0.00	0.00	0.00	0.08	0.03	0.00	0.00	0.00	0.02

relations. In contrast to Table 6.2, the error rates decrease significantly, especially in the 10 datasets with fifth-order hidden relations; for example, they drop from 0.7 to 0.03 in dataset D5-6 and from 0.72 to 0.02 in dataset D5-10. This suggests that pruning schemes need to be selected carefully according to task requirements.

Although the error rate decreases when a different FAR candidate set is used, Table 6.2 shows that some individual dataset outcomes still have errors (e.g., D5-5, D5-6, D5-10). As the causal relations search in Algorithm 12 is exhaustive, the reason for some datasets still showing error rates higher than 0 lies in the FAR candidates supplied by CCFARM which, in essence, adopts an evolutionary algorithm (EA) for its rule searching. In other words, the FAR search is based on heuristics, hence, it is not expected that a complete set of FARs hidden in a dataset will be extracted. Therefore, in some cases, CCFARM could, provide FARs that do not cover all the genuine relations in the underlying mechanism.

6.3.2 Experimental Design for Hierarchical Hidden Relations

The synthetic datasets for chained hidden relations are generated based on the datasets for separated hidden relations. Each one is generated in the following way and has 20 variables. The first 10 variables ($[x_1, x_2, ..., x_{10}]$) are used as independent variable candidates for generating relations for $[x_{11}, x_{12}, ..., x_{15}]$. Then, these 5 variables become independent variable candidates for generating relations for the remaining higher-level variables; for example, supposing that the dataset currently being generated has two higher-level variables (x_{16}, x_{17}), the functions generating them could take the form:

$$x_{16} = x_{12} + x_{14}^2 + x_{15}^3 + 1$$
$$x_{17} = x_{11} + x_{13}^2 + x_{13}^3 + 1$$

The independent variables for each function are sampled from $[x_{11}, x_{12}, ..., x_{15}]$. In order to present chained relations, these candidate variables are also generated from polynomial functions such as those presented below:

$$x_{11} = x_2 + x_9^2 + x_6^3 + 1$$
$$x_{12} = x_4 + x_1^2 + x_8^3 + 1$$
$$x_{13} = x_7 + x_5^2 + x_{10}^3 + 1$$
$$x_{14} = x_3 + x_6^2 + x_1^3 + 1$$
$$x_{15} = x_7 + x_4^2 + x_5^3 + 1$$

A hierarchical dataset is denoted as $Hp - q$, where p indicates the order of its polynomial function and q the number of its higher-level functions. The above example dataset is denoted as $H3-2$, where 3 indicates that the hidden polynomial functions are of the third order and 2 that there are two higher-level functions in the dataset. Figure 6.3 illustrates the coding scheme for such underlying relations. Hierarchical hidden relations increase the order of hidden relations and the level of overlapping among hidden relations. In the previous chapter, we saw the influence

Table 6.4: Error rates of causality orientation for datasets with hierarchical relations

Data	$q = 1$	$q = 2$	$q = 3$	$q = 4$	$q = 5$
$H1$- $(p = 1)$	0.00	0.00	0.00	0.00	0.00
$H2$- $(p = 2)$	0.00	0.00	0.00	0.00	0.00
$H3$- $(p = 3)$	0.00	0.00	0.00	0.00	0.00
$H4$- $(p = 4)$	0.00	0.00	0.00	0.00	0.00
$H5$- $(p = 5)$	0.1 (3/30)	0.17(6/35)	0.15(6/40)	0.09(4/45)	0.16 (8/50)

of hidden relation overlapping on the performance of CCFARM. With hierarchical hidden relations, we can further investigate whether overlapping can influence of the performance of the proposed experimental causal search algorithm.

The experimental results for the error rate are shown in Table 6.4 which indicates that most of the synthetic datasets have average low error rates. However, datasets with hidden functions reaching the fifth order, have relatively higher error rates than the other datasets. As this could be attributed to the lack of sufficient FARs to cover the direct causal links within a layer, the FARs are checked against the underlying functions. The comparison criterion is whether the FARs found from lower-level hypothesis generations include the causal links designed in the synthetic datasets. The results shown in Table 6.4 indicate that, in the current experiment, the error rate is caused by the quality of coverage provided by the input FARs.

6.4 Experiments with Play-board Context

One component that the above analysis of causal searching on synthetic datasets is missing is 'context', which is at the heart of a causal search. The context provides regularities and constraints that allow causal relations to materialise and give meaning for further interpretations. The overall process of hypothesis generation enables an agent to observe, learn about, and use reverse engineering for controlling its environment. The experiments presented in the following subsections instantiate an agent that uses CCFARM and an experimental causal search algorithm in order

to learn how to retrieve the underlying causal structure of a coloured puzzle.

6.4.1 Experimental Design for Reverse Engineering Game

The game is a coloured control play-board, as shown in Figure 6.4. The rainbow colours in the upper left box possess a set of underlying mutually influential relations. Each coloured stripe has a corresponding sliding controller in the upper right box that allows for external intervention on its stripe. The underlying influence mechanism behind the play-board is shown in Figure 6.5. The play-board has two main states: active and stable. During the active state, the stripes change colour by their own underlying mechanisms which allows the learning agent to observe and record the values of the colours. The colour variables in the lower left box in Figure 6.5 ($C = \{c_1, c_2, c_3, ..., c_i, ..., c_7\}$) refer to the coloured stripes. The underlying relations of variables c_3, c_5 and c_7 are defined in Equations 6.3, 6.4, 6.5 and the values of variables $\{c_1, c_2, c_4, c_6\}$ are all in the range of $[-5, 5]$.

The arrows in Figure 6.5 define the direction of influence. In any time step, if any change happens to a colour (c_i), its influence materialises over all the mechanisms in the network after time step Δt. This assumes that the observation time step is sufficiently long for all effects to materialise, for example, when c_4 changes its colour in the play-board, by the end of time step Δt, all its effects (i.e., c_3 and c_5) are materialised.

In the stable state, the controls are connected to the colour stripes, which allow an agent to experiment with the colour stripes according to its observations and knowledge (e.g. FARs). The notations, $\{k_1, k_2, k_3, ..., k_i, ..., k_7\}$ in Figure 6.5, are used to represent the control knobs. The dashed lines in Figure 6.5 mean that the controls do not exert influence on the colour stripes when the play board is in the active state. When the agent has collected a certain number of observations forming dataset $\mathbf{X}$ (500 instances in the current experiments), it moves to the CCFARM process to search for interesting associations. This association search is followed by the proposed causal search procedure, during which the agent interacts with the play

board to examine potential causal relations. For example, suppose one of the outputs of CCFARM is a FAR ($c_1 \rightarrow c_2$). The agent will investigate whether c_1 causes c_2 to change. It first sorts the dataset $\mathbf{X}$ according to the values of c_1 in ascending order. Then, it selects 30 instances from the sorted dataset ($\mathbf{X'_{c_1}}$) with equal intervals (in this experiment it selects one instance every 16 instances for covering the range of c_1). This selection forms an experimental dataset D_s. For each instance in D_s, $D_{s,j}$ ($j = 1, 2, 3, ..., 30$), the agent adjusts the values of the colour stripes to those stored in $D_{s,j}$ by moving the knobs. It moves the knob connected to c_1 to adjust its value from ($c_{1,j}$) to ($c_{1,j} + \Delta c_1$). The values of the rest of the colour stripes are adjusted to the same values as in $D_{s,j}$ except c_2. The values of $\{c_1, c_2, c_3, ..., c_i, ..., c_7\}$ after the above manipulation are recorded to form an instance $D'_{s,j}$. This process is iterated over each instance in D_s to create its counterfactual set D'_s. When all 30 instances in D_s are examined with the above procedure, the values of c_2 in D_s and D'_s are analysed to determine whether c_1 is its cause (as in Algorithm 12 Line 15). When all the FARs have been examined, the final output is given in a causal network form.

Algorithm 13: Process flow for game control learning agent

1 Record coloured stripe values of play-board every Δt time to form observational data (D)
2 Apply CCFARM on D to extract FARs
3 Use experimental causal searching algorithm in Algorithm 12 to determine casual relations
4 Output derived causal network

The functions that constitute this underlying control mechanism is list in the following equations.

