Doctoral Dissertation

Techniques for Estimating Variable Relations from Small Samples

Yasuhiro Sogawa

January, 2013

Graduate School of Engineering,
Osaka University

Doctoral Dissertation

Techniques for Estimating Variable Relations from Small Samples

Yasuhiro Sogawa

January, 2013

Graduate School of Engineering,
Osaka University

Preface

This dissertation presents techniques for estimating variable relations from small samples, which was achieved by the author during his Ph.D. course at the Division of Electrical, Electronic, and Information Engineering, Graduate School of Engineering, Osaka University. The dissertation is organized as follows.

Chapter 1 describes recent background in a field of machine learning. Moreover, past frameworks for estimating variable relations are reviewed and their difficulties are discussed. The frameworks can fall roughly into two categories. One is a technique to obtain some knowledge on the directed network representing the ordering of effects among all observed variables. This technique aims to find important variables in the directed network or to identify an entire structure of the directed network. We discuss this technique in Chapter 2 and Chapter 3. The other is to estimate undirected relations between a particular variable (a label variable) and the other explanatory variables, which we discuss in Chapter 4. Both techniques are important and utilized in bioinformatics, economics, marketing and so on. In this chapter, the outline of these techniques is described. Then, we clarify a position of our works.

In Chapter 2, we first present a linear non-Gaussian acyclic model (LiNGAM model), which is one of the models to represent variable relations including the orderings of the effects. Conventional methods based on this LiNGAM model enable a robust estimation of the network of all variables including their orderings. However, the accuracy of the estimation becomes worse for data containing the huge number of variables and small samples (e.g. gene datasets). In our work, instead of estimating the entire structure of the directed network, we focus on exogenous variables that work as origins activating a state change of other variables in the network. We propose a method for estimating them from small samples. In this chapter, we investigate performance of the proposed method by numerical experiments with artificial datasets. Moreover, we apply the method to gene datasets and confirm its practicality by comparing the results from domain knowledge.

Chapter 3 proposes another LiNGAM-model-based method for a more accurate estimation of the directed network under a situation with noisy and small sample data. Our proposed method is achieved by an improvement of a statistical independence measure and introducing a more sophisticated solution search algorithm into the conventional method. In numerical experiments, we compare the proposed method with the conventional method and show an advantage of our method under the situation with noisy and smaller sample data.

In Chapter 4, we further review techniques for estimating the relations between a label variable and explanatory variables. This technique is known as regression. In contrast to Chapter 2 and 3 which focus on the situation that the number of samples of all the observed variables are small, we focus on the situation that all samples have values of their explanatory variables but only small number of the samples have their label variable. For example, in a car insurance company, an insurance fee (a label variable) is determined by its company's employees based on car information, driver's driving records and so on (explanatory variables). Such determination by hand needs enormous cost and time. As a result, the number of labeled samples becomes small and unlabeled ones are large. In recent years, active learning that utilizes both labeled and unlabeled samples in such data has been proposed. In contrast to conventional passive machine learning algorithms, active learning selects some unlabeled samples expected to be informative for learning, asks an user to label them and enables more accurate estimation from small labeled samples. However, conventional active learning methods have an impractical assumption that an user always gives correct labels on selected samples. In this chapter, we propose a new active learning algorithm for estimating variable relations which works accurately even under the situation with noisy labels. We extend a querying measure and incorporate robust divergences into the extended measure. The proposed method is compared with conventional methods and its practicality is evaluated by experiments with artificial and real-world datasets.

Chapter 5 concludes this dissertation.

Acknowledgement

This dissertation presents techniques for estimating variable relations from small samples. These techniques have been carried out during my Ph.D. course at the Division of Electrical, Electronic and Information Engineering, Graduate School of Engineering, Osaka University.

I would like to express my deepest gratitude to my supervisor, Prof. Takashi Washio of the Division of Information and Quantum Sciences, the Institute of Scientific and Industrial Research, Osaka University, for his patient instruction, encouragement, valuable comments and various supports throughout this research.

I am deeply grateful to Prof. Noboru Babaguchi of the Division of Electrical, Electronic and Information Engineering, Graduate School of Engineering, Osaka University and Associate Prof. Yoshinobu Kitamura of the Division of Information and Quantum Sciences, the Institute of Scientific and Industrial Research, Osaka University, who provided insightful suggestions, careful reviews and valuable criticism on the whole content of this dissertation.

I would like to express my deep sense of appreciation to Assistant Prof. Akihiro Inokuchi, Assistant Prof. Shohei Shimizu and Assistant Prof. Yoshinobu Kawahara of the Division of Information and Quantum Sciences, the Institute of Scientific and Industrial Research, Osaka University, for their patient instructions, encouragements and valuable discussions.

I am indebted to Prof. Tetsuya Takine, Prof. Kenichi Kitayama, Prof. Seiichi Sampei, Prof. Kyo Inoue, Prof. Zenichiro Kawasaki of the Division of Electrical, Electronic and Information Engineering, Graduate School of Engineering, Osaka University, for their thoughtful comments.

I appreciate Prof. Aapo Hyvärinen of University of Helsinki, Associate Prof. Seiya Imoto and Assistant Prof. Teppei Shimamura of Institute of Medical Science, University of Tokyo, and Tsuyoshi Ueno of Japan Science and Technology Agency with whom I have cooperatively completed papers involving this dissertation.

I thank all the past and present members of the Department of Reasoning for Intelligence, the Division of Information and Quantum Sciences, the Institute of Scientific and Industrial Research, Osaka University (Washio Laboratory), who offered warm encouragement and friendship that gave me strength through difficult times.

Without the financial support from the Research Fellowships of Japan Society for the Promotion of Science (JSPS) for Young Scientists, this research could not carried out. I would like to give my appreciation to JSPS for its special helps as well as financial supports.

Last, but by no means least, I am heartily thankful to my family, including grandfather, grandmother, father, mother and sister, for their selfless support in all aspect of my life, their kindness, and their encouragement during my entire education.

Contents

Chapter 1 Introduction **1**

Chapter 2 Identification of Exogenous Variables from Small Samples **5**

2.1 Introduction 5

2.2 Background Principles 7

2.2.1 Independent Component Analysis 7

2.2.2 A Linear Non-Gaussian Acyclic Model (LiNGAM Model) . . . 8

2.3 A New Method to Identify Exogenous Variables 9

2.3.1 A Variant of Linear Non-Gaussian Acyclic Structural Equation Model 9

2.3.2 Central Limit Theorem for Independent and Non-Identically Distributed Random Variables 10

2.3.3 Identification of exogenous variables based on non-Gaussianity and uncorrelatedness 11

2.3.4 ExoGenous Generating Variable Finder: EggFinder 13

2.4 Experiments 14

2.4.1 Experiments on Artificial Data 14

2.4.2 Application to Microarray Gene Expression Data 16

2.5 Conclusion 18

Chapter 3 An Improvement of Methods for Learning a LiNGAM model **20**

3.1 Introduction 20

3.2 Related Works 22

3.2.1 An ICA-based Method for Learning a LiNGAM Model 22

3.2.2 A Direct Method for Learning a LiNGAM Model 24

3.3 Approaches for Improving the Conventional Methods 28

3.3.1 Extending the Independence Measure 28

3.3.2 Extending the Search Algorithm 31

3.4 Experiments on Artificial Data 36

3.4.1 Experimental setup 36

3.4.2 Kernel-based variants 37

3.4.3 Variants employing Beam search 39

3.5 Conclusion 40

Chapter 4 Robust Active Learning for Linear Regression via Density Power Divergence 47

4.1 Introduction 47

4.2 Background 49

4.2.1 Linear Regression Model 49

4.2.2 Pool-based Active Learning 50

4.2.3 A Conventional Method using KL-divergence 52

4.3 Extending a Querying Measure by Asymptotic Analysis 53

4.3.1 Asymptotic Analysis on M-estimator 54

4.3.2 Density Power Divergence 55

4.4 Empirical Measures for Querying 59

4.4.1 Approximation of Querying Measure 59

4.4.2 Optimization of Querying Measure 60

4.5 Experiments 61

4.5.1 Evaluation of Robustness 62

4.5.2 Evaluation with Real-world data 64

4.6 Conclusion 65

Chapter 5 Conclusion 68

List of Figures

2.1 An illustration of the linear acyclic model 9

2.2 Percentages of datasets where all the top m estimated variables were actually exogenous under (a) n=30; (b) n=60; (c) n=100; (d) n=200. . 16

2.3 Temporal gene expression levels of (a) HBEGF (EGF stimulation); (b) HBEGF (HRG stimulation); (c) JUN (EGF stimulation); (d) JUN (HRG stimulation); (e) NAB2 (EGF stimulation); (f) NAB2 (HRG stimulation). 19

3.1 The LiNGAM model constructed by the residuals $r_i^{(1)}$ 27

3.2 The procedure to select the $\kappa = 2$ candidate pairs of the ordering and the variable by the beam search algorithm. 35

3.3 The procedure to select the candidate of the exogenous variable by DirectLiNGAM (Beam-DirectLiNGAM with $\kappa = 1$). 35

3.4 Median numbers of errors with increasing the number of outliers. 39

4.1 An illustration of the contaminated distribution and the weighted likelihood estimator 53

4.2 Difference among the five methods under various amounts of noisy labels. 63

4.3 Comparisons of the means-squared error among six methods at each learning step. The left figures are for the MSE without noisy labels and the right are for the MSE with 5% noisy labels. 67

List of Tables

2.1 Candidates for exogenous genes found by EggFinder from the dataset. . 17

3.1 Median errors of the conventional methods based on the LiNGAM model and their variants under (A) 8 variables; (B) 16 variables; (C) 32 variables. 42

3.2 Median computational time (sec) to estimate the ordering by the conventional methods and their variants under (A) 8 variables; (B) 16 variables; (C) 32 variables. 43

3.3 Median errors with the different scale variables. 44

3.4 Median errors of the variants using the beam search with the width of the beam κ=2, 4 and 8 under (A) 8 variables; (B) 16 variables. 45

3.5 Median computational time (sec) of the variants using the beam search with the width of the beam κ=2, 4 and 8 under (A) 8 variables; (B) 16 variables. 46

4.1 Specifications of Datasets 63

Chapter 1

Introduction

In recent years, along the development of computers, their network and their data storage, massive data are stored and utilized to obtain useful knowledge in various fields such as medical service, economics, marketing and so on. Techniques for obtaining useful knowledge from such data are known as data mining or machine learning. In studies of data mining/machine learning, one of the latest topics is to find relations between events or objects from their associated observed data. For instance, in the field of marketing, a relation between a price of a product and the number of its purchasers is informative to determine a price for another product. A further example is the relation between a distance of a house from a city center and its house rent. Such events or objects are taken as random variables in the studies and many statistical techniques for estimating the variable relations have been proposed in a last decade.

Techniques for estimating the variable relations can be roughly categorized into two types. One is a technique to obtain some knowledge on the directed network, where the vertices and the directed edges respectively represent the variables and effects propagating among them. The purpose of this technique is to find important variables in the network or to identify an entire structure of the directed network. As we mentioned before, many empirical sciences and applications aim to estimate relations underlying their objective systems such as natural phenomena, human social behavior and so on. Thus, this technique is employed to know how each variable affects the others and how observed data are generated. A representative model used in this technique is a non-Gaussianity-based model called LiNGAM model. By utilizing non-Gaussianity of variables which is frequently observed in many real-world data, methods based on this model achieve strong identifiability of the directed network [45, 46].

However, the conventional methods cannot estimate the entire structure of the directed network accurately under a situation providing small samples only which is found in many real-world problems. For instance, in bioinformatics, the number of samples in a gene dataset is quite small because of ethical concerns. In such a situation, an estimation of relations between genes could fail and we cannot obtain any knowledge on the network of the genes. Therefore, in Chapter 2, we will first propose a variant of LiNGAM model based on some realistic assumptions, and present a new method based on the model to obtain useful knowledge on the directed network from small sample data. The key idea of this method is to find important variables which work as triggers that activate the chain of the effect in the network. These variables are origins in the directed network, and are called exogenous variables. Their identification is important for various applications.

In Chapter 3, to accurately and robustly estimate the entire structure of the directed network from noisy and small sample data, we will present two principles to modify the past LiNGAM-model-based methods. One is to incorporate kernel based independence measure for enhancing the robustness and the accuracy of the network estimation. The other is to employ the beam search algorithm to avoid the local optima. Then, we will propose variants of the LiNGAM-model-based methods to estimate the directed network accurately and robustly. In these manners, we will discuss our study based on the LiNGAM model and propose methods to obtain useful knowledge on the directed network from small samples in Chapter 2 and 3. The study in Chapter 2 is related to the work published in [51, 48, 50], and that in Chapter 3 is related to [49].

The other technique is to estimate undirected relations between a particular variable and the others. The particular variable is called a label variable and otherwise are called explanatory variables in the domain of machine learning. A representative model used in this technique is widely known as a regression model. While the previous technique aims to obtain some knowledge on the directed network of all observed variables in the data, techniques based on the regression model aim to find the undirected relations between a label variable and explanatory variables only. However, these methods based on the model can be applied to data containing the larger number of the explanatory variables and has been widely employed in various areas because of its applicability.

Nevertheless, the number of the samples in the data to be analyzed by the methods are small in many applications, and the naive methods could fail to estimate the relations. This is because cost to obtain the values of the label variable is more expensive than one for the explanatory variables. As a result, in various applications, many samples lack the values of their label variable while the values of the other variables are known in all samples. For example, in medical service, a degree of severity of a patient is evaluated by a doctor based on the patient's blood pressure, body fat percentage and so on which data are obtained semi-automatically by an examination. The evaluation is time-consuming for the doctor and therefore the samples having the evaluated values are small. Here, we note that the evaluation of the values of the label variable is called labeling. Moreover, the samples having evaluated values are called labeled samples and otherwise are called unlabeled samples.

Recently, a new framework called active learning has been proposed to utilize both a set of the small labeled samples and the unlabeled ones. In contrast to the naive technique for estimating the variable relations, active learning selects some unlabeled samples expected to be informative for the estimation, asks a user to label their label variable and enables more accurate estimation from small labeled samples. However, conventional active learning methods have an impractical assumption that a user always gives correct labels on selected samples while a real-world user is likely to be noisy and thus makes a mistake. Therefore, the methods should be extended to be robust against such noise of the labeling for application to the real-world datasets. Chapter 4 will address the problem of the conventional active learning methods and propose method to robustly estimate the relations under the noisy real-world situation. The study in Chapter 4 is related to the work in [52, 53].

As described above, the techniques for estimating the variable relations are very important and useful in various areas. However, the state of the art has a gap to analyze real-world datasets which usually have small samples and are frequently noisy. Thus, in this dissertation, we close this gap between the conventional methods and real-world problems by addressing the problems of the small samples and the noise. A summary of our contribution in this dissertation is as follows.

- The first contribution is to propose a non-Gaussianity-model based method to estimate exogenous variables in a directed variable network from the data having small samples such as gene datasets.

- The second is to propose another non-Gaussianity-model based method, which enables more accurate and robust estimation of the directed variable network from noisy and small sample data.

- The third is to propose a novel active learning method to robustly estimate the relations between a label variable and explanatory variables from small labeled samples.

In Chapter 2, we first describe backgrounds of the non-Gaussianity-based model and its associated methods. Then, we present the first contribution for obtaining useful knowledge on the variable network from small samples. In Chapter 3, we propose another method along the second contribution to estimate the entire network more accurately under the situation with the data having small samples. Chapter 4 reviews technical backgrounds of active learning and the method for estimating relations between a label variable and explanatory variables. Subsequently, we propose a new method for the third contribution. In Chapter 5, we conclude our work.

Chapter 2

Identification of Exogenous Variables from Small Samples

2.1 Introduction

Many methods have been proposed to obtain some knowledge on the directed network of all observed variables in classical situations where much more samples than observed variables are given ($p \ll n$, p: the number of variables and n: the number of samples). Especially, most of them aim to identify an entire structure of the directed network and use a linear acyclic model to analyze and represent effects between continuous random variables [40, 54]. Estimation of the model commonly uses covariance structure of data only and in most cases cannot estimate the complete structure of the entire directed network (orderings of the variables and connection strengths) of the model without using prior knowledge on the network [40, 54]. Recently, the authors of [45] proposed a non-Gaussian linear acyclic model called LiNGAM model. By utilizing the non-Gaussianity which is frequently observed in the real-world data, they showed that the full structure of a linear acyclic model is identifiable based on non-Gaussianity without any prior knowledge. This is a significant advantage over the conventional methods [40, 54].

However, most statistical works for the identification of the directed variable network including the non-Gaussianity-based methods [45, 46] were established for classical situations having fewer variables than samples ($p<n$), whereas modern statistical analyses using high-dimensional models tackle data containing orders of magnitude more variables than samples ($p \gg n$) [14, 35]. For example, in bioinformatics, the number of

samples in microarray gene expression data are much smaller than the observed genes (variables). This is because experiments with genes are restricted by ethical concerns and cost for the experiments. Thus, we consider situations in which p is in the order of 1,000 or more, while n is around 30 to 200. For such high-dimensional and small sample data, the past methods are often computationally intractable or statistically unreliable.