Color c_3:

$$c_3 = sin(c_2 \times c_2) + c_4; \tag{6.3}$$

Color c_5:

$$c_5 = c_3 \times c_3 + c_4; \tag{6.4}$$

Color c_7

$$c_7 = e^{c_1} \times sin(c_1) + c_6; \tag{6.5}$$

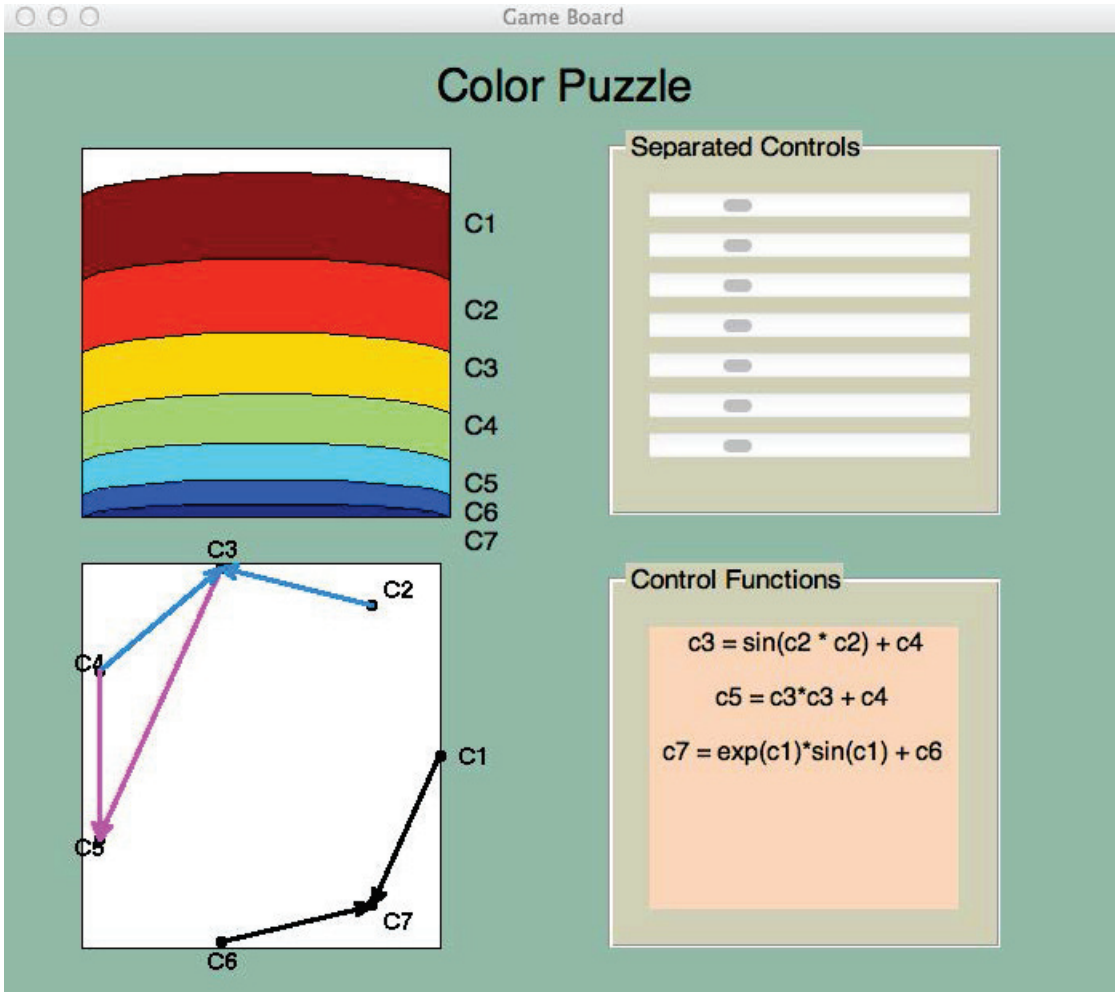


Figure 6.4: Colour control environment for a causal game

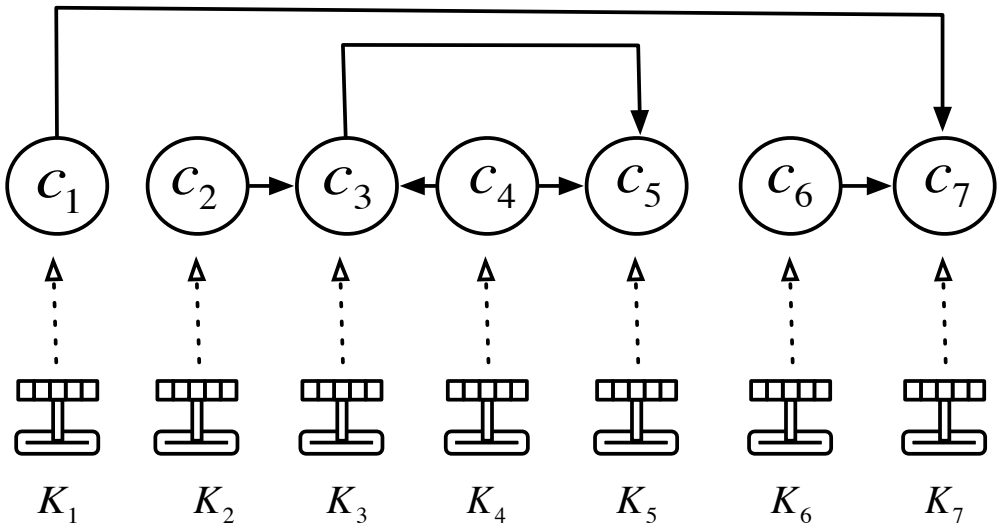


Figure 6.5: Illustration of the causal game

In summary, this play-board environment provides the context for a causal search, possesses continuous underlying relations and has a complex influencing mechanism that challenges conventional causal modelling. The parameters used in this experiment are shown in Table 6.5.

Table 6.5: List of parameters used in play-board experiments

Parameters	Symbol	Value
Rule population size	n_r	30
ANN population size	n_a	14
Generation size	n_g	50
Crossover rate	R_c	0.8
Mutation rate	R_m	0.1
Accuracy threshold	h_r	0.8
Learning rate	lr	0.1
No. epochs	n_e	500
No. nodes (hidden)	n_u	10

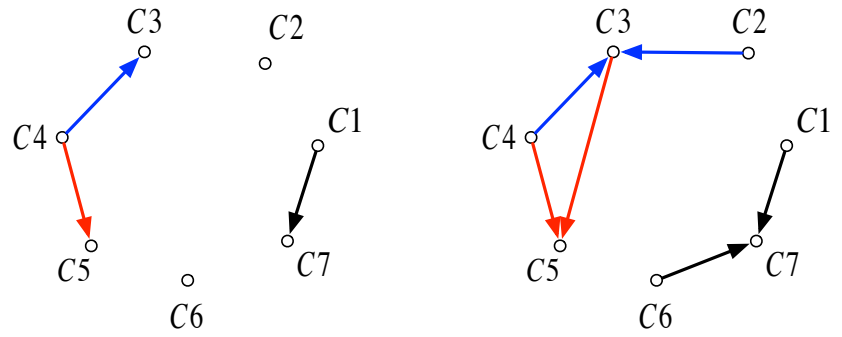
6.4.2 Experiment Results for the Play-board Game

The experiments on the play-board game reflect other aspects that can influence the performance of the experimental causal search algorithm. The play-board experiments use 30 different seeds for the FARM step. After applying the experimental causal search algorithm, we find that the error rate for retrieving the causal structure is 0.5.

The causal structure retrieved by the algorithm is shown in Figure 6.6, together with the original causal structure. Each causal relation in Figure 6.6(b) is marked by a different colour. There are three underlying causal relations in total corresponding to Equations 6.3, 6.4, and 6.5. From Figure 6.6 (a), each underlying relation has a missing link compared with the original relations. The reasons for the present missing links can be categorised as two types: from the coverage of the FARs; and from testing accuracy.

The two missing links ($c_2 \rightarrow c_3$ and $c_6 \rightarrow c_7$) are caused by the coverage of the FARs, that is, for the first, after checking the FARs on which the experimental causal search algorithm is based, there is no rule for its RHS to be c_3 and, at the same time, for its LHS variables to include c_2. The same situation applies to the missing link $c_6 \rightarrow c_7$. The coverage of the FARs affects the quality of the experimental causal search results.