In this chapter, we propose a method to obtain knowledge on the variable network based on the non-Gaussianity-based model, which requires much smaller sample sizes than conventional methods and works even when $p \gg n$. The key idea is to identify variables which work as triggers that activate a chain of effects in the network instead of estimating the entire structure of the network. These trigger variables are called as exogenous variables, and their identification leads to more efficient experimental designs requiring practical interventions and better understanding of the objective systems. One of promising applications is a detection of drug-target genes [14]. The new method proposed in this chapter can be used to find genes firstly affected by a drug and triggering the gene network. The simpler task of finding exogenous variables than that of the entire model structure would require fewer samples to work reliably. The new method uses a non-Gaussianity measure developed in a fairly recent statistical technique called independent component analysis [25].

This chapter is structured as follows. We first review independent component analysis and linear non-Gaussian acyclic models in Section 2.2. We then define our non-Gaussianity-based model and propose a new algorithm to find exogenous variables in Section 2.3. The performance of the algorithm is evaluated by using artificial data and real-world gene expression data in Section 2.4. Section 2.5 concludes this chapter. This chapter is related to the work published in [51, 48, 50].

2.2 Background Principles

2.2.1 Independent Component Analysis

Independent component analysis (ICA) [25] is a statistical technique originally developed in signal processing. ICA model for a p-dimensional observed continuous random vector x is defined as

$$\mathbf{x} = \mathbf{A}\mathbf{s}, \tag{2.1}$$

where s is a p-dimensional continuous random vector whose components s_i are mutually independent and non-Gaussian, and $\mathbf{A}$ is a constant $p \times p$ invertible matrix. s_i are called independent components. Without loss of generality, we assume each s_i to be of zero mean and unit variance. Let $\widetilde{\mathbf{W}}=\mathbf{A}^{-1}$. Then we have $\mathbf{s}=\widetilde{\mathbf{W}}\mathbf{x}$. It is known that the matrix $\widetilde{\mathbf{W}}$ is identifiable up to permutation of the rows [12].

Let $\widehat{\mathbf{s}}=\mathbf{W}\mathbf{x}$. A major estimation principle for $\widetilde{\mathbf{W}}$ is to find such $\mathbf{W}$ that maximizes the sum of non-Gaussianity of estimated independent components $\widehat{s}_i$, which is known to be equivalent to maximize independence between the estimates when the estimates are constrained to be uncorrelated [25]. In [24], a class of non-Gaussianity measures was proposed:

$$J(\widehat{s}_i) = J_G(\mathbf{w}_i) = \left[\mathbb{E}[G(\mathbf{w}_i^{\mathrm{T}}\mathbf{x})] - \mathbb{E}[G(z)]\right]^2, \tag{2.2}$$

where $\mathbf{w}_i^{\mathrm{T}}$ is the i-th row of the matrix $\mathbf{W}$ and is constrained so that $\mathbb{E}[\widehat{s}_i^2]=\mathbb{E}[(\mathbf{w}_i^{\mathrm{T}}\mathbf{x})^2]=1$ because of the aforementioned assumption on unit variance of s_i. $G(\cdot)$ is a nonlinear and non-quadratic function and z is a Gaussian variable with zero mean and unit variance. In practice, the expectations in Eq. (2.2) are replaced by their sample means. In the rest of this chapter, we say that *a variable u is more non-Gaussian than a variable v if $J(u)>J(v)$*. In the domain of ICA, the following conjecture is widely made [25].

Conjecture 1 *The global maximum of $J_G(\mathbf{w})$ is one of s_i for most reasonable choices of $G(\cdot)$ and the distributions of s_i.*

In particular, if $G(s)=s^4$, Conjecture 1 is true for any continuous random variable whose moments exist and kurtosis is non-zero [24], and it can also be proven that there are no spurious optima [13]. Then the global maximum of the measure in Eq. (2.2) should be one of s_i. However, kurtosis often suffers from sensitivity to outliers. In practice, $G(s)$=exp$(-s^2/2)$ is a suitable candidate for the function $G(\cdot)$ [25].

2.2.2 A Linear Non-Gaussian Acyclic Model (LiNGAM Model)

Relationships between continuous observed variables x_i $(i = 1, \cdots, p)$ are typically assumed to be *linear* and *acyclic* [40, 54]. Each relation can be represented as a linear combination of the variables (Linearity), and each variable never affect itself even through the other variables (Acyclicity). For simplicity, we assume that the variables x_i are of zero mean and unit variance. Let $o(i)$ denote such an ordering of x_i that no later variable affects any earlier variable and b_{ij} denote the connection strength from x_j to x_i. Then the relationship in the linear acyclic model can be expressed as

$$x_i := \sum_{o(j)<o(i)} b_{ij}x_j + e_i, \tag{2.3}$$

where e_i are external influences associated with x_i and are of zero mean and unit variance. Furthermore, '*faithfulness*' [54] is typically assumed. In this context, the faithfulness implies that correlations between variables x_i are entailed by the graph structure only, *i.e.*, the zero/non-zero status of b_{ij}. Finally, the external influences e_i are assumed to be independent, which means there are '*no unobserved confounders*' [54]. Here, unobserved confounders are unobserved variables behind the multiple external influences to statistically change their values. If such unobserved confounders exist, some e_i are not mutually independent.

We emphasize that x_i is equal to e_i if it is not influenced by any other observed variable x_j $(j{\neq}i)$, *i.e.*, all the b_{ij} $(j{\neq}i)$ are zeros. In other words, an external influence e_i is *observed* as x_i. Then, such e_i or x_i are called exogenous variables. Otherwise, e_i is called an *error*. For example, consider the model defined by

$$\begin{aligned} x_1 &= e_1, \\ x_2 &= 1.5x_1 + e_2, \\ x_3 &= 0.8x_1 - 1.3x_2 + e_3. \end{aligned} \tag{2.4}$$

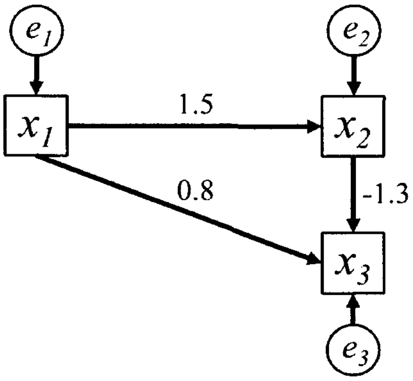


Figure 2.1: An illustration of the linear acyclic model

x_1 is equal to e_1, *i.e.*, it is not influenced by either x_2 or x_3. Moreover, x_2 is influenced by x_1 and x_3 is influenced by both x_1 and x_2. Thus, $x_1(=e_1)$ is an exogenous variable, and e_2 and e_3 are errors. Note that there *exists at least one exogenous variable* $x_i(=e_i)$ because of the model assumption of the acyclicity and no unobserved confounders. Fig. 2.1 shows an illustration of the linear acyclic model of the example Eq. (2.4).

Recently, the authors of [45] proposed a linear non-Gaussian acyclic model called LiNGAM model, where the external influences e_i are assumed to be non-Gaussian while conventional models have assumed that the external influences are Gaussian. Methods based on the LiNGAM model have strong identifiability of the entire variable network under the classical situation with $p \ll n$. In the next section, we will define a variant of the LiNGAM model to identify exogenous variables from small sample data.

2.3 A New Method to Identify Exogenous Variables

2.3.1 A Variant of Linear Non-Gaussian Acyclic Structural Equation Model

We make an additional assumption on the distributions of e_i in the model (2.3) and define our non-Gaussian linear acyclic model, which is a variant of LiNGAM model [45]. Recall that the set of the external influences e_i consists of both exogenous variables and errors. To characterize the difference between exogenous variables and errors, we make the following additional assumption.

Assumption 1 *External influences are non-Gaussian but errors are less non-Gaussian than exogenous variables, i.e.,* $J(e_i) > J(e_j)$ *if* $x_i{=}e_i$ *and* e_j *is an error associated with an endogenous variable* x_j.

The only difference between LiNGAM model and our model is the assumption that errors are less non-Gaussian than exogenous variables. Let a p-dimensional vector $\mathbf{x}$ be a set of the observed variables x_i and a p-dimensional vector $\mathbf{e}$ be a set of external influences e_i. Let a $p{\times}p$ matrix $\mathbf{B}$ consist of the connection strengths b_{ij} where the diagonal elements b_{ii} are all zeros. Then we write our model (the model (2.3) + Assumption 1) in a matrix form as:

$$\mathbf{x} = \mathbf{B}\mathbf{x} + \mathbf{e}. \tag{2.5}$$

Assumption 1 reflects three facts: i) observed data are often considerably non-Gaussian in many fields [25]; ii) exogenous variables are directly affected by an external factor, which usually has non-Gaussianity; iii) in statistics, errors have been typically considered to arise as sums of a number of unobserved non-Gaussian independent variables, which is why classical methods assume that errors are Gaussian resorting to Theorem 1 [9] in the next subsection, though in reality, many variables are not exactly Gaussian. Therefore, we assume the errors to be non-Gaussian as long as they are less non-Gaussian than exogenous variables each of which is directly affected by the non-Gaussian external factor only. This distinction between exogenous variables and errors leads to a simple estimation of exogenous variables proposed in Subsections 2.3.3 and 2.3.4.

2.3.2 Central Limit Theorem for Independent and Non-Identically Distributed Random Variables

Assumption 1 which states that external influences are non-Gaussian but errors are less non-Gaussian than exogenous variables is motivated by Theorem 1 below. The classical central limit theorem states that the probability distribution of the sum of a large number of independent and *identically* distributed random variables will be less non-Gaussian than the original variables. However, the identicality among the distributions does not always hold in many practical cases, and thus less non-Gaussianity of the summed variables is not obviously ensured by the central limit theorem. A past study assessed a wider condition called Lindeberg's condition where the sum of such random variables will be less non-Gaussian [9]. Let us assume that x_ℓ $(\ell = 1, \cdots, L)$ are independent random variables following their own probability density functions $f_\ell(\cdot)$ each of which has a finite mean $\mu_\ell = E[x_\ell]$ and a finite variance $\sigma_\ell^2 = \mathrm{Var}[x_\ell]$. We denote the sum of the variances by $D_L = \sum_{\ell=1}^{L} \sigma_\ell^2$. Then, the Lindeberg's condition is as follows.

Theorem 1 (Lindeberg's condition) *If random variables satisfy the Lindeberg's condition:*

$$\lim_{L\to\infty} \frac{1}{D_L} \sum_{\ell=1}^{L} \int_{|x_\ell - \mu_\ell| \geq \alpha\sqrt{D_L}} (x_\ell - \mu_\ell)^2 f_\ell(x_\ell) dx_\ell = 0 \;\; for \;\; \forall\alpha > 0,$$

the sum of the independent random variables will converge in distribution to Gaussian as $L \to \infty$. □

It can be expected that random variables hardly have distributions other than ones having the Lindeberg's condition in most cases. Therefore, if errors are sums of many unobserved independent variables that have approximately the same magnitudes of non-Gaussianity as exogenous variables, it can be expected that they are less non-Gaussian than exogenous variables. Because of the limitation of the number of summed variables, errors would not to be exactly Gaussian. These considerations motivate the aforementioned Assumption 1.

2.3.3 Identification of exogenous variables based on non-Gaussianity and uncorrelatedness

We relate the linear non-Gaussian acyclic model (2.5) with ICA similarly to [45]. Let us solve the model (2.5) for x and then we have an ICA model represented by Eq. (2.1) as follows

$$\mathbf{x} \;=\; (\mathbf{I}-\mathbf{B})^{-1}\mathbf{e} = \mathbf{A}'\mathbf{e}. \tag{2.6}$$

Note that $\mathbf{I}-\mathbf{B}$ is invertible since it can be permuted to be lower triangular due to the acyclicity assumption [45] and its diagonal elements are all non-zero (unity). In the next subsection, we propose a new algorithm to find exogenous variables $x_i(=e_i)$ using the relation (2.6). In this subsection, we present two lemmas that ensure the validity of the algorithm.

Lemma 1 *Assume that the input data* $\mathbf{x}$ *follows the model (2.5) and that Conjecture 1 (Section 2.2.1) is true. Let us denote by* V_x *the set of all the observed variables* x_i*. Then, the most non-Gaussian observed variable in* V_x *is exogenous:* $J(x_i)$ *is maximum in* V_x $\Rightarrow x_i{=}e_i$. □

Proof Eq. (2.6) shows that the model (2.5) is an ICA model, where external influences e_i are independent components (ICs). The set of the external influences consists of exogenous variables and errors. Due to the model assumption (Assumption 1 in Subsection 2.3.1), exogenous variables are more non-Gaussian than errors. Therefore, the most non-Gaussian *exogenous* variable is the most non-Gaussian IC. Next, according to Conjecture 1 that is here assumed to be true, the most non-Gaussian IC, *i.e.*, the most non-Gaussian *exogenous* variable, is the global maximum of the non-Gaussianity measure $J(\mathbf{w}^\mathsf{T}\mathbf{x}){=}J_G(\mathbf{w})$ among such linear combinations of observed variables $\mathbf{w}^\mathsf{T}\mathbf{x}$ with the constraint $\mathbb{E}[(\mathbf{w}^\mathsf{T}\mathbf{x})^2]{=}1$, which include all the observed variables x_i in V_x. Therefore, the most non-Gaussian observed variable is the most non-Gaussian *exogenous* variable. ■

Lemma 2 *Assume the assumptions of Lemma 1. Let us denote by* E *a strict subset of exogenous variables. That is,* E *is a subset of exogenous variables but there exists at least one exogenous variable not contained in* E*. Let us denote by* U_E *the set of observed variables uncorrelated with any variable in* E*. Then the most non-Gaussian observed variable in* U_E *is exogenous:* $J(x_i)$ *is maximum in* $U_E \Rightarrow x_i{=}e_i$. □

Proof First, the set V_x is the union of three disjoint sets: E, U_E and C_E, where C_E is the set of observed variables in $V_x \backslash E$ correlated with a variable in E. By definition, any variable in U_E are not correlated with any variable in E. Since the faithfulness is assumed, the zero correlations are only due to the graph structure. Therefore, there is no directed path from any variable in E to any variable in U_E. Similarly, there is a directed path from each (exogenous) variable in E to a variable in C_E. Next, there can be no directed path from any variable in C_E to any variable in U_E. Otherwise, there would be a directed path from such a variable in E, from which there is a directed path to a variable in C_E, to a variable in U_E through the variable in C_E. Then, due to the faithfulness, the variable in E must correlate with the variable in U_E, which contradicts the definition of U_E.

To sum up, there is no directed path from any variable in $E \cup C_E$ to any variable in U_E. Since any directed path from the external influence e_i associated with any variable x_i in V_x must go through x_i, there is no directed path from the external influence associated with any variable in $E \cup C_E$ to any variable in U_E. In other words, there can be directed paths from *only* the external influences associated with any variables in U_E to some variables in U_E. Then, we again have an ICA model: $\tilde{\mathbf{x}}=\tilde{\mathbf{A}}'\tilde{\mathbf{e}}$, where $\tilde{\mathbf{x}}$ and $\tilde{\mathbf{e}}$ are vectors whose elements are the variables in U_E and corresponding external influences in $\mathbf{e}$ in Eq. (2.6), and $\tilde{\mathbf{A}}'$ is the corresponding submatrix of $\mathbf{A}'$ in Eq. (2.6). Recursively applying Lemma 1 shows that the most non-Gaussian variable in U_E is exogenous. ■

To find uncorrelated variables, we simply use the ordinary Gaussianity-based testing method [33] and control the false discovery rate [8] to 3% for multiplicity of tests. Though non-parametric methods [33] are desirable for more rigorous testing in the non-Gaussian setting, we used the Gaussian method that is more computationally efficient and seems to work relatively well in our simulations.

2.3.4 ExoGenous Generating Variable Finder: EggFinder

Based on the discussions in the previous subsection, we propose an algorithm to successively find exogenous variables, which we call EggFinder (ExoGenous Generating variable Finder). Algorithm 1 shows a pseudo code of EggFinder. At Step 2(c) in Algo-

Algorithm 1 ExoGenous Generating Variable Finder: EggFinder

```
1. Given $V_x$, initialize $E=\emptyset$, $U_E^{(1)}=V_x$, and $m:=1$.
2. Repeat until no variables $x_i$ are uncorrelated with exogenous variable candidates, i.e., $U_E^{(m)}=\emptyset$:
    (a) Find the most non-Gaussian variable $x^{(m)}$ in $U_E^{(m)}$:
            $x^{(m)} = \arg \max_{x \in U_E^{(m)}} J(x),$
        where $J$ is the non-Gaussianity measure in Eq. (2.2) with
            $G(x) = \exp(-x^2/2).$
    (b) Add the most non-Gaussian variable $x^{(m)}$ to $E$, that is, $E=E \cup \{x^{(m)}\}$.
    (c) Let $U_E^{(m+1)}$ be the subset of $U_E^{(m)}$ where variables are uncorrelated with $x^{(m)}$, and $m=m+1$.
```

rithm 1, we use the Gaussianity-based correlation testing method and control the false discovery rate to 3% to remove the variables correlated with the selected candidates of the exogenous variables.

2.4 Experiments

2.4.1 Experiments on Artificial Data

We studied the performance of EggFinder when $p \gg n$ under a linear non-Gaussian acyclic model having a sparse graph structure and various non-Gaussianity conditions for errors. Many real-world networks such as gene networks are often considered to have scale-free graph structures. However, as far as we know, there is no standard way to create a *directed* scale-free graph. Therefore, we randomly created a sparse directed acyclic graph with $p=1{,}000$ variables using a software Tetrad [1]. The resulting graph contained 1,000 edges and ℓ=171 exogenous variables. We randomly determined each element of the matrix $\mathbf{B}$ in the model (2.5) to follow this graph structure and make the standard deviations of x_i owing to parent observed variables ranged in the interval $[0.5, 1.5]$.

We generated exogenous variables and errors as follows. We randomly generated a non-Gaussian exogenous variable $x_i(=e_i)$ that was sub- or super-Gaussian with probability 50%. We first generated a Gaussian variable z_i with zero mean and unit variance and subsequently transformed it to a non-Gaussian variable by $e_i = \text{sign}(z_i)|z_i|^{\delta_i}$. The nonlinear exponent δ_i was randomly selected to lie in $[0.5, 0.8]$ or $[1.2, 2.0]$ with probability 50%. The former gave a sub-Gaussian symmetric variable, and the latter a super-Gaussian symmetric variable. Finally, the transformed variable e_i was scaled to the standard deviation randomly selected in the interval $[0.5, 1.5]$ and was taken as an exogenous variable. Next, for each error e_j, we randomly generated h (h=1, 3, 5 and 50) non-Gaussian variables having unit variance in the same manner as for exogenous variables and took the sum of them. We then scaled the sum to the standard deviation selected similarly to the cases of exogenous variables and finally took it as an error e_j. A larger h would generate a less non-Gaussian error due to Theorem 1.

Finally, we randomly generated 500 datasets under each combination of h and n (n=30, 60, 100 and 200) and fed the datasets to EggFinder. For each combination, we computed percentages of datasets where all the top m estimated variables were actually exogenous. In Fig. 2.2, the relations between the percentage and m are plotted. First, the percentages generally increase when the sample size n increases. This is clear since a larger n enables more accurate estimation of non-Gaussianity and correlation. Second, similar changes of the percentages are observed when h is larger. This is reasonable because a larger h generates data more consistent with Assumption 1 of the model (2.5) as we mentioned before. In summary, EggFinder successfully finds a set of exogenous variables up to more than m=10 in many conditions. However, EggFinder may not find all the exogenous variables when $p \gg n$, although it asymptotically finds all the exogenous variables if all the assumptions made in Lemmas 1 and 2 hold.

Interestingly, EggFinder did not fail completely and estimated a couple of exogenous variables even for the h=1 condition where the distributional assumption on errors was

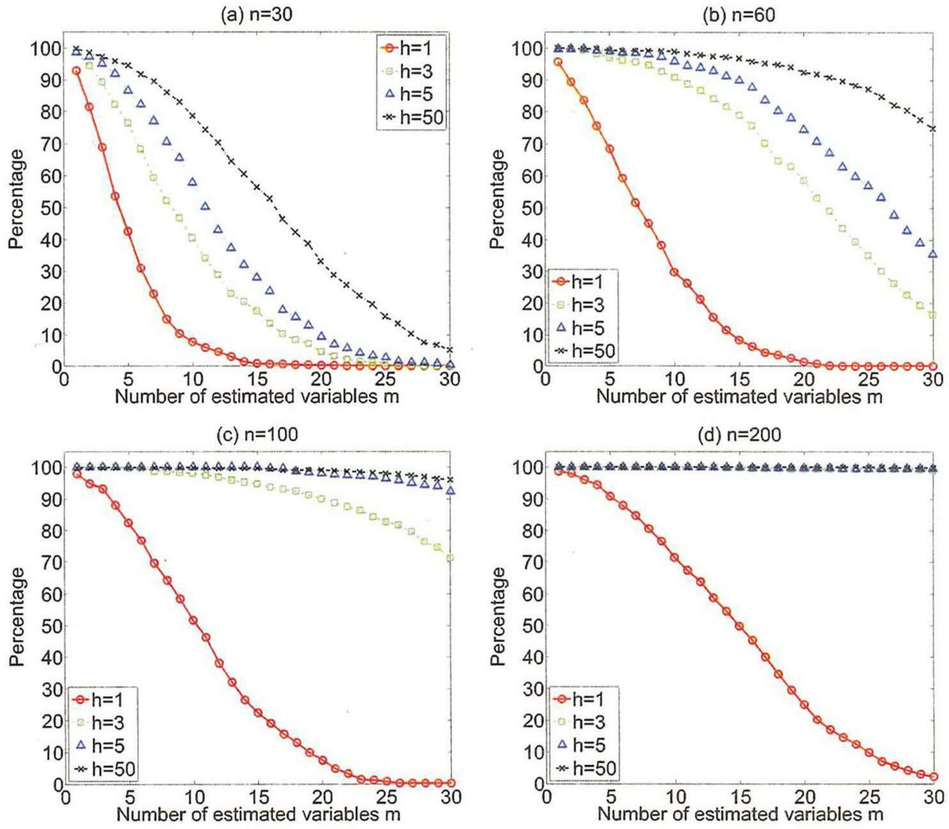


Figure 2.2: Percentages of datasets where all the top m estimated variables were actually exogenous under (a) n=30; (b) n=60; (c) n=100; (d) n=200.

most likely to be violated. This is presumably because all the variables in the network might satisfy the condition mentioned in Theorem 1. Therefore, due to Theorem 1, the endogenous observed variables, which are lower in the network, are more likely to be less non-Gaussian than the exogenous variables, even if the errors and the exogenous variables have the same degree of non-Gaussianity.

2.4.2 Application to Microarray Gene Expression Data

To evaluate the capability of EggFinder in a real situation, we analyzed a real-world dataset of DNA microarray data collected in experiments on a human breast cancer cell line MCF-7 [38], where two ligands of ErbB family receptor, epidermal growth

Table 2.1: Candidates for exogenous genes found by EggFinder from the dataset.

Probe ID	Symbol	Entrez Gene Name
203821_at	HBEGF	heparin-binding EGF-like growth factor
201466_s_at	JUN	jun proto-oncogene
216017_s_at	NAB2	EGR1 binding protein 2

factor (EGF) and heregulin (HRG), were dosed to MCF-7 under four different concentrations, and the gene expression levels were measured. EGF and HRG induce distinct kinase activity patterns and phenotypes of MCF-7 cells. It is known that EGF binds to ErbB1 receptor (EGFR) and induces EGF-stimulated transient activation of extracellular signal-regulated kinase (ERK) induced cell proliferation. While HRG first binds to ErbB3 or ErbB4 receptor and then induces trans-activation of ErbB2 receptor, and HRG-stimulated sustained activation of ERK induces cell differentiation. The number of dose concentrations was eight (0.1, 0.5, 1.0, and 10.0 nmol/ℓ for either EGF or HRG). The gene expression values were measured at seven time points (5, 10, 15, 30, 45, 60 and 90 minutes) after dosing. The total number of experimental conditions was 55 instead of $56{=}8 \times 7$. This is because no experiment under the condition of the concentration of EGF 10.0 nmol/ℓ at 60 minutes elapsed time was conducted. For each condition, the expression levels of 22,277 genes were measured using Affymetrix GeneChip microarrays. As a preprocessing, we focused on 62 genes, which had been selected as genes regulated by both EGF and HRG with multiplicative decomposition model [38]. To estimate exogenous genes under both stimulations from 62 genes, we applied EggFinder to the data matrix of 55 conditions and 62 genes. This is a challenging situation with $p > n$.

EggFinder found three candidates for exogenous genes shown in Table. 2.1. Fig. 2.3 shows temporal gene expression levels of these three candidates. As described in Fig. 2.3, HBEGF and JUN show different expression patterns between EGF and HRG stimulations; under HRG stimulation, the expressions of HBEGF monotonically more increased, and JUN's expressions achieved a higher peak at 45 or 60 min than under EGF stimulation. Biologically, JUN binds to FOS [29], which was identified as a master regulator determining cell fate in [38]. Thus, the analysis of EggFinder suggested that JUN is a candidate master regulator that determines kinase activity patterns and phenotypes caused by EGF and HRG. HBEGF also binds to EGFR [28] and ErbB4 [16] and then induces activation of ERK [47]. Note that HRG-stimulated sustained activation of ERK requires consecutive formation of ErbB1(EGFR)-ErbB3 and ErbB2-ErbB3 heterodimers [19]. Thus, the analysis of EggFinder produced a biological hypothesis that HBEGF plays a crucial role as an accelerator that amplifies expressions of downstream genes of ERK pathway only when stimulated by HRG. Although expression levels of NAB2 do not have clear differences between dose concentrations, we found that the average expression level of NAB2 under HRG stimulation were lower than that under EGF stimulation. NAB2 represses transcriptions induced by EGR family (EGR1 and EGR2) [55] which are regulated by FOS. Since EGR1 increases expression of human EGFR mRNA and protein [39], decreased expression of NAB2 under HRG stimulation also might be related with consecutive formation of ErbB family. In these manners, the genes worth examining are suggested by EggFinder.

2.5 Conclusion

We defined the variant of conventional non-Gaussianity-based model and proposed the method to estimate exogenous variables from data having small samples. The accuracy of our proposed method was evaluated by the experiments with the artificial datasets and the gene expression dataset. Particularly in the experiments on microarray gene expression data, our method suggested the genes worth examining. These results showed the applicability of our non-Gaussianity-based model and our method. We believe this is an important first step for developing advanced network analysis methods which can find exogenous variables in the network even under the challenging situations $p \gg n$.

One of the important issues for our future research is to establish a way of determining the number of valid exogenous variable candidates. Moreover, relaxing our non-Gaussianity-based model to more general nonlinear model is also important. Further, future work would address what is the better correlation testing procedure taking non-Gaussianity into account to remove the correlated variables in our algorithm.

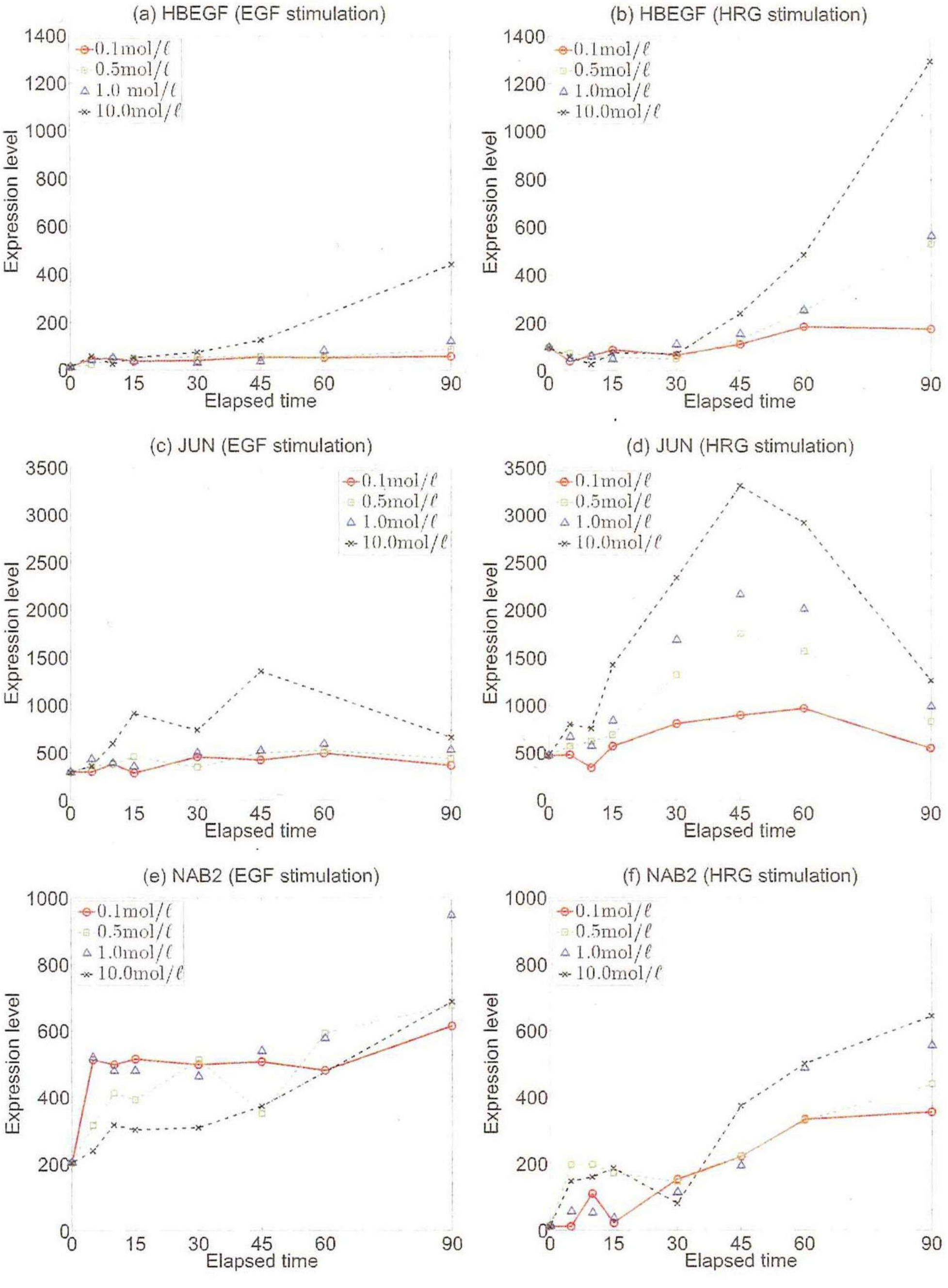


Figure 2.3: Temporal gene expression levels of (a) HBEGF (EGF stimulation); (b) HBEGF (HRG stimulation); (c) JUN (EGF stimulation); (d) JUN (HRG stimulation); (e) NAB2 (EGF stimulation); (f) NAB2 (HRG stimulation).

Chapter 3

An Improvement of Methods for Learning a LiNGAM model

3.1 Introduction

The methods based on the LiNGAM model [45, 46] have strong identifiability of the directed network representing the effect among the observed variables. However, for correct network identification, they practically need to properly examine independence between variables in the network and search a correct network by using finite samples. Nevertheless, the current LiNGAM-model-based methods do not meet with these requirements sufficiently since they employ incomplete measure to evaluate the independence and a simple greedy search algorithm. Particularly in real-world situations having small samples such as gene data analysis, the accuracy of the estimation of the directed network is not acceptable because of the statistical sampling fluctuation. In addition, the robustness to outliers in real-world data is important in estimating the network. Therefore more sophisticated independence measures and more advanced search algorithms should be introduced into the methods to estimate a network of the observed variables more accurately and robustly under small samples.

In this chapter, we propose two approaches to improve the LiNGAM-model-based methods and present our methods by unifying them to enhance their accuracy and robustness under small samples while maintaining their tractable computational time.

The first approach is to modify the independence measures to more sophisticated ones. The methods based on the LiNGAM model need to apply various transformations to the variables and compute their correlations since two variables are independent if and only if their arbitrary bounded transformations of the variables have zero correlation. However, the independence measures used in [45] and [46] include a few types of nonlinear correlations only. In the field of ICA, many independence measures that use wider varieties of the transformations have been proposed [44, 6]. Among such independence measures, a kernel based independence measure studied in [6] supports much varieties of the transformations and examines the independence more strictly than the conventional independence measures. In addition, the kernel based measure has sufficient computational efficiency because of a technique called Kernel Trick, which we will explain in Section 3.3.1. Under these considerations, we propose variants of the methods based on the LiNGAM model which adopt the kernel based independence measure. In the original paper of the kernel based independence measure [6], its robustness to the outliers is well ensured because of the varieties of the transformation. Thus, the measure is expected to provide more accurate and robust estimation of the network to our variants of the LiNGAM-model-based methods.

The second approach is to use beam search [61] instead of a greedy search algorithm used in [46] to more accurately assess the network structure. This beam search algorithm always maintains the constant number of suboptimum solutions at a step in contrast to the greedy search that always selects only one best solution at the step. This search algorithm is expected to provide more accurate network estimation under small samples since it uses more complete search than the greedy search. Here, we note that the beam search algorithm is not expected to enhance the robustness to outliers. This is because outliers statistically affects the independence measure but not to the search process.

We briefly review the conventional methods, ICA-LiNGAM [45] and DirectLiNGAM [46] in the next section. Further in Section 3.3, we propose four variants of the LiNGAM-model-based methods by using kernel based independence measure and/or the beam search. Moreover, we experimentally characterize the conventional methods and our variants in terms of their accuracy, computational cost and robustness to outliers in the section 3.4. Finally, we discuss our results and give a conclusion in Section 3.5. This chapter is related to the work published in [49].

3.2 Related Works

3.2.1 An ICA-based Method for Learning a LiNGAM Model

Firstly, we recall the linear non-Gaussian acyclic model representing the variable network. Let $o(i)$ denote such an ordering of an observed variable x_i that no later variable affects any earlier variable and b_{ij} denote the connection strength from x_j to x_i. Then, the linear non-Gaussian acyclic model, LiNGAM model, is defined as follows:

$$x_i := \sum_{o(j)<o(i)} b_{ij}x_j + e_i, \tag{3.1}$$

where e_i are non-Gaussian external influences associated with x_i. Further, let a p-dimensional vector $\mathbf{x}$ be a set of observed variables x_i and a p-dimensional vector $\mathbf{e}$ be a set of non-Gaussian external influences e_i. Then, the LiNGAM model in matrix form is defined as follows:

$$\mathbf{x} = \mathbf{B}\mathbf{x} + \mathbf{e}, \tag{3.2}$$

where $\mathbf{B}$ is the $p \times p$ strictly lower triangular matrix each element of which is a connection strength b_{ij}. In [45], a method for estimating networks of observed variables in the LiNGAM model by using ICA was proposed which is called ICA-LiNGAM. In this subsection, we explain how the method estimates the connection strength matrix $\mathbf{B}$ and identifies networks.