The missing link $c_3 \rightarrow c_5$ is a result of the difference test of the experimental causal search algorithm. The FARs in the input to the search algorithm include this $c_3 \rightarrow c_5$ association but the difference test (i.e., paired t-test incorporated in Algorithm 12) in the algorithm does not identify the difference of c_5 between before and after applying interventions on variable c_3. Table 6.6 shows examples of FARs including the link $c_3 \rightarrow c_5$ in different seeds, and according to the prior knowledge of Equation 6.4, we can tell that applying interventions on c_3 will cause the values of c_5 to change. However, the missing link in the final output (i.e., Figure 12 (a)) indicates that the difference test fails to confirm the change of c_5 from the intervention. This situation suggests another factor that affects the performance of the experimental



(a) retrieved causal structure (b) underlying causal structure

Figure 6.6: Comparison of causal structure retrieved from experimental causal search algorithm and underlying causal structure

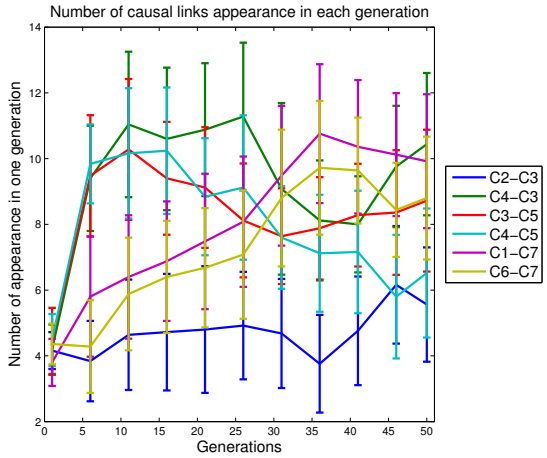
causal search algorithm, that is, the performance of the difference test used for comparison.

Table 6.6: Examples of FARs including causal link $c_3 \rightarrow c_5$ in different seeds

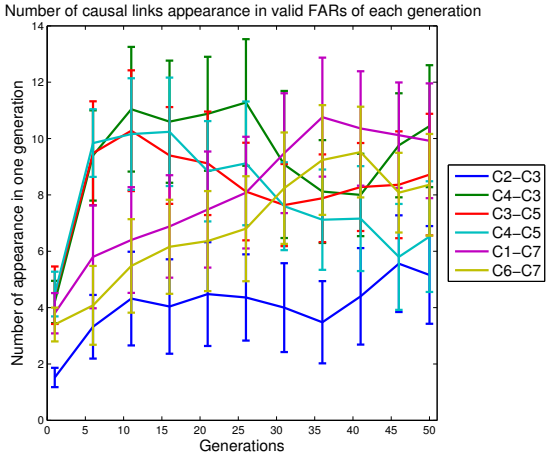
FAR	Predictive accuracy
$f(x_3, x_4) \Rightarrow x_5$	0.987 (Seed 2)
$f(x_3) \Rightarrow x_5$	0.992 (Seed 1)
$f(x_3, x_4) \Rightarrow x_5$	0.990 (Seed 14)
$f(x_3, x_4) \Rightarrow x_5$	0.990 (Seed 15)

Table 6.7 shows the percentages of causal links found in all 30 seeds. We checked the FARs in the experimental outputs using different seeds, to see whether all the causal links are included. It can be seen that $c_2 \rightarrow c_3$ and $c_6 \rightarrow c_7$ are missing from all experiments regardless of seed. This result reveals that in the final output of CCFARM, there are no associative relations specifying the links (c_2, c_3) and (c_6, c_7). This raises a question of whether it is CCFARM that cannot identify associative relations. Therefore, we record the number of causal links in the valid and archived FARs in each generation in different steps.

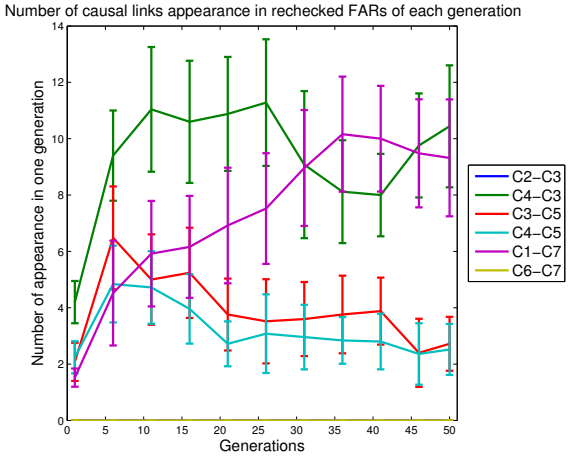
The results are shown in Figure 6.7 in which it can be seen that, in the first two plots, all six causal links are covered in each generation. However, after the



(a) number of causal links in the population



(b) number of causal links in the valid rules of each population



(c) number of causal links in achieved rules of each population

Figure 6.7: The number of the causal links in each generation at different step with 95% confidence interval

Table 6.7: Percentage of causal links for 30 seeds

Causal links	Percentage
$C2 - C3$	0
$C4 - C3$	0.9
$C3 - C5$	0.8
$C4 - C5$	0.9
$C1 - C7$	0.95
$C6 - C7$	0

archiving process, two links disappear from the FARs. The archival process includes a variable selection procedure (presented in Section 5.2.5 and Section 4.1.2.2) which deletes a variable from a given valid FAR if, without that variable, the FAR still has a predictive accuracy higher than a threshold ($h_r - \Delta a$). These experimental results reveal that, for real world systems, for the purpose of providing input for causal hypothesis generation, the archive process can be skipped to preserve more potential causal links. However, if CCFARM is applied to conventional quantitative ARM tasks, where the main goal is to extract concise association rules, it is still necessary to conduct the variable selection procedure in the mining process.

6.5 Chapter Summary

In this chapter, an experimental causal search algorithm for CHGP was proposed. It relaxes the assumptions currently made about continuous variables when their causal relations are under investigation. It completes the final component of the methodology for the general hierarchical hypothesis generation problem presented in this thesis. Experiments on synthetic data showed the relatively high accuracy of retrieving the underlying causal relations as an approach for automatic hypothesis generation.

On the other hand, the play-board game revealed two factors identified as af-

fecting the performance of the experimental causal search algorithm: the coverage of the FARM results; and the quality of the statistical test of the intervention results. Further analysis of the causal links revealed that the coverage of the FAR could be affected by the archiving step in the CCFARM algorithm. This suggested an option for improving CCFARM to increase its coverage of the associative relations hidden in observational data by FARs could be to skip the variable selection step during archiving of the FARs found.

The advantage of the proposed experimental causal search algorithm was that the causal influences identified in the results were consistent with the original causal relations. In addition, due to its experimental nature, the causal hypothesis generation was not sensitive to hidden variables that influenced the measured variables. Identifying the factors that affected the performance of the experimental causal search algorithm presented the possibility of improving the algorithm, as discussed in the next chapter.

Chapter 7

Conclusion and Future Work

The advances in data collection, transmission and storage have given rise to hypothesis generation research which uses data mining and machine learning techniques to automatically find patterns of interest in datasets. A pattern representation is usually defined by the researcher, with the techniques involving fitting it to observed data [52]. This, provides a paradigm for automatic knowledge discovery with limited involvement of human experts. This thesis explored its potential extension to knowledge discovery for a general system, the underlying structure of which is unknown, measured by a set of continuous variables. The primary contributions of this thesis are:

- defining a generalised hypothesis generation problem;
- decomposing a problem into an associative hypothesis generation problem (AHGP) and a causal hypothesis generation problem (CHGP);
- developing functional association rules (FARs) and FAR mining (FARM) algorithms as solutions to the AHGP;
- developing an experimental causal search algorithm based on an agent architecture as a solution to the CHGP; and
- using new metrics that enable evaluations and visualisations of the perfor-

mances of the hypotheses generated and comparisons of different algorithms..

Section 7.1 provides a summary of the research contributions made by this thesis and analyses the conclusions drawn in previous chapters. Section 7.2 discusses the limitations of this work and considers possible avenues for future development and application of the proposed autonomous hypothesis generation approach.

7.1 Summary of Research Contributions and Conclusions

This work contributes to knowledge discovery in a generalised hypothesis generation scenario where an unknown system can be measured by a set of continuous variables. *A priori* knowledge of the underlying structure of the system is not available in advance and the causal structure that dominates its dynamics is automatically explored. The contributions of this research are detailed below.

7.1.1 Generalised Hypothesis Generation Problem Definition and Decomposition

A new problem definition for hypothesis generation in continuous domains was proposed. It focused on situations in which a general unknown system was measured by a set of continuous variables and aimed to provide initial insights into the structure of the underlying system. Compared to conventional hypothesis generation practice, it avoids bias introduced by the limitation of human domain knowledge in the manual design of hypotheses. Our hypothesis generation problem was further decomposed into two sub-problems, the AHGP and the CHGP.

7.1.2 Development Solutions to Proposed Hypothesis Generation Problems

This category of contributions had the following components.

- Two FAR representations based on the regression model and ANNs. They addressed the issue of representing associative hypotheses for the first sub-problem, and could process the linear and non-linear relations existing among the variables respectively. They do not require conversion of variables into intervals as in the conventional association rule form. The definition of FARs introduces a new AR definition to the field of ARM.

- Three evolutionary algorithm (EA)-based search approaches for linear FAR mining (LFARM). They provided solutions to the problem of how to identify LFARs from observational data. The associative hypothesis generation problem was cast as a heuristic search based on the characteristics of the proposed associative hypothesis representation.

- A cooperative co-evolutionary algorithm for mining general FARs. CCFARM provides another solution to the problem of associative hypothesis generation when the representation of a hypothesis is concerned with general associative relations. Also, it decomposes FARM into two search problems: one searches for the valid associative hypothesis; and the other for the appropriate ANN initialisation that is likely to collaborate best with the potentially valid associative hypothesis. Comparison experiments with two state-of-the-art ARM algorithms have shown its competitive performance.

- An experimental causal search algorithm for causal hypothesis generation based on FARs. FARs places the variables into sub-sets in which the variables within one sub-set are interdependent. Causal hypotheses are built on such variable sets. We presented this algorithm as an agent architecture which systematically applies interventions on the interrelated variables and, depending on the consequences of these interventions, can establish the causal structure

of the system. The algorithm does not require the conventional assumptions that the underlying structure of the unknown system is limited to causal relations, or that the dependency patterns in observational data are only due to causal relations as in conventional automatic causal modelling.

7.1.3 Performance Metrics for the Hypothesis Generation Approaches in Continuous Domains

The metrics proposed and experiments conducted are summarised as follows.

- New metrics that enabled evaluations and visualisations of the performances of the hypotheses generated and comparisons of different algorithms were proposed. They assisted in evaluating the characteristics of the generated hypotheses. For LFARM, to compare the performances of different LFARM algorithms, complexity and perceptual selectivity were proposed. For general FARM, to evaluate and visualise the quality of the general FARs, matching, frequency and active ratio were used.

- Empirical experiments were conducted to study the performances of the designed solutions. Empirical comparisons of the proposed LFARM algorithms using complexity and perceptual selectivity suggested that there was no significant difference among the different EA-based LFARM approaches. Experiments on CCFARM indicated similarities between the FARs mined and the hidden associative relations in the observational data. The main factor that affected performance was identified as the overlapping of the underlying relations. Experiments on the causal hypothesis generation problem showed a low error rate for determining underlying causal relations. Also, the experiments showed that the error rate was influenced by three factors: the coverage of the FARs; the difference test; and the variable selection step in CCFARM.

7.2 Limitations of This Study and Suggestions for Future Work

Hypothesis generation as a complementary tool for knowledge discovery has become popular in a number of research fields conventionally dominated by hypothesis testing research. This thesis has begun to explore its potential extension to situations in which domain knowledge is not sufficiently specific for the precise design of a domain-specific knowledge representation and generation strategy. The problem and solutions proposed in this thesis provide a basis for further research in a number of different directions.

7.2.1 Computational Efficiency and Additional Experiments

An issue identified by the experiments using FARM was computational cost. The collaboration scheme adopted by CCFARM was a complete mixing, in which two populations evolved in parallel and each individual was paired up with every individual in the other sub-population. Although this scheme was used to handle the epistasis in co-evolution, the computational power required would become an issue when the number of variables involved increases. Other collaboration schemes could be investigated and compared with this scheme in terms of the performance of FARM; for example, the search process could evolve in a round-robin fashion [180], that is, one sub-population being evolved while the other remains fixed.

A second potential extension on the FARM algorithm is that niching techniques can be added to the mining procedure. Unlike the conventional association rules, where the Apriori property can guarantee all possible frequent patterns extracted, functional association rules rely on heuristic search to find as many rules as possible. Niching methods extend evolutionary computation approaches to domains that require the location and maintenance of multiple solutions [115]. Different niching techniques can be implemented with the FARM algorithms for the purposes of increasing the diversity of the solutions and the coverage of the mining process.

7.2.2 Interrelation of the Functional Association Rule Mining and Experimental Causal Search

The experiments for causal hypothesis generation identified that the quality of the FARs essentially affected the error rate of causal discovery. Although FARs can be seen as a necessary condition of causal relations, it is not a sufficient condition and not all associative relations are due to potential causal relations. It would benefit an experimental causal search if a FARM algorithm could exclude most FARs unrelated to potential causal relations. This can maybe achieved using the advantages of evolutionary computation by including additional criteria in the objective functions of the EA and treating the problem as multi-objective.

7.2.3 Alternative Applications of Hypothesis Generation

The focus of the hypothesis generation problem in this thesis was on a general unknown system and the solution presented provided an insight into basic approaches that could be adopted automatically. The paradigm introduced could be incorporated into an intelligent agent architecture as part of an approach for assisting it to understand its environment. Based on this, the agent could adapt to different scenarios without requiring hard-coded information of their environments. Another potential application could be in cyber security, where new activity patterns could be hypothesised without the need to rely on old knowledge.

7.2.4 Experiments Examining Hypothesis Generation Using Datasets With Different Characteristics

Our experiments in Chaps. 4-6. have shown promising results for hypothesis generation from a number of well-known datasets. However, we have only considered a relatively small number of datasets in this thesis. There may be datasets with certain characteristics for which our algorithms perform differently according to the metrics presented in this thesis. Another area of future work is thus to widen the

types and characteristics of datasets studied to understand the strengths and weaknesses of our algorithm in response to data with different characteristic patterns and trends.

7.3 Concluding Remarks

Hypothesis generation provides a complementary tool for automating the knowledge discovery process. In general, it requires researchers to formulate the necessary knowledge representation and generation strategy with a certain amount of domain knowledge. Then, with the assistance of machine learning and data mining techniques, the patterns that encompass the knowledge of interest can be automatically extracted. This thesis has begun to explore the potential extension of hypothesis generation to situations in which the available domain knowledge is not sufficiently specific for the precise design of its representation and generation strategy. The problem and solution development it proposed provide a basis for further research in several different directions.

Bibliography

[1] H. A. Abbass. An evolutionary artificial neuralnetworks approach for breast cancer diagnosis. *Artificial Intelligence in Medicine*, 25(3):265–281, 2002.

[2] H. A. Abbass and E. Petraki. The causes for no causation: A computational perspective. *Information Knowledge Systems Management*, 2011.

[3] H. Abdi. Multiple correlation coefficient. *Encyclopedia of Measurement and Statistics*, pages 648–651, 2007.

[4] D. H. Ackley, G.E. Hinton, and T. J. Sejnowski. A learning algorithm for boltzmann machines. *Cognitive science*, 9:147–169, 1985.

[5] C. Aggarwal and P. Yu. A new framework for itemset generation. In *Proceedings of the 1998 ACM symposium Principles of Database Systems*, pages 18–24, 1998.

[6] R. Agrawal, T. Imielinski, and A. Swami. Mining association rules between sets of items in large databases. In *Proceedings of ACM-SIGMOD international on management of data*, pages 207–216, 1993.

[7] R. Agrawal and R. Srikant. Fast algorithms for mining association rules in large databases. In *Proceedings of 20th International Conference on Very Large Data Bases*, volume 1215, pages 487–499, 1994.

[8] Bilal Alatas, Erhan Akin, and Ali Karci. Modenar: Multi-objective differential evolution algorithm for mining numeric association rules. *Applied Soft Computing*, 8:646–656, 2008.

[9] G. An. The effects of adding noise during backpropagation training on a generalization performance. *Neural computation*, 8, 1996.

[10] I. P. Androulakis. From data to models: General framework for data-driven hypothesis generation, 2007.

[11] P. J. Angeline and J. B. Pollack. Competitive environments evolve better solutions for complex tasks. In *Proceedings of the 5th International Conference on Genetic Algorithms*, pages 264–270, 1993.

[12] C.F. Auerbach and L. B. Silverstein. *Qualitative data: an introduction to coding and analysis*. New York University Press, 2003.

[13] Y. Aumann and Y. Lindell. A statistical theory for quantitative association rules. *Journal of Intelligent Information System*, pages 255–283, 2003.