Let us solve Eq. (3.2) for $\mathbf{x}$. Then we obtain

$$\mathbf{x} = \mathbf{A}\mathbf{e}, \tag{3.3}$$

where $\mathbf{A} = (\mathbf{I} - \mathbf{B})^{-1}$ is a mixing matrix. The mixing matrix $\mathbf{A}$ is identifiable [12] if the observed variables are linear, invertible mixtures of non-Gaussian independent source variables [25] and a sufficient number of samples on the observed variables are given. Since the external influences e_i are independent of each other and non-Gaussian, the LiNGAM model Eq. (3.3) can be defined as the ICA model [25] which is known to be identifiable.

Algorithm 2 ICA-LiNGAM

```
1. Given a p-dimensional variable vector $\mathbf{x}$ and its $p \times n$ data matrix $\mathbf{X}$, apply FastICA [26] to estimate $\mathbf{A}$.
2. Find the only one $\widetilde{\mathbf{W}}$, where $\widetilde{\mathbf{W}}$ denote permuted rows of $\mathbf{W} = \mathbf{A}^{-1}$ to minimize $\sum_i 1/|\widetilde{\mathbf{W}}_{ii}|$ for ensuring the non-zero diagonals.
3. Divide each row of $\widetilde{\mathbf{W}}$ by its corresponding diagonal element, to yield a new matrix $\widetilde{\mathbf{W}}'$ with all ones on the diagonal.
4. Compute an estimate $\hat{\mathbf{B}} = \mathbf{I} - \widetilde{\mathbf{W}}'$.
5. To find an ordering of the observed variables, derive the permutation matrix $\widetilde{\mathbf{P}}$ which yields a matrix $\widetilde{\mathbf{B}} = \widetilde{\mathbf{P}}\widetilde{\mathbf{B}}\widetilde{\mathbf{P}}^T$ which is as close as possible to strictly lower triangular.
```

Though ICA can estimate $\mathbf{A}$ (and $\mathbf{W} = \mathbf{A}^{-1}$), there are still indeterminacies of permutation and scaling. In spite of these indeterminacies, the correct permutation can be found [45] since $\mathbf{B}$ should be a matrix that can be permuted to be strictly lower triangular, in other words, $\mathbf{W} = \mathbf{I} - \mathbf{B}$ is to be lower triangular and have no zeros in the diagonal if $\mathbf{W}$ is correctly permuted. Additionally, the correct scaling of the independent external influences can be found by using the unity on the diagonal of $\mathbf{W} = \mathbf{I} - \mathbf{B}$. Accordingly, ICA-LiNGAM can estimate $\mathbf{B} = \mathbf{I} - \mathbf{W}$ and identify networks without using any prior knowledge. Pseudo code of ICA-LiNGAM is shown in Algorithm 2.

However, there are two potential problems that most ICA algorithms used in the ICA-LiNGAM may not converge to a correct solution in a finite number of steps, and that a permutation algorithm used in ICA-LiNGAM are not scale-invariant. Therefore, they could give a wrong identification of the network. Additionally, ICA-LiNGAM doesn't estimate the networks correctly if it doesn't examine independence between variables in the network properly.

Algorithm 3 DirectLiNGAM

```
1. Given a $p$-dimensional variable vector $\mathbf{x}$, its $p \times n$ data matrix $\mathbf{X}$, and a set $U$ of subscripts of all $x_i \in \mathbf{x}$, initialize an ordering list of variables $K = \emptyset$ and $m := 1$.
2. Repeat until $p - 1$ subscripts are added to $K$.
    (a) Regress $x_i$ on $x_j$ for all $i \in U \backslash K (i \neq j)$ and, derive the residual data matrix $\mathbf{R}^{(j)}$ from the data matrix $\mathbf{X}$ for all $j \in U \backslash K$ by Eq. (3.4). Find a variable $x^{(\lambda^{(m)})}$ which is most independent of its residuals:
            $\lambda^{(m)} = \operatorname*{argmin}_{j \in U \backslash K} T(x_j, U \backslash K),$
        where $T$ is the independence measure shown in Eq. (3.5) and $\lambda^{(m)}$ is the subscript of the selected candidate variable.
    (b) Add the subscript $\lambda^{(m)}$ of the variable that minimize $T$ to the end of $K$.
    (c) Let $\mathbf{X} := \mathbf{R}^{(\lambda^{(m)})}$ and $m := m + 1$.
3. Add the subscript of the remaining variable to the end of $K$.
4. Construct the connection strength matrix $\mathbf{B}$ by ordinary least squares of Eq. (3.6) based on the ordering $K$.
```

3.2.2 A Direct Method for Learning a LiNGAM Model

In [46], another method called DirectLiNGAM for identifying the networks was proposed. In this subsection, we explain how DirectLiNGAM estimates the networks of the observed variables.

Pseudo code of the DirectLiNGAM algorithm is presented in Algorithm 3. At first, it tries to find an exogenous variable as the top variable in an ordering of the network.

Let us denote by $r_i^{(j)}$ the residuals when x_i is regressed on x_j:

$$r_i^{(j)} = x_i - \frac{\mathrm{cov}(x_i, x_j)}{\mathrm{var}(x_j)} x_j, \quad (i \neq j). \tag{3.4}$$

Then, the variable x_j is exogenous if and only if it is independent of its residuals $r_i^{(j)}$ with all $x_i (i \neq j)$ [46]. The independence measure used in DirectLiNGAM [46] is

$$T(x_j, U) = \sum_{i \in U, i \neq j} [|\mathrm{corr}\{Q(r_i^{(j)}), x_j\}| + |\mathrm{corr}\{r_i^{(j)}, Q(x_j)\}|], \tag{3.5}$$

where U is the set of subscripts of all observed variables x_i, and $Q(\cdot)$ is a nonlinear and non-quadratic function $tanh(\cdot)$ which originally used in FastICA [26]. This original paper [26] focuses on the variables following non-Gaussian distributions with high/low kurtosis, and evaluate the independence between the variables. In this condition, the function $Q(\cdot) = tanh(\cdot)$ transforms the non-Gaussian variables to reduce the effect of their kurtosis and make the Gaussianity-based correlation analysis possible to evaluate the independence between the non-Gaussian variables. Further explanation of this function is described in [25]. In many cases, such a nonlinear correlation would evaluate the independence accurately enough as described in the ICA literature [25]. Thus, DirectLiNGAM selects the variable $x_{\lambda^{(m)}}$ that minimize the statistics Eq. (3.5) as the exogenous variable at Step 2(a), and added the subscript $\lambda^{(m)}$ to the end of the ordering list K at Step 2(b). Next, in data X, the component of the exogenous variable to the other latter variables is removed, and we obtain the residual data matrix $\mathbf{R}^{(\lambda^{(m)})}$ by performing the least square regression of Eq. (3.4). The LiNGAM model still holds for the remaining residuals in $U \backslash \{j\}$, and an ordering of the residuals is equivalent to that of the corresponding original observed variables (The proof of the LiNGAM model composed of the residuals is given in [46]). Therefore, DirectLiNGAM can recursively find the second top variable as the *exogenous* variable in the LiNGAM model composed of the residuals. Thus, we set $\mathbf{X} =: \mathbf{R}^{(\lambda^{(m)})}$ at Step 2(c). By repeating these operations 2(a)-(c), the ordering of the observed variables K is obtained. Finally, based on the obtained ordering K, the structure of the connection strength matrix $\mathbf{B}$ and its element

b_{ij} is estimated by ordinary least squares [41] to be:

$$b_{ij} = \underset{b_{ij}}{\text{argmin}} \sum_{u=1}^{n} \left(x_i^u - \sum_{o(j)<o(i)} b_{ij} x_j^u \right)^2, \quad (i \neq j), \tag{3.6}$$

where x_i^u is the u-th sample of the corresponding variable. In addition, we recall that $o(i)$ is the ordering of the observed variable x_i. Since no later variable affects any earlier variable, we set $b_{ij} = 0, (o(i) \geq o(j), i \neq j)$. Further, the diagonal element b_{ii} is zero because each observed variable does not affect itself.

We show an example of the procedure to estimate the network by DirectLiNGAM. Suppose we have a dataset $\mathbf{X}$ containing three observed variables x_1, x_2 and x_3 obtained from the network corresponds to the LiNGAM model shown in Fig. 2.1. Firstly in $m = 1$ iteration, DirectLiNGAM tries to find an exogenous variable by evaluating the independence between x_j and its residuals $r_i^{(j)}$ for all $j \in \{1, 2, 3\}$ and $i \in \{1, 2, 3\} \backslash j$ at Step 2(a) in Algorithm 3. Suppose we obtain the following statistics of the three possible candidates:

$$\begin{aligned} T(x_1, U) &= 0,11 \\ T(x_2, U) &= 5.92, \\ T(x_3, U) &= 0.15. \end{aligned}$$

Here, we recall that if a variable is the most independent of its residuals and is likely to be exogenous, the statistic T would be small. Thus, in this example, x_1 is selected as the exogenous variable in 1st iteration $(m = 1)$. Subsequently, the subscript $\lambda^{(1)} = 1$ is added to the ordering list of variables K, *i.e.* $K = \{1\}$ at Step 2(b). Then, the data matrix $\mathbf{X}$ is updated to the regressed data matrix $\mathbf{R}^{(1)}$ by using the least square regression of Eq. (3.4). As can be shown in Fig. 3.1, the network constructed by the remaining residuals $r_2^{(1)}$ and $r_3^{(1)}$ is also the LiNGAM model and therefore one iteratively tries to find an *exogenous* variable (residual). Secondly in $m = 2$ iteration, if we obtain the following statistics:

$$\begin{aligned} T(x_2, U \backslash K) &= 0.09, \\ T(x_3, U \backslash K) &= 5.65, \end{aligned}$$

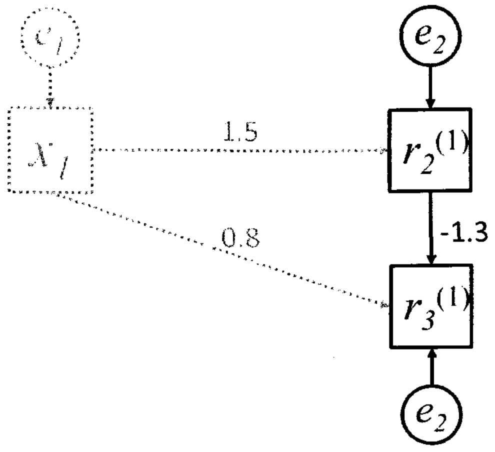


Figure 3.1: The LiNGAM model constructed by the residuals $r_i^{(1)}$

then, x_2 (the residual $r_2^{(1)}$) is selected as the *exogenous* variable at Step 2(a) because of the smallest value of $T(x_2, U\backslash K) = 0, 09$, and the subscript $\lambda^{(2)} = 2$ is appended to the end of K at Step 2(b), *i.e.*, $K = \{1, 2\}$. Finally, the remaining subscript $\{3\}$ is appended to K, and we obtain the ordering list of variables $K = \{1, 2, 3\}$ at Step 3. The connection strength matrix **B** is constructed by ordinary least squares of Eq. (3.6) based on the ordering K at Step 4.

Similarly to ICA-LiNGAM, DirectLiNGAM cannot identify the correct networks if it does not examine the independence between variables in the network properly. Therefore, a choice of the independence measure is important. In addition, once DirectLiNGAM selects a wrong variable as an exogenous variable, one can never find a correct network. Thus, the search algorithm is also important in selecting the candidate exogenous variable.

Nevertheless, DirectLiNGAM has advantages over ICA-LiNGAM. One of the advantages is that DirectLiNGAM always converges to a solution while the convergence of ICA-LiNGAM is not guaranteed because of ICA algorithm. The other is that DirectLiNGAM ensures the scale-invariance while ICA-LiNGAM is strongly influenced by the scale at the permutation procedure. With these advantages, DirectLiNGAM can estimate the network more accurately and has wider applicability than ICA-LiNGAM.

3.3 Approaches for Improving the Conventional Methods

In the previous section, we reviewed the conventional LiNGAM-model-based methods for estimating the variable network. For both ICA-LiNGAM and DirectLiNGAM, an accurate evaluation of the independence between the variables is required to obtain the correct network. In addition, the search algorithm is important for DirectLiNGAM. In this section, we focus on the independence measure and the search algorithm and propose variants of ICA-LiNGAM and DirectLiNGAM.

3.3.1 Extending the Independence Measure

ICA-LiNGAM [45] and DirectLiNGAM [46] use only one type of nonlinear correlations such as Eq.(3.5). Unfortunately, the independence measure used in ICA-LiNGAM [45] and DirectLiNGAM [46] cannot evaluate the independence between the variables accurately. This is because the measure used in them considers only one nonlinear transformation to evaluate the independence while the variables x and y are independent if and only if they satisfies the following condition:

$$\mathrm{corr}\{f(x), g(y)\} = 0, \qquad \forall f(\cdot), \forall g(\cdot), \tag{3.7}$$

for any nonlinear bounded transformations $f(\cdot)$ and $g(\cdot)$. Therefore, we need to consider the correlation between various nonlinear transformations of the variables in evaluating the independence accurately. As just described in the previous chapter, the accurate evaluation of the independence between the variables is important to identify the network. Thus, in this subsection, we propose a variant of the LiNGAM-model-based methods by extending the independence measures employed in [45] and [46] to cover wider classes of the transformation for enhancing the accuracy and the robustness to outliers.

The extension is made by introducing a kernel based independence measure proposed in [6]. The measure is based on kernel canonical correlation analysis (Kernel CCA). Kernel CCA is a method using kernel functions [6] to look for nonlinear transformations of the variables that maximize correlation in the transformed higher-dimensional space. Here, let us denote n samples of x and y by x^i and y^i $(i = 1, \cdots, n)$, respectively. Then, the kernel functions are defined as inner products of the transformations:

$$k_x(x^i, x^j) = \langle \phi_x(x^i), \phi_x(x^j) \rangle,$$
$$k_y(y^i, y^j) = \langle \phi_y(y^i), \phi_y(y^j) \rangle,$$

where $\phi_x(\cdot)$ and $\phi_y(\cdot)$ are nonlinear transformations which map x and y into higher-dimensional space, and $\langle \cdot, \cdot \rangle$ is an operation to take an inner product. Then, Kernel CCA obtains n-dimensional coefficient vectors $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ that maximize the correlation $\rho_{x,y}$ between the transformations of x and y in higher-dimensional space. In other words, Kernel CCA tries to find the nonlinear correlation which is most sensitive to the independence between the variables. The equation to compute the kernel canonical correlation is given as follows:

$$\rho_{x,y} = \max_{\boldsymbol{\alpha},\boldsymbol{\beta}} \boldsymbol{\alpha}^{\mathrm{T}} \mathbf{K}_x \mathbf{K}_y \boldsymbol{\beta}, \quad \textit{subject to} \quad \boldsymbol{\alpha}^{\mathrm{T}} \mathbf{K}_x^2 \boldsymbol{\alpha} = \boldsymbol{\beta}^{\mathrm{T}} \mathbf{K}_y^2 \boldsymbol{\beta} = 1, \tag{3.8}$$

where $\mathbf{K}_x$ and $\mathbf{K}_y$ are $n \times n$ centered Gram matrices and

$$\mathbf{K}_x = \begin{bmatrix} k_x(x^1, x^1) - \frac{1}{n}\sum_{j=1}^n k_x(x^1, x^j) & \cdots & k_x(x^1, x^n) - \frac{1}{n}\sum_{j=1}^n k_x(x^1, x^j) \\ \vdots & \ddots & \vdots \\ k_x(x^n, x^1) - \frac{1}{n}\sum_{j=1}^n k_x(x^n, x^j) & \cdots & k_x(x^n, x^n) - \frac{1}{n}\sum_{j=1}^n k_x(x^n, x^j) \end{bmatrix},$$
$$\mathbf{K}_y = \begin{bmatrix} k_y(y^1, y^1) - \frac{1}{n}\sum_{j=1}^n k_y(y^1, y^j) & \cdots & k_y(y^1, y^n) - \frac{1}{n}\sum_{j=1}^n k_y(y^1, y^j) \\ \vdots & \ddots & \vdots \\ k_y(y^n, y^1) - \frac{1}{n}\sum_{j=1}^n k_y(y^n, y^j) & \cdots & k_y(y^n, y^n) - \frac{1}{n}\sum_{j=1}^n k_y(y^n, y^j) \end{bmatrix}.$$

The optimization of the canonical correlation $\rho_{x,y}$ of Eq. (3.8) comes down to a generalized eigenvalue problem as follows:

$$\begin{bmatrix} O & \mathbf{K}_x\mathbf{K}_y \\ \mathbf{K}_y\mathbf{K}_x & O \end{bmatrix} \begin{bmatrix} \boldsymbol{\alpha} \\ \boldsymbol{\beta} \end{bmatrix} = \rho_{x,y} \begin{bmatrix} \mathbf{K}_x^2 & O \\ O & \mathbf{K}_y^2 \end{bmatrix} \begin{bmatrix} \boldsymbol{\alpha} \\ \boldsymbol{\beta} \end{bmatrix}.$$

As a result, the kernel canonical correlation $\rho_{x,y}$ corresponds to the maximal eigenvalue which can be derived by Cholesky decomposition [6]. Then, we employ an independent measure based on Kernel generalized variance (KGV):

$$I(x,y) = -\frac{1}{2}\log(1-\rho^2_{x,y}), \tag{3.9}$$

which is equivalent to the mutual information between the variable x and y [6]. The mutual information is zero if and only if the variables are mutually independent. Therefore, we employ KGV as our independence measure.