[14] K. Bache and M. Lichman. Uci machine learning repository, http://archive.ics.uci.edu/ml.

[15] S. Baker. Analyisis of survival data from a randomised trial with all-or-none compliance: Estimating the cost-effectiveness of a cancer screening program. *Journal of the American Statitical Association*, 93:929–934, 1998.

[16] J. Balcazar. Closure-based confidence boost in association rules. *Journal of Machine Learning Research*, 11:74–80, 2008.

[17] S. Baluja. Population-based incremental learning: a method for integrating genetic search based function optimization and competitive learning. Technical report, Carnegie Mellon University, 1994.

[18] A. R. Barron. Complexity regularization with application to artificial neural networks. In *Nonparametric functional estimation and related topics*. Springer, 1991.

[19] A. R. Barron. Neural net approximation. In *Proceedings of the 7th Yale workshop on adpative and learning systems*, pages 69–72, 1992.

[20] P. Bartlett and T. Downs. Training a neural network with a genetic algorithm. Technical report, University of Queensland, 1990.

[21] J. Baxter. The evolution of learning algorithms for artificial neural network. *Complex Systems*, pages 313–326, 1992.

[22] R. K. Belew, J. McInerney, and N. N. Schraudolph. Evolving networks: using genetic algorithm with connectionist learning. Technical report, University of California, 1991.

[23] S. Bengio, Y begio, J. Cloutier, and J. Gecsei. On the optimization of a synaptic learning rule. In *Proceedings of a synaptic learning rule*, 1992.

[24] Y. Bengio and S. Bengio. Learning a synaptic learning rule. Technical report, University of Montreal, 1990.

[25] Y. Bengio and Y. LeCun. Scaling learning algorithms towards ai. In *Large scale kernel machines*. MIT Press, 2007.

[26] E. Bengoetxea, P. Larranaga, I. Bloch, and A. Perchant. Estimation of distribution algorithms: a new evolutionary computation approach for graph matching problem. *Lecture notes in computer science*, 2134:454–469, 2001.

[27] E. Bernado-Mansilla and J. M. Garrell-Guiu. Accuracy-based learning classifier system: models, analysis and applications to classification tasks. *Evolutionary computation*, 11(3):209–238, 2003.

[28] M. Bichsel and P. Seitz. Minimum class entropy: a maximum information approach to layered networks. *Neural networks*, 2:133–141, 1989.

[29] L. Bloch, C. M. Haase, and R. W. Robert. Emotion regulation predicts marital satisfaction: more than a wives' tale. *Emotion*, 4, 2013.

[30] S. Bornholdt and D. Graudenz. General asymmetric neural networks and structure design by genetic algorithms. *Neural networks*, 5(2):327–334, 1992.

[31] Y. Boureau and Y. L. Cun. Sparse feature learning for deep belief networks. In *Advances in neural information processing system*, pages 1185–1192, 2008.

[32] S. Brin, R. Motwani, J. Ullman, and S. Tsur. Dynamic itemset counting and implication rules for market basket data. In *ACM SIGMOD Record*, volume 26, pages 255–264, 1997.

[33] R. Cai, A. Tung, Z. Zhang, and Z. Hao. What is unequal among the equals? ranking equivalent rules from gene expression data. *IEEE Transactions on Knowledge and Data Engineering*, 20(11):1735–1748, 2011.

[34] T. P. Caudell and C. P. Dolan. Parametric connectivity: training of contrained networks using genetic algorithms. In *Proceedings of the 3rd International conference genetic algorithms and their applications*, pages 370–374, 1989.

[35] D. J. Chalmers. The evolution of learning: an experiment in genetic connectionism. In *Proceedings of the 1990 connectionist models summer school*, pages 81–90, 1990.

[36] W. Chen, T. Weise, Z. Yang, and K. Tang. Large-scale global optimization using cooperative coevolution with variable interaction learning. In *Proceedings of the 11th international conference on parallel problem solving from nature: Part II*, pages 300–309, 2010.

[37] D. L. Chester. Why two hidden layers are better than one. In *International joint conference on neural networks*, volume 1, pages 265–268, 1990.

[38] D. M. Chickering. Optimal structure identification with greedy search. *Journal of machine learning research*, pages 507–554, 2002.

[39] T. J. Chin, J. Yu, and D. Suter. Accelerated hypothesis generation for multi-structure robust fitting. *IEEE Transaction on Pattern Analysis and Machine Learning*, 99:1–15, 2011.

[40] D. K.Y. Chiu and T. W.H. Lui. Nhop: A nested associative pattern for analysis of consensus sequence ensembles. *IEEE Transactions on Knowledge and Data Engineering*, 25(10):2314–2324, 2012.

[41] G. F. Cooper and E. Herskovitz. A bayesian method for the induction of probabilistic networks from data machine learning. *Machine Learning*, 9:309–347, 1992.

[42] J. Corander, M. Ekdahl, and T. Koski. Parallel interacting mcmc for learning topologies of graphical models. *Data mining and knowledge discovery*, 17:431–456, 2008.

[43] G. Cybenko. Continuous valued neural networks with two hidden layers are sufficient. Technical report, University of Illinois at Urbana-Champaign, 1988.

[44] G. Cybenko. Approximation by superpositions of a singmoidal function. *Mathematics of control, signals, and systems*, 2:303–314, 1989.

[45] H. H. Dam, H. A. Abbass, and C. Lokan. Dxcs: an xcs system for distributed data mining. In *Proceedings of 2005 conference on genetic and evolutionary computation*, pages 1883–1890, 2005.

[46] P. J. Darwen and X. Yao. Speciation as automatic categorical modularization. *IEEE Transactions on Evolutionary Computation*, 1(2):101–108, 1997.

[47] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan. A fast and elitist multiobjective genetic algorithm: Nsga-ii. *IEEE Transactions on Evolutionary Computation*, 6(2):182–197, 2002.

[48] N. Dodd, D. Macfarlane, and C. Marland. Optimization of artificial neural network structure using genetic techniques implemented on multiple transputers. In *Proceedings of transputing 1991*, pages 687–700, 1991.

[49] B. Dunkel and N. Soparkar. Data organization and access for efficient data mining. In *Proceedings of 15th International conference on data engineering*, pages 522–529, 1999.

[50] J. Evans and A. Rzhetsky. Machine science. *Science*, 329:399–400, 2010.

[51] S. Fahlman and C. Lebiere. The cascade-correlation learning architecture. Technical report, Carnegie Mellon University, 1990.

[52] U. Fayyad, G. Piatetsky-Shapiro, and P. Smyth. From data mining to knowledge discovery in databases. *AI Magazine*, pages 37–55, 1996.

[53] D. B. Fogel. An information criterion for optimal neural network selection. *IEEE Transactions on Neural Networks*, 2:490–497, 1991.

[54] D. B. Fogel, L. J. Fogel, and V. W. Porto. Evolving neural neworks. *Biological cybernectics*, 63(6):487–493, 1990.

[55] D. B. Fogel, E. C. Wasson, and E. M. Boughton. Evolving neural networks for detecting breast cancer. *Cancer letter*, 96(1):49–53, 1995.

[56] D. B. Fogel, E. C. Wasson, and V. W. Porto. A step toward computer-assisted mammography using evolutionary. *Cancer letter*, 119(1):93–97, 1997.

[57] L. Fogel, A. Owens, and M. Walsh. *Artificial intelligence through simulated evolution*. John Wiley & Sons, 1966.

[58] L. N. Foner and P. Maes. Paying attention to what is important: using focus attention to improve unsurpervised learning. In *Proceedings of the 3rd international conference on the simulation of adaptive behaviour*, pages 1–20, 1994.

[59] J. F. Fontanari and R. Meir. Evolving a learning algorithm for the binary perceptron. *Network*, 2(4):353–359, 1991.

[60] N. Friedman, I. Nachman, and D. Pe'er. Learning bayesian network structure from massive datasets: the sparse candidate algorithm. In *Proceedings of 16th conference on uncertainty in artificial intelligence*, pages 196–205, 1999.

[61] Y. Fu and J. Han. Meta-rule guided mining of association rules in relational databases. In *Proceedings of International workshop integration of knowledge discovery with deductive and object oriented databases*, pages 39–46, 1995.

[62] T. Fukuda, Y. Morimoto, and S. Morishita. Data mining using two dimensional optimized association rules: scheme algorithms and visualization. In *Proceedings of the 1996 ACM-SIGMOD international conference management of data*, pages 13–23, 1996.

[63] K. Funahashi. On the approximate realization of continuous mappign by neural networks. *Neural networks*, 2(183-192), 1989.

[64] G. Gardarin, P. Pucheral, and F. Wu. Bitmap based algorithms for mining association rules. In *Proceedings of the 14th Bases de Donnes Avances (BDA 98)*, pages 157–176, 1998.

[65] S. Geman, E. Bienenstock, and R. Doursat. Neural networks and the bias/variance dilemma. *Neural computation*, 4, 1992.

[66] D. E. Goldberg. *Genetic algorithms in search, optimization, and machine learning*. Addison-Wesley, 1989.

[67] I. Goodfellow, Q. Le, A. Saxe, and N. Ng. Measuring invariances in deep networks. In *Advances in neural information processing system*, pages 646–654, 2009.

[68] G. Grahne and J. Zhu. Efficiently using prefix-trees in mining frequent itemsets. In *Proceedings of the IEEE ICDM workshop on frequent itemset mining implementations*, volume 90, pages 110–121, 2003.

[69] H. Altay Guvenir and I. Uysal. Functional approximation repository.

[70] J. Han and Y. Fu. Discovery of multiple-level association rules from large databases. In *Proceedings of 1995 international conference on very large data bases*, pages 420–431, 1995.

[71] J. Han, M. Kamber, and J. Pei. *Data mining: concepts and techniques*. Morgan Kaufmann, 2006.

[72] J. Han, J. Pei, and Y. Yin. Mining frequent patterns without candidate generation. *SIGMOD Record*, 29(2):1–12, 2000.

[73] S. A. Harp, T. Aamad, and A. Guha. Toward the genetic systhesis of neural networks. In *Proceedings of the 3rd International conference genetic algorithms and their applications*, pages 360–369, 1989.

[74] S. A. Harp, T. Samad, and A. Guha. Designing application specific neural netowrks using the genetic algorithm. In *Advances in neural information processing system*, pages 447–454. Morgan Kaufmann, 1990.

[75] S. S. Haykin. *Neural networks: a comprehensive foundation*. Prentice hall international, 1999.

[76] D. O. Hebb. *The organization of Behavior*. Taylor and Francis, 1949.

[77] J. J. Heckman. Causal inference and nonrandom samples. *Journal of Educational Statstics*, 14:159–168, 1989.

[78] G. E. Hinton. Training produces of experts by minimizing contrastive divergence. *Neural Computation*, 14(8):1711–1800, 2002.

[79] G.E. Hinton and R.R. Slalkhutdinov. Reducing the dimensionality of data with neural networks. *Science*, 313(5786):504–507, 2006.

[80] J. Holland and J. Reitman. Cognitive systems based on adaptive agents. In *Pattern-directed inference systems*. 1978.

[81] J. H. Holland. *Adaption in natural and artificial systems: an introductory analysis with applications to biology, control and artificial intelligence*. University of Michigan Press, 1975.

[82] J. H. Holland. *Escaping brittleness: the possibilities of general purpose learning algorithms applied to parallel rule-based systems*, volume 2. Morgan Kaufmann, 1986.

[83] P. Holland. Statistics and causal inference. *Journal of the American Statitical Association*, 81:945–970, 1986.

[84] J. H. Holmes, P. L. Lanzi, W. Stolzmann, and S. W. Wilson. Learning classifier systems: new models, successful applications. *Information process letters*, 82(1):23–30, 2002.

[85] L. Homostrom and P. Koistinen. Using additive noise in back-propagation training. *IEEE Transactions on Neural Networks*, 3:24–38, 1992.

[86] H. Hornik. Approximation capabilities of multilayer feedforward networks. *Neural networks*, 4(2):251–257, 1991.

[87] K. Hornik. Multilayer feedforward networks are universal approximators. *Neural Networks*, 2:359–366, 1989.

[88] D. Hume. *An enquiry concerning human understanding*, volume Harvard Classics Volume 37. P. F. Collier and Son, 1910.

[89] J. Ilonen, J. Kamarainen, and J. Lampinen. Differential evolution training algorithm for feed-forward neural networks. *Neural Processing Letters*, 17:93–105, 2003.

[90] INCOSE. Systems engineering handbook, version 3. In *Proceedings of international council of systems engineering*, 2006.

[91] N. Jaitly and G. Hinton. Learning a better representation of speech soundwaves using restricted boltzmann machines. In *Proceedings of 2011 IEEE International conference on acoustics, speech and signal proceeding (ICASSP)*, pages 5884–5887, 2011.

[92] D. J. Janson and J. F. Frenzel. Application of genetic algorithms to the training of higher order neural networks. *Journal of System Engineering*, 2:272–276, 1992.

[93] H. Jin, J. Chen, H. He, C. Kelman, D. McAullay, and C. O'Keefe. Signaling potential adverse drug reactions from administrative health databases. *IEEE Transactions on Knowledge and Data Engineering*, 22(6):839 – 853, 2010.

[94] K. A. De Jong. *Evolutionary computation: a unified approach*. The MIT Press, 2001.

[95] K. De Jong. *Analysis of behavior of a class of genetic adaptive system*. PhD thesis, University of Michigan, 1975.

[96] J. F. Hair Jr., W. C. Black, B. J. Babin, and R. E. Anderson. *Multivariate data analysis*. Pearson, 2010.

[97] R. J. Bayardo Jr. Efficiently mining long patterns from databases. In *Proceedings of the 1998 ACM-SIGMOD international conference on management of data*, pages 85–93, 1998.

[98] D. B. Kell and S. G. Oliver. Here is the evidence, now what is the hypothesis? the complementary roles of inductive and hypothesis-driven science in the post genomic era. *BioEssays*, 26:99–105, 2003.

[99] R. D. King, Kenneth E. Whelan, et al. Functional genomic hypothesis generation and experimentation by a robot scientist. *Nature*, pages 247–251, 2004.

[100] H. Kitano. Designing neural network using genetic algorithms with graph generation system. *Complex systems*, 4(4):461–476, 1990.

[101] J. Kivinen and D. Williams. Multiple texture boltzmann machines. In *JMLR Workshop and Conference Proceedings: AISTATS 2012*, volume 22, pages 638–646, 2012.

[102] M. Klemettinen, H. Mannila, and P. Ronkainen. Finding interesting rules from large sets of discovered association rules. In *Proceedings of International conference information and knowledge management*, pages 401–408, 1994.

[103] J. R. Koza and J. P. Rice. Genetic generation of both the weights and architecture for a neural netowrk. In *Proceedings of 1991 IEEE International joint conference of neural networks*, volume 2, pages 397–404, 1991.

[104] L. Lakshmanan, R. Ng, and J. Han. Optimization of constrained frequent set queries with 2-variable constraints. In *Proceedings of International conference management of data*, pages 157–168, 1999.

[105] H. Larochelle, . Bengio, J. Louradour, and P. Lamblin. Exploring strategies for training deep neural networks. *Journal of machine learning research*, 10:1–40, 2009.

[106] H. Larochelle, M. Mandel, R. Pascanu, and Y. Bengio. Learning algorihms for the classification restricted boltzmann machine. *The journal of machine learning research*, 13(1):643–669, 2012.

[107] P. Larranaga and J. A. Lozano. *Estimation of distribution algorithms*. Kluwer academic publisher, 2001.

[108] Y. LeCun, J. S. Kenker, and S. A. Solla. Optimal brain damage. *Advances in Neural Information Processing systems*, 2:598–605, 1990.

[109] H. Lee, C. Ekanadham, and N. Ng. Sparse deep belief net model for visual area v2. In *Proceedings of advances in neural information processing system*, pages 873–888, 2008.

[110] S. W. Lee. Off-line recognition of totally unconstrained hand written numerals using multilayer cluster neural networks. *IEEE Transactions on pattern analysis and machine intelligence*, 18(6):684–652, 1996.

[111] B. Lent, A. Swami, and J. Widom. Clustering association rules. In *Proceedings of the 1997 interantional conference on data engineering*, pages 220–231, 1997.

[112] J. Li and Z. J. Wang. Controlling the false discovery rate of the association/causality structure learned with the pc algorithm. *Journal of machine learning research*, 10:475–514, 2009.

[113] D. I. Lin and M. H. Dunham. Mining association rules: Anti skew algorithms. In *Proceedings of the 14th International Conference on Data Engineering*, pages 486–493, 1998.