Though, generally, computing the higher-dimensional transformations of the variables requires high computational time and is not feasible to compute, [2] have shown that the inner products can be replaced by the kernel functions such as Gaussian kernel defined as follows:

$$k_x(x^i, x^j) = \langle \phi_x(x^i), \phi_x(x^j) \rangle = \exp\left(-\frac{(x^i - x^j)^2}{\sigma^2}\right),$$
$$k_y(y^i, y^j) = \langle \phi_y(y^i), \phi_y(y^j) \rangle = \exp\left(-\frac{(y^i - y^j)^2}{\sigma^2}\right).$$

This technique is called Kernel Trick. Therefore, we can directly obtain the inner products without computing each transformation $\phi_x(x^i)$ and $\phi_y(y^i)$ [1]. In this dissertation, we employ this Gaussian kernel as the kernel functions k_x and k_y which is widely used in the field of machine learning, and set the parameter σ to the default value used in [6] where the good performance is shown. In these manners, by using kernel functions, we can obtain more accurate independence measure of Eq. (3.9) while maintaining its computational feasibility.

As described in the previous section, the independence measure used in ICA-LiNGAM and DirectLiNGAM of Eq. (3.5) focus on kurtosis of the distribution of the variable. However, kurtosis is strongly influenced by outliers [25] which are usually contained in real-world datasets. Therefore, the robustness to outliers of the independence measure is not well ensured. In contrast, since the kernel independence measure considers various types of nonlinear transformation and focus on not only kurtosis but also other statistics, it is expected to robustly evaluate the independence [6]. Therefore, the independence measure will provide more accurate and robust estimation of the network in our variants of ICA-LiNGAM and DirectLiNGAM.

[1] Further explanation of the relation between the kernel function and the high-dimensional transformation is given in [2].

Based on these considerations, we first propose a variant of ICA-LiNGAM by replacing FastICA method used at Step 1 of Algorithm 2 by the ICA method with KGV, which is called KernelICA-KGV. Here we call this variant of ICA-LiNGAM as KernelICA-LiNGAM.

Secondly, we propose a variant of DirectLiNGAM in which the independence measure used at Step 2(a) in Algorithm 3 is replaced by the independence measure Eq. (3.9). In other words, we propose to replace the statistic Eq. (3.5) by a statistic using the kernel based independence measure Eq. (3.9) as

$$T_{kernel}(x_j, U) = \sum_{i \in U, i \neq j} I(x_j, r_i^{(j)}) = \sum_{i \in U, i \neq j} -\frac{1}{2} \log(1 - \rho^2_{x_j, r_i^{(j)}}), \tag{3.10}$$

where $\rho_{x_j, r_i^{(j)}}$ is the kernel canonical correlation coefficient between a variable x_j and its residuals $r_i^{(j)}$ when x_i is regressed on x_j by the least square regression of Eq. (3.4). If a variable x_j and its residuals $r_i^{(j)}$ are independent, this independence measure has a small value. We call the variant as Kernel-DirectLiNGAM.

3.3.2 Extending the Search Algorithm

With small sample data, the independence statistic Eq. (3.5) is inaccurate because of the sampling fluctuation. Particularly in DirectLiNGAM, it employs a simple greedy search algorithm and always selects only a unique variable as an exogenous variable that minimizes the statistic Eq. (3.5). Therefore, once a wrong exogenous variable is selected as an exogenous variable because of the inaccurate value of T, a widely wrong network tends to be obtained. To alleviate this problem, a more advanced search algorithm which always keeps multiple candidate orderings of the variables in the search is expected to provide a more accurate identification of the networks under small samples. Accordingly, we propose to introduce the search algorithm called beam search [61] at Step 2(a) of DirectLiNGAM algorithm. We point out that the objective to use the advanced search algorithm is to enhance the accuracy and not to improve the robustness against outliers because the outliers affects the independence measure only and do not change the search process.

We present pseudo code of the algorithm incorporating the beam search in Algorithm 4.[2] At the first step of the iteration, initialize $H_\ell = 0$ and $K_\ell = \emptyset$ for all $\ell \in \{1, \cdots, \kappa\}$, where κ is a width of the beam search to keep κ ordering lists K_ℓ, and H_ℓ is a value representing the summation of all statistics T along the ordering list K_ℓ. In our formulation, the total independence H_ℓ is employed since it is to be small if the correct ordering of the variables are identified. Moreover, prepare copies of the data matrix $\mathbf{X}_\ell = \mathbf{X}$. At Step 2(a) in Algorithm 4, one selects the κ pairs of the subscripts $\{\tau^{(h)}, \lambda^{(h)}\}$ $(h \in \{1, \cdots, \kappa\})$ of K_ℓ and x_j each of which is likely to be independent from its residuals and gives the smaller total independence measure $H_\ell + T(x_j, U \backslash K_\ell)$ among $j \in U \backslash K_\ell$ for all $\ell \in \{1, \cdots, \kappa\}$. At Step 2(b), the κ current ordering lists and measures are stored, *i.e.*, $K_\ell^c = K_\ell$ and $H_\ell^c = H_\ell$. Then, at Step 2(c), H_ℓ and K_ℓ are replaced by the new ordering lists $K_{\tau(\ell)}^c \cup \lambda^{(\ell)}$ and the new total independence measures $H_{\tau(\ell)}^c + T(x_{\lambda^{(\ell)}}, U \backslash K_{\tau(\ell)}^c)$, respectively. Next, at Step 2(d), the κ data matrices $\mathbf{X}_\ell$ are updated to the regressed data matrix $\mathbf{R}_{\tau(\ell)}^{\lambda(\ell)}$ which is derived by the least square regression of Eq. (3.4). In these manner, one recursively selects the κ candidate pairs of the ordering list and the exogenous variable. Finally, one selects the best ordering K from K_ℓ which has the smallest total independence measure H_ℓ and appends the remaining subscript to the end of K at Step 3. Similar to the DirectLiNGAM algorithm, the connection strength matrix $\mathbf{B}$ is constructed by ordinary least squares of Eq. (3.6) based on the ordering K at Step 4.

We show an example of this new algorithm as follows. Suppose that we have a dataset having three observed variables x_1, x_2, x_3 and define $\kappa = 2$. At first, the algorithm initializes ordering lists $K_1 = K_2 = \emptyset$ and the total independence measure $H_1 = H_2 = 0$. If the statistics of three possible exogenous variables are

$$\begin{aligned} T(x_1, U) &= 0.13, \\ T(x_2, U) &= 3.25, \\ T(x_3, U) &= 0.08, \end{aligned}$$

we obtain $\kappa = 2$ candidates of the exogenous variables, x_1 and x_3 at Step 2(a) in 1st

[2] In this dissertation, we do not introduce this search method to ICA-LiNGAM since introducing this beam search to ICA-LiNGAM seems not to be made in a straightforward way.

Algorithm 4 Beam-DirectLiNGAM

1. Given a p-dimensional variable vector $\mathbf{x}$, its $p \times n$ data matrix $\mathbf{X}$, the positive integer κ ($\kappa \leq p$) for the beam search and a set U of subscripts of all $x_i \in \mathbf{x}$, initialize κ ordering lists $K_\ell := \emptyset$, the κ total independence measures $H_\ell := 0$, κ copies of data matrix $\mathbf{X}_\ell := \mathbf{X}$ and $m := 1$.

2. Repeat until $p - 1$ subscripts are added to each K_ℓ.

   (a) For each $\ell \in \{1, \cdots, \kappa\}$, regress x_i on x_j for all $i \in U \backslash K_\ell (i \neq j)$ and derive the residual data matrix $\mathbf{R}_\ell^{(j)}$ from the data matrix $\mathbf{X}_\ell$ for all $j \in U \backslash K_\ell$ by performing the least square regression of Eq. (3.4). Then, find κ pairs of the candidate of the exogenous variable $x_{\lambda^{(h)}}$ and the ordering list $K_{\tau^{(h)}}$ which give the top κ smallest values of the total independence measure:

   $$\{\tau^{(h)}, \lambda^{(h)} | h \in \{1, \cdots, \kappa\}\} = \underset{\substack{\{\ell, j\},\\ \ell \in \{1, \cdots, \kappa\}, j \in U \backslash K_\ell}}{\text{arg top } \kappa \min} \{H_\ell + T(x_j; U \backslash K_\ell)\},$$

   where T is the independence measure shown in Eq. (3.5) and $\{\tau^{(h)}, \lambda^{(h)}\}$ are the κ candidate pairs of the subscripts of the ordering list and the variable that give top κ smallest values in the above measure.

   (b) Store the ordering lists and the total independence measures, $K_\ell^c := K_\ell$, $H_\ell^c := H_\ell$ for each $\ell \in \{1, \cdots, \kappa\}$.

   (c) For each $\ell \in \{1, \cdots, \kappa\}$, update the measure, $H_\ell = H^c_{\tau^{(\ell)}} + T(x_{\lambda^{(\ell)}}, U \backslash K^c_{\tau^{(\ell)}})$. Then, let $K_\ell = K^c_{\tau^{(\ell)}}$ and add the subscript $\lambda^{(\ell)}$ to the end of K_ℓ.

   (d) For each $\ell \in \{1, \cdots, \kappa\}$, let $\mathbf{X}_\ell := \mathbf{R}_{\tau^{(\ell)}}^{(\lambda^{(\ell)})}$ and $m := m + 1$.

3. Select the list K_ℓ having the smallest total independence measure H_ℓ as the best ordering K and add the subscript of the remaining variable to the end of K.

4. Construct the connection strength matrix $\mathbf{B}$ by ordinary least squares of Eq. (3.6) based on the ordering K.

iteration ($m = 1$), and the values of $T(x_{1_1}, U)$ and $T(x_{3_2}, U)$ are stored to H_1 and H_2 respectively at Step 2(b). Subsequently, each subscript $\lambda^{(1)} = 1$, $\lambda^{(2)}$=3 are appended to each ordering list K_1 and K_2 at Step 2(c). Now we have the ordering lists $K_1 = \{1\}$ and $K_2 = \{3\}$, and the total independence measures $H_1 = 0.13$ and $H_2 = 0.08$. Then, at Step 2(e), one updates the data matrices $\mathbf{X}_1$ and $\mathbf{X}_2$ to the residual data matrices $\mathbf{R}_1^{(1)}$ and $\mathbf{R}_2^{(3)}$ derived by performing Eq. (3.4).

Next, suppose we obtain the following statistics

$$\begin{aligned} T(x_2, U\backslash K_1) &= 0.02, \\ T(x_3, U\backslash K_1) &= 5.65, \\ T(x_1, U\backslash K_2) &= 4.85, \\ T(x_2, U\backslash K_2) &= 3.66. \end{aligned}$$

Here, we recall that if the total independence measure given by the pair of the ordering list and the candidate variable is small, the pair is more feasible ordering in our formulation. Then, at Step 2(a) in 2nd iteration ($m = 2$), the $\kappa = 2$ candidate pairs of the subscripts of the ordering list and the *exogenous* variable, $\{\tau^{(1)} = 1, \lambda^{(1)} = 2\}$ and $\{\tau^{(2)} = 2, \lambda^{(2)} = 2\}$ are obtained each of which gives the smaller total independence measure $H_\ell + T(x_j, U\backslash K_\ell)$. Subsequently, copies $K_\ell^c := K_\ell$ and $H_\ell^c := H_\ell$ are created at Step 2(b). Then, at Step 2(c), H_1, H_2 are replaced by $H_1^c + T(x_2, U\backslash K_1^c)$ and $H_2^c + T(x_2, U\backslash K_2^c)$. Moreover, the ordering lists K_1, K_2 are replaced by K_1^c and K_2^c, and each subscript $\{\lambda^{(1)} = 2\}$ and $\{\lambda^{(2)} = 2\}$ are appended to the end of K_1 and K_2. Here we have the ordering lists $K_1 = \{1, 2\}$ and $K_2 = \{3, 2\}$, and the total independence measures $H_1 = 0.15$ and $H_2 = 3.74$. Next, the ordering K_1 is selected as the best ordering K because of the smallest measure $H_1 = 0.15$. Then, the remaining subscript are appended to the ordering list K, and we obtain $K = \{1, 2, 3\}$ at Step 3. Finally, the connection coefficient matrix $\mathbf{B}$ is derived by ordinary least squares of Eq. (3.6) based on the obtained ordering K at Step 4.

Fig. 3.2 shows an illustration of this procedure to select $\kappa = 2$ pairs of the ordering and the variable by the beam search algorithm. As can be seen in Fig. 3.3, if we use the past DirectLiNGAM algorithm, the other ordering $K = \{3, 2, 1\}$ is resulted which has the large total independence measure because of the small difference of $T(x_1, U)$

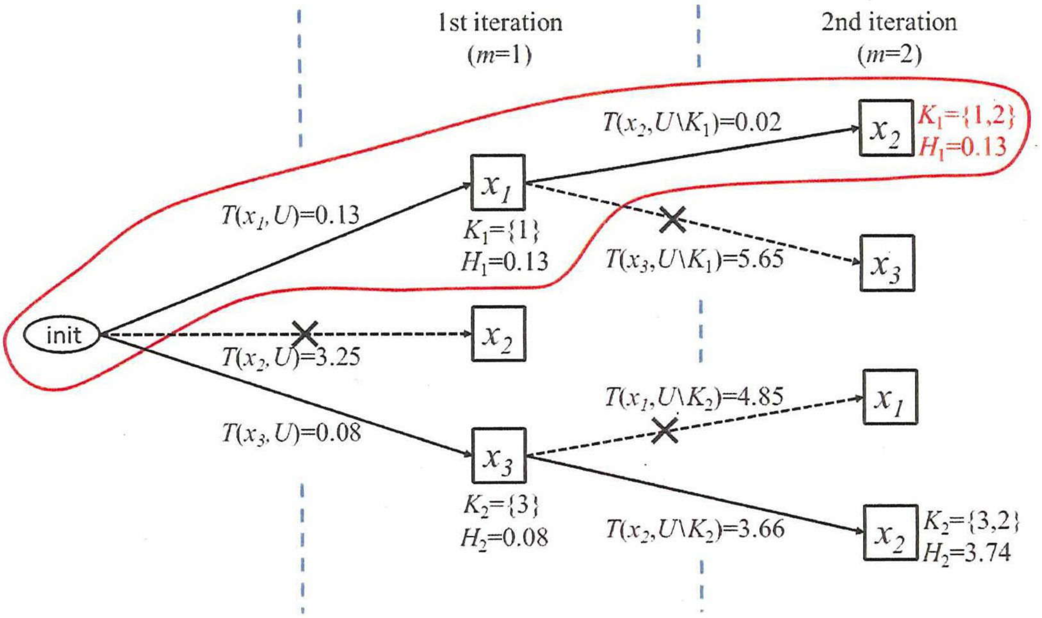


Figure 3.2: The procedure to select the κ = 2 candidate pairs of the ordering and the variable by the beam search algorithm.

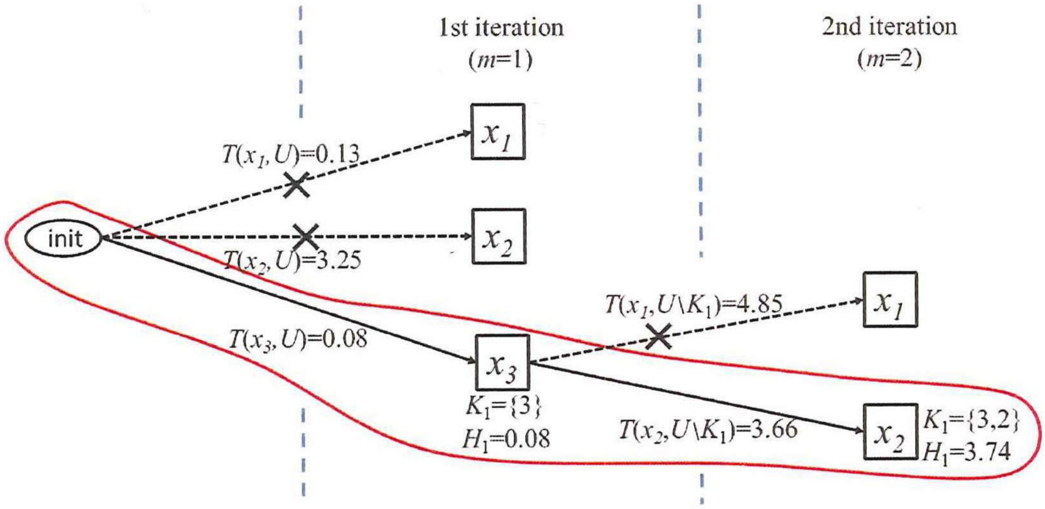


Figure 3.3: The procedure to select the candidate of the exogenous variable by DirectLiNGAM (Beam-DirectLiNGAM with κ = 1).

and $T(x_3, U)$ in 1st iteration $(m = 1)$. In this manner, the small difference of T which could be caused by the statistical sampling fluctuation would lead the quite different estimation of the network. Thus, the modification for the search algorithm is important for more accurate network identification in DirectLiNGAM.

We call the variant of DirectLiNGAM using the beam search as Beam-DirectLiNGAM. Additionally, we apply the beam search to Kernel-DirectLiNGAM that we proposed in the previous subsection and call this variant as Beam-Kernel-DirectLiNGAM.

3.4 Experiments on Artificial Data

In this section, we experimentally characterized the conventional methods, ICA-LiNGAM, and DirectLiNGAM, and their variants, KernelICA-LiNGAM, Kernel-DirectLiNGAM, Beam-DirectLiNGAM and Beam-Kernel-DirectLiNGAM. To design the experiments in an efficient way, we partitioned the experiments into two stages. Firstly, we compared ICA-LiNGAM, DirectLiNGAM, KernelICA-LiNGAM and Kernel-DirectLiNGAM to investigate the accuracy, the computational cost, and the robustness to outliers which is expected to be provided by the kernel based independence measure. Further, we examine the scale-invariance of the framework of DirectLiNGAM. Secondly, based on the result of the previous experiments, we compared Beam-DirectLiNGAM and Beam-Kernel-DirectLiNGAM in terms of accuracy and computational cost to examine the effect to the accuracy by the beam search.

3.4.1 Experimental setup

We explain how artificial datasets are generated and how the accuracy of the methods is evaluated.

At first, we employed 17 non-Gaussian distributions used in [6] from which we drew independent non-Gaussian external influences e_i. These distributions included a double exponential distribution, an uniform distribution, a t-distribution with 5 degrees of freedom, an exponential distribution, mixtures of two exponential distributions, symmetric and asymmetric mixtures of some Gaussian distributions. Then we randomly generated a dataset with a combination of number of variables p and sample size n as follows.

1. We randomly constructed $p \times p$ strictly lower triangular matrix $\mathbf{B}$ so that standard deviations of variables x_i, owing to parent variables ranged in the interval [0.5, 1.5].

2. We generated n samples by independently drawing the external influence e_i $(i = 1, \cdot, p)$ from non-Gaussian distributions randomly selected from the 17 distributions with zero means and standard deviations randomly selected from [0.5, 1.5].

3. The n sample values of the observed variables x_i were generated according to the LiNGAM model Eq. (3.2) with n samples of the external influences.

4. We randomly permuted the ordering of x_i, *i.e.*, obtained the row-permuted data matrix $\mathbf{X}$.

Because of the permutation at Step 4, the true connection strength matrix to be estimated is also permuted by the corresponding ordering of x_i. Then, we denote the permuted matrix as $\mathbf{B}_{perm}$.

In each numerical experiment, we evaluated accuracy of an estimated ordering as follows. We first permuted the rows and columns of $\mathbf{B}_{perm}$ according to the estimated ordering K. If the estimated ordering corresponds to the true ordering, the permuted $\mathbf{B}_{perm}$ is strictly lower triangular. Thus, we counted the number of non-zero elements in the strictly upper triangular part of the permuted $\mathbf{B}_{perm}$ as the number of errors. The number of errors is zero if the estimated ordering is correct. In all the experiments for every combination of p and n, we generated 101 datasets and counted the number of errors on each dataset and took the median of the 101 numbers of errors. In comparing the computational time of the methods, we took the median computational time of the 101 trials.

3.4.2 Kernel-based variants

At first, we tested two of the variants, KernelICA-LiNGAM and Kernel-DirectLiNGAM, and made a comparison with the other LiNGAM-model-based methods. We generated datasets with combinations of the number of the variables p=8, 16 and 32, and the sample size n=100, 200, 500, 1000, 2000 and 5000. We evaluated accuracy of orderings estimated by those four methods using the datasets. Further, we compared their computational time. We did not test KernelICA-LiNGAM for p=16 and 32 since it needs much larger computational time than other methods. The medians of the numbers of errors are shown in Table 3.1. In Table 3.1, the median errors of Kernel-DirectLiNGAM are often smallest. This is because the kernel based independence measure considers the various nonlinear transformation and evaluates the independence correctly. Furthermore, the computational times are shown in Table 3.2. The computation amount of the Kernel-DirectLiNGAM is rather larger than DirectLiNGAM. However, its computation amount is considered to be still tractable for data consisting of dozens of variables and a few thousand samples.

Next, we examined scale-invariance of ICA-LiNGAM, DirectLiNGAM, KernelICA-LiNGAM and Kernel-DirectLiNGAM since the algorithm of ICA-LiNGAM is known to be scale-sensitive while that of DirectLiNGAM is scale-invariant as explained before. We first generated datasets with combinations of $p = 8$ and n=100, 200, 500, 1000, 2000 and 5000, then all the values of four randomly chosen variables from the p variables were respectively amplified by two orders of magnitude in each dataset so that the variables have rather different scales. Table 3.3 shows the median errors of the four methods. DirectLiNGAM and Kernel-DirectLiNGAM are advantageous over the other two methods in terms of scale-invariance. Though both DirectLiNGAM and Kernel-DirectLiNGAM are scale-invariant, Kernel-DirectLiNGAM worked best since the kernel based independence measure considers various nonlinear transformations of the variables and evaluates the independence more accurately.

Finally, we examined robustness against outliers. We first generated datasets with a combination of $p = 8$ and 1000. Then we added a random value having either +5 or -5 up to randomly chosen 14 samples [6]. The median errors resulted by these experiments were plotted in Fig. 3.4. It shows that Kernel-DirectLiNGAM achieved the smaller number of errors and was not very much affected by the existence of outliers. As described in Section 3.3, the kernel based independence measure considers various statistics while the conventional measure focus on kurtosis of the variable which usu-

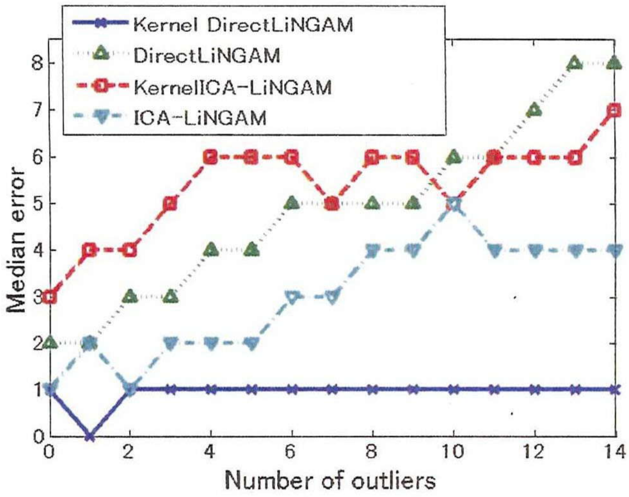


Figure 3.4: Median numbers of errors with increasing the number of outliers.

ally suffers from outliers. Therefore, Kernel-DirectLiNGAM can estimate the network robustly.

As a result, we can conclude Kernel-DirectLiNGAM is more accurate and robust than the other methods. Moreover, Kernel-DirectLiNGAM is also scale-invariant.

3.4.3 Variants employing Beam search

In this subsection, based on the result of the previous experiment, we focused on DirectLiNGAM and its variant, Kernel-DirectLiNGAM which provides more accurate and tractable network identification than ICA-based methods. In addition, as described in Section 3.3, the beam search is expected to enhance the accuracy of the network estimation. Therefore, we investigated differences of the accuracy and the computational time between the greedy search and the beam search in DirectLiNGAM and Kernel-DirectLiNGAM. We generated 101 datasets with combinations of p=8, 16 and n=100, 200, 500, 1000, 2000, 5000, and selected the width of the beam κ=2, 4 and 8 which is the number to keep κ candidate orderings. In Table 3.4, the median numbers of errors are shown. Moreover, the median computational times of the compared methods are presented in Table 3.5. Table 3.4 shows that the variants using the beam search more accurately estimated the orderings of the observed variable. By using the kernel based independence measure, the more accurate total independence of the ordering is evaluated in the beam search, and thus the ordering is more correctly estimated even under the situation having small samples. Therefore, the significantly accurate identification of the ordering was made by Beam-Kernel-DirectLiNGAM. However, as can be seen in Table 3.5, the computational time of Beam-Kernel-DirectLiNGAM is highest of all and increases linearly with the width of the beam κ. There is a trade-off between the accuracy and the computational time. Nevertheless, we can control the computational time to be feasible by choosing the width based on the characteristics of given datasets (e.g. the number of observed variables, sample size and/or required accuracy) in applying the method to real-world datasets.

3.5 Conclusion

We proposed two ideas to improve accuracy and robustness of the conventional LiNGAM-model-based methods. One is to use a more sophisticated independence measure than that in ICA-LiNGAM and DirectLiNGAM, which provides both accuracy and robustness to outliers. The other is to use beam search instead of greedy search which enhances the accuracy of DirectLiNGAM. In the experiments, we firstly examined the LiNGAM-model-based methods and our methods using kernel based independence measure in terms of the accuracy, the computational cost, the robustness to outliers and scale-invariance. Based on the result of the first experiment, we compared the accuracy and the computational cost of DirectLiNGAM, Kernel-DirectLiNGAM, Beam-DirectLiNGAM and Beam-Kernel-DirectLiNGAM. The result of these numerical experiments implies that the variant using both kernel method and the beam search provides the more accurate and robust network identification than the previous LiNGAM-model-based methods even under the various real-world situations such as small sample and noisy data. Though the computational time of Beam-Kernel-DirectLiNGAM is higher than the conventional methods, it is tractable and controllable.

An important topic for future research is to investigate how other meta-heuristics including tabu search could be used in our method. Moreover, we can incorporate other independence measures such as Fast kernel ICA [44] into our method. These further investigation could lead more accurate and/or time-efficient methods for estimating the variable network.

Table 3.1: Median errors of the conventional methods based on the LiNGAM model and their variants under (A) 8 variables; (B) 16 variables; (C) 32 variables.

(A) 8 variables

n	Kernel-Direct LiNGAM	Direct LiNGAM	KernelICA-LiNGAM	ICA-LiNGAM
100	*6*	7	7	8
200	*4*	6	8	4
500	*1*	3	6	3
1000	*1*	2	3	1
2000	0	0	0	0
5000	0	0	0	0

(B) 16 variables

n	Kernel-Direct LiNGAM	Direct LiNGAM	KernelICA-LiNGAM	ICA-LiNGAM
100	30	31	-	*29*
200	*18*	24	-	33
500	*7*	14	-	33
1000	*4*	8	-	22
2000	*2*	4	-	14
5000	*1*	*1*	-	6

(C) 32 variables

n	Kernel-Direct LiNGAM	Direct LiNGAM	KernelICA-LiNGAM	ICA-LiNGAM
100	145	130	-	*125*
200	93	*87*	-	88
500	*54*	80	-	161
1000	*29*	41	-	157
2000	*12*	27	-	138
5000	*5*	11	-	56

Table 3.2: Median computational time (sec) to estimate the ordering by the conventional methods and their variants under (A) 8 variables; (B) 16 variables; (C) 32 variables.

(A) 8 variables

n	Kernel-Direct LiNGAM	Direct LiNGAM	KernelICA-LiNGAM	ICA-LiNGAM
100	0.63	*0.04*	19.68	1.30
200	0.70	*0.04*	22.60	1.18
500	0.90	*0.05*	29.41	1.15
1000	1.24	*0.07*	39.14	1.18
2000	2.01	*0.11*	41.85	1.18
5000	5.53	*0.27*	77.58	1.19

(B) 16 variables

n	Kernel-Direct LiNGAM	Direct LiNGAM	KernelICA-LiNGAM	ICA-LiNGAM
100	4.94	*0.27*	-	0.54
200	5.61	*0.30*	-	0.66
500	7.18	*0.39*	-	0.94
1000	10.05	*0.55*	-	0.59
2000	16.09	0.86	-	*0.41*
5000	43.02	1.96	-	*0.40*

(C) 32 variables

n	Kernel-Direct LiNGAM	Direct LiNGAM	KernelICA-LiNGAM	ICA-LiNGAM
100	40.52	1.99	-	*0.97*
200	45.79	2.25	-	*1.44*
500	58.17	2.99	-	*1.82*
1000	81.97	4.27	-	*2.47*
2000	131.48	6.78	-	*3.54*
5000	346.20	17.07	-	*1.90*

Table 3.3: Median errors with the different scale variables.

n	Kernel-Direct LiNGAM	Direct LiNGAM	KernelICA-LiNGAM	ICA-LiNGAM
100	*6*	13	19	19
200	*4*	11	19	17
500	*1*	8	18	17
1000	*1*	5	17	16
2000	*0*	2	16	16
5000	*0*	1	16	16

Table 3.4: Median errors of the variants using the beam search with the width of the beam κ=2, 4 and 8 under (A) 8 variables; (B) 16 variables.

(A) 8 variables

n	Direct LiNGAM	Beam-DirectLiNGAM			Kernel-Direct LiNGAM	Beam-Kernel DirectLiNGAM		
		$(\kappa = 2)$	$(\kappa = 4)$	$(\kappa = 8)$		$(\kappa = 2)$	$(\kappa = 4)$	$(\kappa = 8)$
100	7	7	8	7	6	*5*	*5*	*5*
200	6	6	6	7	4	*3*	*3*	*3*
500	3	2	2	2	*1*	*1*	*1*	*1*
1000	2	1	1	1	1	*0*	*0*	*0*
2000	0	0	0	0	0	0	0	0
5000	0	0	0	0	0	0	0	0

(B) 16 variables

n	Direct LiNGAM	Beam-DirectLiNGAM			Kernel-Direct LiNGAM	Beam-Kernel-DirectLiNGAM		
		$(\kappa = 2)$	$(\kappa = 4)$	$(\kappa = 8)$		$(\kappa = 2)$	$(\kappa = 4)$	$(\kappa = 8)$
100	31	29	29	29	30	27	22	*21*
200	24	23	20	20	18	15	15	*14*
500	14	13	12	12	7	6	*5*	*5*
1000	8	5	5	5	4	3	3	*2*
2000	4	3	3	2	2	2	*1*	*1*
5000	1	1	1	1	1	*0*	*0*	*0*

Table 3.5: Median computational time (sec) of the variants using the beam search with the width of the beam κ=2, 4 and 8 under (A) 8 variables; (B) 16 variables.

(A) 8 variables

n	Direct LiNGAM	Beam-DirectLiNGAM			Kernel-Direct LiNGAM	Beam-Kernel DirectLiNGAM		
		($\kappa = 2$)	($\kappa = 4$)	($\kappa = 8$)		($\kappa = 2$)	($\kappa = 4$)	($\kappa = 8$)
100	*0.04*	0.09	0.17	0.35	0.80	1.60	3.20	6.42
200	*0.05*	0.09	0.18	0.37	0.94	1.88	3.77	7.55
500	*0.06*	0.12	0.24	0.48	1.33	2.65	5.24	10.66
1000	*0.08*	0.16	0.33	0.66	1.90	3.83	7,71	15.47
2000	*0.13*	0.27	0.53	1.07	3.12	6.22	12.59	25.53
5000	*0.29*	0.57	1.15	2.30	10.74	21.90	43.95	88.36

(B) 16 variables

n	Direct LiNGAM	Beam-DirectLiNGAM			Kernel-Direct LiNGAM	Beam-Kernel-DirectLiNGAM		
		($\kappa = 2$)	($\kappa = 4$)	($\kappa = 8$)		($\kappa = 2$)	($\kappa = 4$)	($\kappa = 8$)
100	*0.30*	0.59	1.19	2.38	6.52	13.00	25.95	51.94
200	*0.33*	0.66	1.33	2.66	7.55	15.07	30.14	60.31
500	*0.44*	0.87	1.75	3.51	10.73	20.98	42.90	86.35
1000	*0.63*	1.26	2.51	5.04	15.11	30.14	61.00	122.70
2000	*1.03*	2.05	4.10	8.21	25.36	50.52	101.78	205.07
5000	*2.44*	4.87	9.74	19.49	86.68	170.83	344.21	696.70

Chapter 4

Robust Active Learning for Linear Regression via Density Power Divergence

4.1 Introduction

In Chapter 2 and 3, we proposed the methods to obtain some knowledge on the directed variable network. Besides them, estimating a relation between an important variable and the other observed variables, which is known as a linear regression, is also important in many applications. In the context of the linear regression, the particular variable is called a label variable and otherwise are called explanatory variables. Here, we note that a sample which has values associated with both the label variable and the explanatory variables is called a labeled sample. Otherwise, a sample which has only values of the explanatory variables is called an unlabeled sample. In contrast to the technique for estimating the entire network shown in Chapter 3 which can apply only to less than 100 dimensional data, the linear regression technique can be applied to more than 100 dimensional data. Because of its applicability, the linear regression model is widely used to represent the relation between the label variable and the other explanatory variables in many domains such as medical service [60, 30], social science [55, 17], marketing [20, 9] and so on.