[114] D. Madigan, S. A. Andersson, M. D. Perlman, and C. T. Volinsky. Bayesian modl averaging and model selection for markov equivalence classes of acyclic digraphs. *Communications in statistics: theory and methods*, 25(11):2493–2519, 1996.

[115] S. W. Mahfoud. *Niching Methods for Genetic Algorithms*. PhD thesis, University of Illinoise at Urbana-Champaign, 1995.

[116] J. Manyika, M. Chui, B. Brown, J. Bughin, R. Dobbs, C. Roxburgh, and A. Byers. Big data: The next frontier for innovation, competition, and productivity. 2011.

[117] C. Marinica and F. Guillet. Knowledge-based interactive postmining of association rules using ontologies. *IEEE Transactions on Knowledge and Data Engineering*, 22(6):784–798, 2010.

[118] L. Marti. Genetically generated neural networks i: representation effects. In *Proceedings of 1992 International joint conference on neural networks*, volume 537-542, 1992.

[119] J. Mata, J. L. Alvarez, and J. C. Riquelme. An evolutionary algorithm to discover numeric association rules. In *Proceedings of the 2002 ACM symposium on Applied computing*, pages 590–594, 2002.

[120] W. S. McCulloch and W. H. Pitts. A logical calculus of the ideas immanent in nervous activity. *Bulletin of Mathematical Biophysics*, 5:115–133, 1943.

[121] F. Menczer and D. Parisi. Evidence of hyperplanes in the genetic learning of neural networks. *Biological cybernectics*, 66:283–289, 1992.

[122] K. Merrick, M. Maher, and R. Saunders. Archieving adaptable behaviour in intelligent room using curious supervised learning agents. In *Proceedings of CAADRiA 2008 Beyond Computer Aided Design*, pages 185–192, 2008.

[123] K. E. Merrick. *Modelling motivation for experience-based attention focus in reinforcement learning*. PhD thesis, University of Sydney, 2007.

[124] J. W. L. Merrill and R. F. Port. Fractally configured neural networks. *Neural networks*, 4(1):53–60, 1991.

[125] J. S. Mill. *A system of logic*. 1843.

[126] B. L. Miller and D. E. Golderg. Genetic algorithms, tournament selection and the effects of noise. *Complex Systems*, 9:193–212, 1995.

[127] M. Minksy and S. Papert. *Perceprons*. MIT Press, 1969.

[128] Tom M. Mitchell. *Machine learning*. MIT Press, 1997.

[129] E. Mjolsness, D. H. Sharp, and B. K. Alpert. Scaling, maching learning, and genetic neural nets. *Advances in applied mathematics*, 10:137–163, 1989.

[130] A. Mohamed, G.E. Dahl, and G. Hinton. Acoustic modeling using deep belief networks. *IEEE Transactions on audio, speech and language processing*, 20(1):14–22, 2011.

[131] A. Mohamed, T. N. Sainath, G. Dahl, and B. Ramabhadran. Deep belief networks using discriminative features for phone recognition. In *Proceedings of 2011 IEEE International conference on acoustics, speech and signal proceeding (ICASSP)*, pages 5060–5063, 2011.

[132] D. Montana and L. Davis. Training feedforward neural networks using genetic algorithms. In *In Proceedings of the 11th International joint conference artificial intelligence*, pages 762–767, 1989.

[133] A. Moore and W. K. Wong. Optimal reinsertion: A new search operator for accelerated and more accurate bayesian network structure learning. In *Proceedings of the 20th international conference on machine learning*, volume 3, pages 552–559, 2003.

[134] D. E. Moriarty and R. Miikkulainen. Forming neural networks through efficient and adaptive coevolution. *Evolutionary Computation*, 5(4):373–399, 1998.

[135] L. Moss, D. Sleeman, M. Sim, et al. Ontology-driven hypothesis generation to explain anomalous patient responses to treatment. *Research and development in intelligent systems*, pages 63–76, 2010.

[136] A. Mueller. Fast sequential and parallel algorithms for association rule mining: a comparison. Technical report, University of MaryLand, 1995.

[137] J. Neyman. Statistical problems in agriculture experimentation. *Supplement of journal of the royal statistical society*, 2:107–180, 1935.

[138] R. Ng, L. Lakshmanan, and J. Han. Exploratory mining and pruning optimisations of constrained association rules. In *Proceedings of international conference on management of data*, pages 13–24, 1998.

[139] S. Nolfi, J. L. Elman, and D. Parisi. Learning and evolution in neural network. Technical report, University of California, 1990.

[140] S. Oliker, M. Furst, and O. Maimon. A distributed genetic algorithm for neural netowrk design and training. *Complex systems*, 6(5):459–477, 1992.

[141] M. A. Oquendo, E. Baca-Garcia, A. Artes-Rodriguez, F. Perez-Cruz, H.C. Galfalvy, H. Blasco-Fontecilla, D. Madigan, and N. Duan. Machine learning and data mining: strategies for hypothesis generation. *Molecular Psychiatry*, 17:956–959, 2012.

[142] S. A. Ozel and H. A. Guveir. An algorithm for mining association rules using perfect hashing and database pruning. In *Proceedings of the 10th Turkish symposium on artificial intelligence and neural networks*, pages 257–264, 2001.

[143] D. Parasi, F. Cecconi, and S. Nolfi. Econets: neural networks that learn in an environment. *Network*, 1(2):149–168, 1990.

[144] J. S. Park, M. S. Chen, and P. S. Yu. Using a hash-based method with transaction trimming and database scan reduction for mining association rules. *IEEE Transactions on Knowledge and Data Engineering*, 9(5):813–825, 1997.

[145] N. Pasquier, Y. Bastide, R. Taouil, and L. Lakhal. Discovering frequent closed itemsets for association rules. In *Proceedings of the 7th international conference on database theory*, pages 398–416, 1999.

[146] J. Pearl and T. Verma. A theory of inferred causation. In *Proceedings of the second international conference on principles of knowledge representation and reasoning*, pages 441–452, 1991.

[147] Judea Pearl. *Causality: Models, reasoning, and inference*. Cambridge University Press, 2000.

[148] J. Pei, G. Dong, W. Zhou, and J. Han. On computing condensed frequent pattern bases. In *Proceedings of the 2002 international conference on data mining*, pages 378–385, 2002.

[149] J. Pei, J. Han, and L. Lakshmannan. Mining frequent itemsets with convertible constraints. In *Proceedings of International conference on data engineering*, pages 433–442, 2001.

[150] G. Piatetsky-Shapiro. Discovery, analysis, and presentation of strong rules. In *Notes AAAI'91 Workshop Knowledge Discovery in Databases*, pages 229–248, 1991.

[151] M. A. Potter and K. A. De Jong. A cooperative coevolutionary approach for function optimization. In *Proceedings of Parallel Problem Solving from nature*, pages 249–257, 1994.

[152] M. A. Potter and K. A. De Jong. Cooperative coevolution: An architecture for evolving coadapted subcomponents. *Evolutionary Computation*, 8(1):1–29, 2000.

[153] J. W. Pratt and R. Schlaifer. On the interpretation and observation of laws. *Journal of Econometrics*, pages 23–52, 1988.

[154] L. Prechelt. Early stopping–but when? *Lecture notes in computer science*, 1524:55–69, 1998.

[155] K. V. Price, R. Storn, and J. A. Lampinen. *Differential Evolution: a practical approach to global optimization*. Springer, 2005.

[156] P. W. Price. *Biological Evolution.* Saunders College Publishing, 1996.

[157] H. Y. Quek, K. C. Tan, and H. A. Abbass. Evolutionary game theoretic approach for modeling civil violence. *IEEE Transactions on Evolutionary Computation*, 13(4):780–800, 2009.

[158] H. Reichenbach. *The direction of time*. Berkeley: University of California Press, 1956.

[159] J. Rissanen. Modelling by shortest data description. *Automatica*, 14:465–471, 1978.

[160] R. W. Robinson. Counting unlabelled acyclic digraphs. *Springer lecture notes in mathematics: Combinatorial Mathematics V*, pages 28–43, 1977.

[161] F. Rosenblatt. The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological review*, 65(6):386–408, 1958.

[162] C. D. Rosin and R. K. Belew. New methods for competitive coevolution. *Evolutionary Computation*, 5(1):1–29, 1997.

[163] N. Le Roux, N. Heess, J. Shotton, and J. M. Shotton. Learning a generative model of images by factoring appearance and shape. *Neural Computation*, 23(3):593–650, 2011.