Recent development of information technology has made it possible to collect huge amount of data automatically in various domains. Nevertheless, in most cases, such data are composed of majority unlabeled samples and minority labeled samples. This is because labeling tasks by human experts or additional experiments called oracles are usually expensive or time-consuming. For example, in a car insurance company, an insurance fee is determined by its company's employees based on car information, driver's driving records and so on. However, such determination by hand needs enormous cost and time. Unfortunately, under the small labeled sample data, the estimation of the linear regression model is often statistically unreliable. For this issue, a technique called active learning has been discussed to make learning processes with majority unlabeled samples and minority labeled samples more efficient [10] in recent years. In contrast to passive learning that estimates a model from given labeled samples only, the active learning algorithm selects some unlabeled samples expected to be informative as queries for learning and asks an oracle to label them. This active learning framework has been widely applied successfully in various regions such as speech recognition [19], classification [35] and regression [56].

One of the most important problems in the active learning framework is how to select unlabeled samples called queries, and several querying measures have been discussed over the last few decades [31, 42]. These conventional active learning methods commonly assume that the oracle always follows a true labeling distribution and gives correct labels on samples. In the real-world, however, human experts might give incorrect labels because of their conditions or additional experiments might make mistakes because of their environments. Such an oracle giving noisy labels is called a noisy oracle which usually follows the noisy labeling distribution called the contaminated distribution. With the noisy oracle, an accuracy of a model estimation by the active learning method could become worse. Thus, in this chapter, we propose a new active learning algorithm for the linear regression to tackle this problem caused by a noisy oracle.

Among various types of querying measures, in this chapter, we employ Variance Reduction Approach (VRA) [41], which is based on an asymptotic variance of parameters (estimators) since its validity is well ensured by the statistical asymptotic analysis. In this approach, active learning algorithms select queries that are expected to minimize the difference between the true parameter and the estimated parameter. A conventional method based on VRA use Kullback-Leibler (KL) divergence in estimating a model parameter (ML-estimator), and employ the asymptotic variance based on the

ML-estimator in determining queries [62]. However, the KL-divergence-based methods do not consider noisy-oracles and thus work worse if there are noisy labels. Therefore, in this chapter, through the asymptotic analysis on M-estimator which is a wider class of estimators including the ML-estimator, we extend the conventional VRA-based querying measure and incorporate robust divergences called density power divergence into our querying measure to achieve the robust estimation of the linear regression model.

Based on these backgrounds, in this chapter, we firstly propose an active learning method to robustly estimate the variable relation from the noisy small labeled samples and the large unlabeled samples. Then, we examine robustness of our proposed methods by the numerical experiments with artificial datasets. Further, by using real-world dataset, we investigate its behavior under more realistic situation.

The remainder of the chapter is organized as follows. In Section 4.2, we first briefly review the linear regression model, the pool-based active learning framework and the conventional active learning method based on VRA. In Section 4.3, we extend VRA through an asymptotic analysis on M-estimator and apply it to the density power divergence. Then, in Section 4.4, we propose a practical querying measure based on the discussion in the previous section. Finally, we investigate the robustness of our active learning method for the linear regression model by using artificial and real-world datasets in Section 4.5, and we conclude this chapter in Section 4.6. This chapter is related to the work published in [51, 52].

4.2 Background

4.2.1 Linear Regression Model

As we explained in Section 4.1, the linear regression model has been widely used to represent the relation between the label variable and the explanatory variables because of its applicability. Therefore, in this chapter, we discuss the general linear regression model.

Let us denote the label and the explanatory variables by the continuous scalar variable y and p-dimensional continuous vector $\mathbf{x}$, respectively. Then, the linear relation is defined as the following equation:

$$y = \mathbf{w}^\top \mathbf{x} + w_0 + \epsilon,$$

where $\mathbf{w}$ is a p-dimensional coefficient vector, w_0 is a constant term and ϵ is a Gaussian noise with zero mean and its variance σ^2. The probabilistic model of this linear regression model is expressed as:

$$\epsilon \sim p(y|\mathbf{x}; \mathbf{w}, w_0, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(y - \mathbf{w}^\top \mathbf{x} - w_0)^2}{\sigma^2}\right).$$

In the rest of this chapter, we denote the collection of these parameters by $\boldsymbol{\theta}$ and the probabilistic model of the linear regression by $p_{\boldsymbol{\theta}}(y|\mathbf{x})$ for simplicity.

4.2.2 Pool-based Active Learning

Active learning techniques are divided into main three branches. The first is Membership query Synthesis [4] where the oracle generates any arbitrary unlabeled sample on demand and give the label on it. The second is stream-based selective active learning [2] where unlabeled sample is generated sequentially from its true distribution and the oracle decides whether to select as a query or discard it. The third is pool-based active learning [33] which is a frequently-discussed framework in machine learning for situations where the distribution of unlabeled samples is unknown but unlabeled samples from their true distribution are given [35]. To estimate the linear regression model, the pool-based active learning is more appropriate to real-world situations where we have small labeled samples and large unlabeled samples. Therefore, in this section, let us consider the pool-based active learning framework to robustly estimate the linear relationship (the linear regression model) from small labeled samples and large unlabeled samples.

Formally, in pool-based active learning framework, it is assumed that one has a small set of labeled samples $\mathcal{L} = \{(\mathbf{x}_1, y_1), \cdots, (\mathbf{x}_{n_l}, y_{n_l})\}$ and a large set of unlabeled samples $\mathcal{U} = \{\mathbf{x}_{n_l+1}, \cdots, \mathbf{x}_{n_l+n_u}\}$ $(n_l \ll n_u)$. Then, one tries to find a set of queries from $\mathcal{U}$ that is expected to be informative for estimating a 'good' model. An overall procedure of the pool-based active learning algorithm is described in Algorithm 5. At Step 2(a), a

Algorithm 5 Pool-based active learning algorithm

```
1. Given a set of labeled samples $\mathcal{L}$, a set of unlabeled samples $\mathcal{U}$, the number of queries per an iteration $Q$ and the number of querying iteration $R$, and initialize $m := 0$.
2. Repeat the following procedure $R$ times.
    (a) Estimate the model $p_{\boldsymbol{\theta}}(y|\mathbf{x})$ from the labeled sample set $\mathcal{L}$.
    (b) Select a set $\mathcal{S}$ of $Q$ unlabeled samples as queries based on the estimated model $p_{\hat{\boldsymbol{\theta}}_n^{(m)}}(y|\mathbf{x})$.
    (c) An oracle gives a label on each query in $\mathcal{S}$.
    (d) Add the labeled sample set $\mathcal{S}$ to $\mathcal{L}$.
    (e) Remove the queries in $\mathcal{S}$ from $\mathcal{U}$ and $m := m + 1$.
3. Estimate the model $p_{\boldsymbol{\theta}}(y|\mathbf{x})$ from the labeled sample set $\mathcal{S}$ and obtain the final model $p_{\hat{\boldsymbol{\theta}}_n}(y|\mathbf{x})$.
```

model with parameter $\boldsymbol{\theta}$, denoted as $p_{\boldsymbol{\theta}}(y|\mathbf{x})$, is estimated from the small set of labeled samples $\mathcal{L}$. Next, based on the estimated model $p_{\hat{\boldsymbol{\theta}}_n}(y|\mathbf{x})$, the algorithm selects the most 'informative' subset of unlabeled samples $\mathcal{S}$ as queries at Step 2(b). Subsequently, each query is labeled by an oracle and added to $\mathcal{L}$ as labeled samples at Step 2(c) and (d). Then, the samples in $\mathcal{S}$ are removed from $\mathcal{U}$ at Step 2(e). These learning and querying steps are repeated iteratively.

As mentioned above, the selection of an informativeness measure for queries is an important problem in developing pool-based active learning algorithms. One of the promising measures is based on the asymptotic variance which evaluates an efficiency of an estimator $\boldsymbol{\theta}$. The strategy for minimizing this asymptotic variance is known as VRA [41]. However, a conventional method, which will be explained in the next subsection, does not consider mis-labeled samples.

4.2.3 A Conventional Method using KL-divergence

In this subsection, we review the conventional active learning method based on VRA [62]. This active learning method estimates a model under an assumption that the model $p_{\boldsymbol{\theta}}(y|\mathbf{x})$ with a parameter $\boldsymbol{\theta}$ includes a true distribution $q(y|\mathbf{x})$, *i.e.*, $q(y|\mathbf{x}) = p_{\boldsymbol{\theta}^*}(y|\mathbf{x})$ where $\boldsymbol{\theta}^*$ is a true parameter. The model parameter $\boldsymbol{\theta}$ is obtained by minimizing the KL-divergence:

$$D_{KL}(q||p_{\boldsymbol{\theta}}) = \iint q(\mathbf{x})q(y|\mathbf{x}) \log \frac{q(y|\mathbf{x})}{p_{\boldsymbol{\theta}}(y|\mathbf{x})} \mathrm{d}y\mathrm{d}\mathbf{x}, \tag{4.1}$$

where $q(\mathbf{x})$ is a true input distribution of the explanatory variables. This is a well-known statistical measure to evaluate a difference between two probabilistic distributions. Now, suppose that we have n labeled samples $\{(\mathbf{x}_1, y_1), \cdots, (\mathbf{x}_n, y_n)\}$ generated from a true distribution $q(\mathbf{x}, y) = q(\mathbf{x})p_{\boldsymbol{\theta}^*}(y|\mathbf{x})$. Then, the model parameter $\hat{\boldsymbol{\theta}}_n$ which minimizes the above KL-divergence is obtained by solving the following equation:

$$\sum_{i=1}^{n} \partial_{\boldsymbol{\theta}} \log p(y_i|\mathbf{x}_i; \hat{\boldsymbol{\theta}}_n) = \mathbf{0}, \tag{4.2}$$

where $\partial_{\boldsymbol{\theta}}$ denotes the partial derivation with respect to $\boldsymbol{\theta}$. The left side of this equation is derived by the derivative of the KL-divergence with respect to the parameter and replacing the expectation over $q(\mathbf{x}, y)$ with the samples, *i.e.*, $\int q(\mathbf{x}, y)h(\mathbf{x}, y)\mathrm{d}y\mathrm{d}\mathbf{x}$ to $\frac{1}{n}\sum_{i=1}^{n} h(\mathbf{x}_i, y_i)$, where $h(\mathbf{x}, y)$ is some function of $\mathbf{x}$ and y. The parameter $\hat{\boldsymbol{\theta}}_n$ estimated by solving Eq. (4.2) is called a maximum likelihood estimator (ML-estimator) and converges to $\boldsymbol{\theta}^*$ sufficiently if $n \to \infty$. An estimator which converges to the true parameter with infinite samples is called a consistent estimator in statistics.

The conventional method selects queries based on the asymptotic variance of the ML-estimator, $\mathbb{E}_{\boldsymbol{\theta}^*}[(\hat{\boldsymbol{\theta}}_n - \boldsymbol{\theta}^*)(\hat{\boldsymbol{\theta}}_n - \boldsymbol{\theta}^*)^\top]$, where $\mathbb{E}_{\boldsymbol{\theta}^*}[\cdot]$ is the expectation over a set of $\{\mathbf{x}, y\}$ with respect to $q(\mathbf{x}, y) = q(\mathbf{x})p_{\boldsymbol{\theta}^*}(y|\mathbf{x})$, and selects queries to minimize the difference between $\hat{\boldsymbol{\theta}}_n$ and $\boldsymbol{\theta}^*$. This measure is called an asymptotic variance and corresponds to the Fisher information $I(\boldsymbol{\theta})$ [62]:

$$\mathbb{E}_{\boldsymbol{\theta}^*}[(\hat{\boldsymbol{\theta}}_n - \boldsymbol{\theta}^*)(\hat{\boldsymbol{\theta}}_n - \boldsymbol{\theta}^*)^\top] = I(\boldsymbol{\theta})^{-1} = \mathbb{E}_{\boldsymbol{\theta}^*}\left[\partial_{\boldsymbol{\theta}} \log p(y|\mathbf{x}) \partial_{\boldsymbol{\theta}} \log p(y|\mathbf{x})^\top\right]^{-1}. \tag{4.3}$$

By using KL-divergence, we can estimate the model parameter from finite samples efficiently [22]. However, it is known that such efficient estimation is strongly affected by

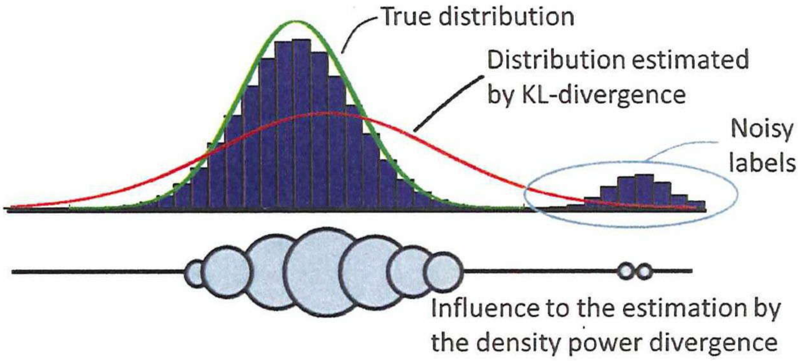

True distribution
Distribution estimated by KL-divergence
Noisy labels
Influence to the estimation by the density power divergence

Figure 4.1: An illustration of the contaminated distribution and the weighted likelihood estimator

the existence of noisy labeled samples. We show an illustration of the model estimation based on the KL-divergence in Fig. 4.1. As can be seen in Fig. 4.1, the estimation by KL-divergence fits all the samples including noisy samples and provides a wrong distribution. In these manners, if the sample with the noisy label exists, the method based on the ML-estimator tends to overfit to the data involving the noisy samples and therefore to behave worse.

4.3 Extending a Querying Measure by Asymptotic Analysis

In this section, we extend the conventional VRA scheme to utilize the other consistent estimator which are based on more robust divergences against the noisy labels than KL-divergence. Here, we recall that the consistent estimator converges to the true parameter if infinite samples are given from the true distribution. A general class of such the consistent estimators is called *M-estimator* in statistics.

In Section 4.3.1, we show the notion of the M-estimators and their statistical characteristics, which are basis of our querying measure. In Section 4.3.2, we introduce robust estimators based on the density power divergence, and propose new querying measures which provide us the robust estimation of the model in the noisy oracle situations.

4.3.1 Asymptotic Analysis on M-estimator

The M-estimator is a general class of the consistent estimators which includes the ML-estimator shown in Section 4.2.3. Through a discussion of the general class of estimators, we describe common statistical characteristics of various M-estimators.

Suppose we have *i.i.d.* n labeled samples $\{(\mathbf{x}_1, y_1), \cdots, (\mathbf{x}_n, y_n)\}$ generated from a distribution $q(\mathbf{x}, y) = q(\mathbf{x})q(y|\mathbf{x}) = q(\mathbf{x})p_{\boldsymbol{\theta}^*}(\mathbf{x}, y)$. Now, let us denote a vector function by $\boldsymbol{\psi}(y|\mathbf{x};\boldsymbol{\theta})$, the dimensionality of which corresponds to that of the parameter $\boldsymbol{\theta}$. The vector function is called an *estimating function* when it satisfies the following conditions for any $\boldsymbol{\theta}$:

$$\mathbb{E}_{\boldsymbol{\theta}}\left[\boldsymbol{\psi}(y|\mathbf{x};\boldsymbol{\theta})\right] = \mathbf{0}, \tag{4.4}$$

$$\det\left|\mathbb{E}_{\boldsymbol{\theta}}\left[\partial_{\boldsymbol{\theta}}\boldsymbol{\psi}(y|\mathbf{x};\boldsymbol{\theta})\right]\right| \neq \mathbf{0}, \tag{4.5}$$

$$\mathbb{E}_{\boldsymbol{\theta}}\left[\|\boldsymbol{\psi}(y|\mathbf{x};\boldsymbol{\theta})\|^2\right] < \infty, \tag{4.6}$$

where $\mathbb{E}_{\boldsymbol{\theta}}[\cdot]$ and $\det|\cdot|$ denote the expectation with respect to $p_{\boldsymbol{\theta}}(\mathbf{x}, y) = q(\mathbf{x})p_{\boldsymbol{\theta}}(y|\mathbf{x})$ and a determinant of the matrix, respectively. Here, we note that $\partial_{\boldsymbol{\theta}}\boldsymbol{\psi}(y|\mathbf{x};\boldsymbol{\theta})$ is the square matrix, where the numbers of row/column is equal to the dimensionality of the parameter $\boldsymbol{\theta}$. If the estimating function exists, an estimator $\hat{\boldsymbol{\theta}}_n$ is obtained by solving the following estimating equation:

$$\sum_{i=1}^{n}\boldsymbol{\psi}(y_i|\mathbf{x}_i;\hat{\boldsymbol{\theta}}_n) = \mathbf{0}. \tag{4.7}$$

A solution of Eq. (4.7) is called an M-estimator in statistics [22]. The following proposition states a convergence of the M-estimator, $\hat{\boldsymbol{\theta}}_n \to \boldsymbol{\theta}^*$ if $n \to \infty$ (consistency) and an existence of its asymptotic variance.