[164] D. B. Rubin. Teaching statistical inference for causal effects in experiments and observational studies. *Journal of education and behavioural statistics*, 3:343–367, 2004.

[165] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Learning representations by back-propagating errors. *Nature*, pages 533–536, 1986.

[166] D. E. Rumelhart and J. L. McClelland. *Parallel distributed processing: explorations in the microstructure of cognition.* MIT Press, 1986.

[167] D. A. Ruths, L. Nakhleh, et al. Hypothesis generation in signaling network. *Journal of computational biology*, pages 1546–1557, 2006.

[168] R. Salakutdinov and G. E. Hinton. Deep boltzmann machines. In *Proceedings of international conference on artificial intelligence and statistics*, pages 448–455, 2009.

[169] A. Savasere, E. Omiecinski, and S. Navathe. Mining for strong negative associations in a large database of customer transactions. In *Proceedings of the 1998 international conference on data engineering*, pages 494–502, 1998.

[170] A. Savasere, E.R. Omiecinski, and S.B. Navathe. An efficient algorithm for mining association rules in large databases. In *Proceedings of 21th International Conference on Very Large Data Bases*, pages 432–444, 1995.

[171] J. D. Schaffer, R. A. Caruana, and L. J. Eshelman. Using genetic search to exploit the emergent behavior of neural network. *Physica D: Nonlinear Phenomena*, 42:244–248, 1990.

[172] G. E. Schwartz. Estimating the dimension of a model. *Annals of Statistics*, 6(2):461–464, 1978.

[173] H. P. Schwefel. *Evolutionary strategy and numerical optimisation.* PhD thesis, Technical university of Berlin, 1975.

[174] K. Shafi. *An online and adaptive signature-based approach for intrusion detection using learning classifier system*. PhD thesis, University of New South Wales Canberra Campus, 2008.

[175] K. Shafi and K. Merrick. A curious agent for network anomaly detection. In *Proceedings of the 10th International conference on autonomous agents and multi-agent systems*, pages 1075–1076, 2011.

[176] A. Siebes, J. Vreeken, and M. Leeuwen. Item sets that compress. In *Proceedings of the 2006 SIAM international conference on data mining*, pages 393–404, 2006.

[177] M. Smith. *Neural networks for statistical modelling*. International thomson computer press, 1996.

[178] S. F. Smith. Flexible learning of problem solving heuristics through adaptive search. In *Proceedings of 8th International Joint Conference Artificial Intelligence*, pages 593–623, 1983.

[179] P. Smolensky. Infomration processing in dynamical systems.: Foundations of harmony theory. In *Parallel distributed processing: Explorations in the microstructure of cognition.*, pages 194–281. MIT Press, 1986.

[180] D. Sofge, K. De Jong, and A. Schultz. A blended population approach to cooperative coevolution for decomposition of complex problems. In *Proceedings of the 2002 Congress on Evolutionary Computation*, volume 1, pages 413–418, 2002.

[181] P. Spirtes. Introduction to causal inference. *Journal of Machine Learning Research*, 11:1643–1662, 2012.

[182] P. Spirtes and C. Glymour. An algorithm for fast recovery of sparse causal graphs. Technical report, Carnegie Mellon University, 1990.

[183] P. Spirtes, C. Glymour, and R. Scheines. *Causation, prediction, and search*. The MIT Press, 2000.

[184] Peter Spirtes. Introduction to causal inference. *Journal of machine learning research*, pages 1643–1662, 2010.

[185] R. Srikant and R. Agrawal. Mining generalised association rules. In *Proceedings of International conference on very large data bases*, pages 55–86, 1995.

[186] Ramakrishnan Srikant and Rakesh Agrawal. Mining quantitative association rules in large relational tables. In *Proceedings of the ACM SIGMOD conference on management of data*, pages 1–12, 1996.

[187] M. Srinivas and L. M. Patnaik. Learning neural network weights using genetic algorithms-improving performance by search-space reduction. In *Proceedings of 1991 IEEE International joint conference on neural networks*, pages 2331–2336, 1991.

[188] K. Stanley and M. Risto. Evolving neural networks through augmenting topologies. *Evolutinoary computation*, pages 99–127, 2002.

[189] W. Stolzman. Anticipatory classifier systems. In *Proceedings of the 3rd annual genetic programming conference*, pages 658–664, 1998.

[190] R. Storn and K. Price. Differential evolution a simple and efficient adaptive scheme for global optimization over continuous spaces. Technical report, International Computer Science institute, 1995.

[191] K. Sun and F. Bai. Mining weighted association rules without preassigned weights. *IEEE Transactions on Knowledge and Data Engineering*, 20(4):489–496, 2008.

[192] A. H. Sung. Ranking importance of input parameters of neural networks. *Expert Systems with Applications*, 15:405–411, 1998.

[193] K.C. Tan, Y.H. Chew, T.H. Lee, and Y. J. Yang. A cooperative coevolutionary algorithm for multiobjective optimization. In *Proceedings of IEEE international conference on systems, man and cybernetics*, volume 1, pages 390–395, 2004.

[194] H. Toivonen. Sampling large databases for association rules. In *Proceedings of International Conference on Very Large Data Bases*, pages 134–145, 1994.

[195] T. Verma and J. Pearl. Equivalence and synthesis of causal models. In *Proceedings of the 6th conference on uncertainty in artificial intelligence*, pages 220–227, 1990.

[196] H. Wang and Y. He. Mining frequent itemsets using support constraints. In *Proceedings of International conference very large data bases*, pages 494–502, 2000.

[197] K. Wang, L. Tang, J. Han, and J. Liu. Top down fp-growth for association rule mining. *Advances in Knowledge discovery and data mining*, 2336:334–340, 2002.

[198] P. Werbos. *Beyond regression: new tools for prediction and analysis in the behavioural sciences*. PhD thesis, Harvard University, 1974.

[199] M. D. Wheeler and K. Ikeuchi. Sensor modeling, probabilistic hypothesis generation, and robust localisation for object recognition. *IEEE Transaction on Pattern Analysis and Machine Learning*, 17(3):252–265, 1995.

[200] D. Whitley, T. Starkweather, and C. Bogart. Genetic algorithms and neural networks: optimizing connections and connectivity. *Parallel computation*, 14:347–361, 1990.

[201] R. P. Wiegand. *An analysis of cooperative coevolutionary algorithm*. PhD thesis, Winthrop University, 1999.

[202] L. Wilkinson. Statistical methods in psychology journals: guidelines and expectations. *American Psychologist*, 54:594–604, 1999.

[203] S. W. Wilson. Mining oblique data with xcs. *Advances in learning classifier systems*, pages 158–174, 2001.

[204] M. Wooldridge and N. Jennings. Intelligent agents: theory and practice. *The knowledge engineering*, 10(2):115–152, 1995.

[205] S. Wright. Correlation and causation. *Journal of agricultural research*, pages 557–585, 1921.

[206] T. Wu, Y. Chen, and J. Han. Re-examination of interestingness measures in pattern mining: a unified framework. *Data Mining and Knowledge Discovery*, 21:371–397, 2010.

[207] Y. Wu, C. Chiang, and A. Chen. Hiding sensitive association rules with limited side effects. *IEEE Transactions on Knowledge and Data Engineering*, 19(1):29–43, 2007.

[208] X. Yao. Evolving artificial neural networks. *Proceedings of the IEEE*, 87(9):1423–1447, 1999.

[209] X. Yao and Y. Liu. A new evolutionary system for evolving artificial neural networks. *IEEE Transactions on Neural Networks*, 8:694–713, 1997.

[210] S. J. Yen and A. L.P. Chen. An efficient approach to discovering knowledge from large databases. In *Proceedings of 4th International conference on parallel and distributed information system*, pages 8–18, 1996.

[211] K. Yoda, T. Fukuda, and Y. Morimoto. Computing optimised rectilinear regions for association rules. In *Proceedings of the 1997 international conference knowledge discovery and data mining*, pages 721–724, 1997.

[212] M. J. Zaki. Scalable algorithm for association mining. *IEEE Transactions on Knowledge and Data Engineering*, 12(3):372–390, 2000.

[213] H. Zhang, B. Padmanabhan, and A. Tuzhilin. Fast mining of spatial collocatio. In *Proceedings of International conference on knowledge discovery in databases*, pages 384–393, 2004.

[214] F. Zhu, X. Yan, J. Han, P. S. Yu, and H. Cheng. Mining colossal frequent patterns by core pattern fusion. In *Proceedings of the 2007 international conference on data engineering*, pages 706–715, 2007.