Proposition 2 *Suppose we have i.i.d. n labeled samples $\{(\mathbf{x}_1, y_1), \cdots, (\mathbf{x}_n, y_n)\}$ generated from a distribution $q(\mathbf{x}, y)$ and a function $\boldsymbol{\psi}(y|\mathbf{x};\boldsymbol{\theta})$ satisfies the conditions* (4.4)-(4.6). *Then, if $n \to \infty$, the M-estimator $\hat{\boldsymbol{\theta}}_n$ converges to $\boldsymbol{\theta}^*$ in probability. Moreover,*

$$\sqrt{n}(\hat{\boldsymbol{\theta}}_n - \boldsymbol{\theta}^*) \sim \mathcal{N}(\mathbf{0}, \mathbf{A}_{\boldsymbol{\theta}^*}^{-1}\mathbf{M}_{\boldsymbol{\theta}^*}(\mathbf{A}_{\boldsymbol{\theta}^*}^{-1})^\top), \tag{4.8}$$

where $\mathcal{N}(\boldsymbol{\mu}, \Sigma)$ denotes a Gaussian distribution with mean $\boldsymbol{\mu}$ and covariance matrix Σ and

$$\mathrm{A}_{\boldsymbol{\theta}^*} = \mathbb{E}_{\boldsymbol{\theta}^*}\left[\partial_{\boldsymbol{\theta}}\psi(y|\mathrm{x};\boldsymbol{\theta}^*)\right], \tag{4.9}$$

$$\mathrm{M}_{\boldsymbol{\theta}^*} = \mathbb{E}_{\boldsymbol{\theta}^*}\left[\psi(y|\mathrm{x};\boldsymbol{\theta}^*)\psi(y|\mathrm{x};\boldsymbol{\theta}^*)^{\top}\right]. \tag{4.10}$$

The proof of this proposition is given in [59]. Proposition 2 remarks if we find an estimating function $\psi(y|\mathrm{x};\boldsymbol{\theta})$, we can obtain an M-estimator $\hat{\boldsymbol{\theta}}_n$ with the asymptotic variance:

$$\mathbb{E}_{\boldsymbol{\theta}^*}[(\hat{\boldsymbol{\theta}}_n - \boldsymbol{\theta}^*)(\hat{\boldsymbol{\theta}}_n - \boldsymbol{\theta}^*)^{\top}] = \frac{1}{n}\mathrm{A}_{\boldsymbol{\theta}^*}^{-1}\mathrm{M}_{\boldsymbol{\theta}^*}(\mathrm{A}_{\boldsymbol{\theta}^*}^{-1})^{\top}. \tag{4.11}$$

For example, if $\psi(y|\mathrm{x};\boldsymbol{\theta})=\partial_{\boldsymbol{\theta}}\log p_{\boldsymbol{\theta}}(y|\mathrm{x})$, it satisfies the conditions (4.4)-(4.6) and the M-estimator $\hat{\boldsymbol{\theta}}_n$ given by Eq. (4.7) corresponds to the ML-estimator. Further, the variance Eq. (4.11) corresponds to the inverse of Fisher information matrix of Eq. (4.3). In these manners, the results in Proposition 2 allow us to generalize the conventional VRA scheme so as to utilize not only the ML-estimator, but also any M-estimators.

4.3.2 Density Power Divergence

As described in Section 4.2.3, the weakness of the conventional KL-based VRA is that overfitting often occurs in the estimation if noisy labels exist. Moreover, the querying measure based on overfitted parameters might give inaccurate queries. To alleviate this weakness of the KL-based VRA, we incorporate robust divergences, β-divergence [6] and γ-divergence [16], into VRA. These robust divergences are called the density power divergences, and they enable the robust estimation with the noisy labels. The estimators based on the density power divergences are known as M-estimators. Therefore, we incorporate the robust divergences into the extended querying measure obtained from the discussion on the M-estimator in the previous subsection.

β-divergence

The density power divergence is a class of statistical measures to evaluate the difference between two probabilistic distributions. This divergence has been developed to provide a robust estimation against unanticipated noisy labels.

Now, let us denote the contaminated labeling distribution of the noisy oracle by:

$$g(y) = (1-\eta)f(y) + \eta\delta(y),$$

where η is a mixture ratio, $f(\cdot)$ is a true distribution and $\delta(\cdot)$ is a distribution of noisy labels. Then, the density power divergence can estimate the true distribution $f(y)$ from the samples given by the contaminated distribution $g(y)$ if the contaminated distribution satisfies the following assumptions:

Assumption 2 *η is sufficiently small, and*

Assumption 3 *$f(y^*)$ is sufficiently small for any noisy label y^*.*

The illustration of the contaminated distribution is shown in Fig. 4.1, where the left side of the mountain stands for the true distribution and the other mountain is the distribution of the noisy labels.

Under the above assumptions, one of the density power divergences called β-divergence has been proposed in [6]. The divergence between $q(y|\mathrm{x})$ and $p_{\boldsymbol{\theta}}(y|\mathrm{x})$ is defined by

$$\begin{aligned} D_\beta(q\|p_{\boldsymbol{\theta}}) = \frac{1}{(1+\beta)}\Bigg\{\frac{1}{\beta}\iint q(y|\mathrm{x})^{1+\beta}\mathrm{d}yq(\mathrm{x})\mathrm{dx} \\ -\iint q(y|\mathrm{x})p_{\boldsymbol{\theta}}(y|\mathrm{x})^{\beta}\mathrm{d}yq(\mathrm{x})\mathrm{dx} + \iint p_{\boldsymbol{\theta}}(y|\mathrm{x})^{1+\beta}\mathrm{d}yq(\mathrm{x})\mathrm{dx}\Bigg\}, \end{aligned}$$

where β is a positive constant. Note that the β-divergence converges to the KL-divergence if $\beta \to 0$. Therefore, this can be regarded as a generalization of the KL-divergence of Eq. (4.1).

Estimation of the model parameter based on the β-divergence can be achieved through the minimization of this divergence. The minimizer of the β-divergence is obtained by the derivative of β-divergence:

$$\partial_{\boldsymbol{\theta}} D_\beta(q\|p_{\boldsymbol{\theta}}) = \mathbb{E}_{\boldsymbol{\theta}^*}\left[\boldsymbol{\psi}_\beta(y|\mathrm{x};\boldsymbol{\theta})\right],$$

where the vector function $\boldsymbol{\psi}_\beta(y|\mathrm{x};\boldsymbol{\theta})$ is:

$$\begin{aligned} \boldsymbol{\psi}_\beta(y|\mathrm{x};\boldsymbol{\theta}) &= p_{\boldsymbol{\theta}}(y|\mathrm{x})^{\beta}\partial_{\boldsymbol{\theta}}\log p_{\boldsymbol{\theta}}(y|\mathrm{x}) \\ &\quad -\iint p_{\boldsymbol{\theta}}(y|\mathrm{x})^{\beta+1}\partial_{\boldsymbol{\theta}}\log p_{\boldsymbol{\theta}}(y|\mathrm{x})\mathrm{d}yq(\mathrm{x})\mathrm{dx}. \end{aligned} \tag{4.12}$$

Now, suppose that a set of labeled samples $\{(\mathrm{x}_1, y_1), \cdots, (\mathrm{x}_n, y_n)\}$ are obtained from the true distribution $q(\mathrm{x}, y)$. Then, the β-divergence-based estimator is given as a solution of the following estimating equation:

$$\frac{1}{n}\sum_{i=1}^{n} p_{\hat{\boldsymbol{\theta}}_n}(y_i|\mathrm{x}_i)^{\beta}\partial_{\boldsymbol{\theta}} \log p_{\hat{\boldsymbol{\theta}}_n}(y_i|\mathrm{x}_i) - \frac{1}{n}\sum_{i=1}^{n}\int p_{\hat{\boldsymbol{\theta}}_n}(y|\mathrm{x}_i)^{\beta+1}\partial_{\boldsymbol{\theta}} \log p_{\hat{\boldsymbol{\theta}}_n}(y|\mathrm{x}_i)\mathrm{d}y = \mathbf{0}, \tag{4.13}$$

which is derived by replacing the expectation over $q(\mathrm{x}, y)$ with the samples. Here, we note that the function $\boldsymbol{\psi}_\beta(y|\mathrm{x}; \boldsymbol{\theta})$ satisfies the conditions (4.4)-(4.6) and therefore β-divergence-based estimator obtained from Eq. (4.13) is an M-estimator [6].

The common property of all density power divergence is to take the self-weighted log-likelihood estimating equation, such as Eq. (4.13). These weighted estimating equations allow us to estimate the parameters robustly against noisy labels. Fig. 4.1 demonstrates how the density-power-divergence-based estimator reduces the influence of noisy labels. Since noisy labels have lower probabilities with the model $p_{\boldsymbol{\theta}}$ because of Assumption 2 and 3, the weights on noisy labels automatically become small in the estimating equation. This characterizes the density power divergence as a robust estimator.

From Proposition 2, an asymptotic variance of the β-divergence-based estimator is as follows:

$$\mathbb{E}_{\boldsymbol{\theta}^*}\left[(\hat{\boldsymbol{\theta}}_n - \boldsymbol{\theta}^*)(\hat{\boldsymbol{\theta}}_n - \boldsymbol{\theta}^*)^{\top}\right] = \frac{1}{n}\mathbf{A}_{\beta,\boldsymbol{\theta}^*}^{-1}\mathbf{M}_{\beta,\boldsymbol{\theta}^*}\left(\mathbf{A}_{\beta,\boldsymbol{\theta}^*}^{-1}\right)^{\top}, \tag{4.14}$$

where

$$\begin{aligned}\mathbf{A}_{\beta,\boldsymbol{\theta}^*} &= \mathbb{E}_{\boldsymbol{\theta}^*}\left[\partial_{\boldsymbol{\theta}}\boldsymbol{\psi}_\beta(y|\mathrm{x}; \boldsymbol{\theta}^*)\right], \\ \mathbf{M}_{\beta,\boldsymbol{\theta}^*} &= \mathbb{E}_{\boldsymbol{\theta}^*}\left[\boldsymbol{\psi}_\beta(y|\mathrm{x}; \boldsymbol{\theta}^*)\boldsymbol{\psi}_\beta(y|\mathrm{x}; \boldsymbol{\theta}^*)^{\top}\right].\end{aligned}$$

Then, we call our active learning method based on β-divergence-based estimator and its asymptotic variance as β-AL.

γ-divergence

The γ-divergence, a variant of the β-divergence, is defined as follows [16]:

$$\begin{aligned} D_\gamma(q\|p_{\boldsymbol{\theta}}) = \frac{1}{\gamma+1}\Bigg\{ &\frac{1}{\gamma}\log\iint q(y|\mathbf{x})^{1+\gamma}\mathrm{d}yq(\mathbf{x})\mathrm{d}\mathbf{x} \\ &-\log\iint q(y|\mathbf{x})p_{\boldsymbol{\theta}}(y|\mathbf{x})^{\gamma}\mathrm{d}yq(\mathbf{x})\mathrm{d}\mathbf{x} + \log\iint p_{\boldsymbol{\theta}}(y|\mathbf{x})^{1+\gamma}\mathrm{d}yq(\mathbf{x})\mathrm{d}\mathbf{x}\Bigg\}, \end{aligned}$$

where γ is a positive constant. The γ-divergence also converges to the KL-divergence if $\gamma \to 0$. The parameter estimation based on the γ-divergence, as well as β-divergence, is obtained by the derivative of the divergence with respect to the parameter:

$$\partial_{\boldsymbol{\theta}} D_\gamma(q\|p_{\boldsymbol{\theta}}) = \mathbb{E}_{\boldsymbol{\theta}^*}\left[\boldsymbol{\psi}_\gamma(y|\mathbf{x};\boldsymbol{\theta})\right],$$

where the vector function $\boldsymbol{\psi}_\gamma(y|\mathbf{x};\boldsymbol{\theta})$ is:

$$\begin{aligned} \boldsymbol{\psi}_\gamma(y|\mathbf{x};\boldsymbol{\theta}) = &\frac{p_{\boldsymbol{\theta}}(y|\mathbf{x})^{\gamma}}{\iint p_{\boldsymbol{\theta}^*}(y|\mathbf{x})p_{\boldsymbol{\theta}}(y|\mathbf{x})^{\gamma}\mathrm{d}yq(\mathbf{x})\mathrm{d}\mathbf{x}}\partial_{\boldsymbol{\theta}}\log p_{\boldsymbol{\theta}}(y|\mathbf{x}), \\ &-\iint \frac{p_{\boldsymbol{\theta}}(y|\mathbf{x})^{\gamma+1}}{\iint p_{\boldsymbol{\theta}}(y|\mathbf{x})p_{\boldsymbol{\theta}}(y|\mathbf{x})^{\gamma+1}\mathrm{d}yq(\mathbf{x})\mathrm{d}\mathbf{x}}\partial_{\boldsymbol{\theta}}\log p_{\boldsymbol{\theta}}(y|\mathbf{x})\mathrm{d}yq(\mathbf{x})\mathrm{d}\mathbf{x}. \end{aligned} \tag{4.15}$$

Suppose that we have a set of *i.i.d* n labeled samples $\{(\mathbf{x}_1, y_1), \cdots, (\mathbf{x}_n, \cdots, y_n)\}$ obtained from the true distribution $q(\mathbf{x}, y)$. Then, the estimating equation of γ-divergence-based estimator is given by replacing the expectation over $q(\mathbf{x}, y)$ with the sample mean:

$$\begin{aligned} &\sum_{i=1}^{n}\left(\frac{p_{\hat{\boldsymbol{\theta}}_n}(y_i|\mathbf{x}_i)^{\gamma}}{\sum_{i=1}^{n} p_{\hat{\boldsymbol{\theta}}_n}(y_i|\mathbf{x}_i)^{\gamma}}\right)\partial_{\boldsymbol{\theta}}\log p_{\hat{\boldsymbol{\theta}}_n}(y_i|\mathbf{x}_i) \\ &\quad -\frac{1}{n}\sum_{i=1}^{n}\int\left(\frac{p_{\hat{\boldsymbol{\theta}}_n}(y|\mathbf{x}_i)^{\gamma+1}}{\frac{1}{n}\sum_{j=1}^{n}\int p_{\hat{\boldsymbol{\theta}}_n}(y|\mathbf{x}_j)^{\gamma+1}\mathrm{d}y}\right)\partial_{\boldsymbol{\theta}}\log p_{\hat{\boldsymbol{\theta}}_n}(y|\mathbf{x}_i)\mathrm{d}y = \mathbf{0}. \end{aligned} \tag{4.16}$$

This can be regarded as the weight-normalized version of Eq. (4.13) and can robustly estimate the model under Assumption 2 and 3. Similar to the β-divergence-based estimator, γ-divergence-based estimator obtained from the above equation is also an M-estimator since the function $\boldsymbol{\psi}_\gamma(y|\mathbf{x};\boldsymbol{\theta})$ satisfies the conditions (4.4)-(4.6) [16].

From Proposition 2, an asymptotic variance of the γ-divergence-based estimator is also as follows:

$$\mathbb{E}_{\boldsymbol{\theta}^*}\left[(\hat{\boldsymbol{\theta}}_n - \boldsymbol{\theta}^*)(\hat{\boldsymbol{\theta}}_n - \boldsymbol{\theta}^*)^{\top}\right] = \frac{1}{n}\mathbf{A}_{\gamma,\boldsymbol{\theta}^*}^{-1}\mathbf{M}_{\gamma,\boldsymbol{\theta}^*}\left(\mathbf{A}_{\gamma,\boldsymbol{\theta}^*}^{-1}\right)^{\top}, \tag{4.17}$$

where

$$\mathbf{A}_{\gamma,\boldsymbol{\theta}^*} = \mathbb{E}_{\boldsymbol{\theta}^*}\left[\partial_{\boldsymbol{\theta}}\boldsymbol{\psi}_\gamma(y|\mathbf{x};\boldsymbol{\theta}^*)\right],$$
$$\mathbf{M}_{\gamma,\boldsymbol{\theta}^*} = \mathbb{E}_{\boldsymbol{\theta}^*}\left[\boldsymbol{\psi}_\gamma(y|\mathbf{x};\boldsymbol{\theta}^*)\boldsymbol{\psi}_\gamma(y|\mathbf{x};\boldsymbol{\theta}^*)^\top\right].$$

Although characteristics of the γ-divergence are similar to those of β-divergence, behaviors of these divergences may be different from each other in active learning context. Therefore, we propose both β-divergence-based and γ-divergence-based active learning methods and empirically assess their properties in comparison. Hereafter, we call the active learning method based on γ-divergence-based estimator and its asymptotic variance as γ-AL.

4.4 Empirical Measures for Querying

Based on the asymptotic variance of Eqs. (4.14) and (4.17) in the previous section, we explain empirical querying measures in our active learning methods. For simplicity, we collectively denote the estimating functions Eqs. (4.12) and (4.15) by $\boldsymbol{\psi}(y|\mathbf{x};\boldsymbol{\theta})$.

4.4.1 Approximation of Querying Measure

Our strategy for selecting queries in active learning is to minimize the variance Eq. (4.11). However, since the true distribution $q(\mathbf{x}, y) = q(\mathbf{x})p_{\boldsymbol{\theta}^*}(\mathbf{x}, y)$ is not known, $\mathbf{A}_{\boldsymbol{\theta}^*}$ and $\mathbf{M}_{\boldsymbol{\theta}^*}$ of Eqs. (4.9) and (4.10) cannot be calculated directly. Therefore, similar to the existing work [62], we approximate $\mathbf{A}_{\boldsymbol{\theta}^*}$ and $\mathbf{M}_{\boldsymbol{\theta}^*}$ by using the estimated parameter $\hat{\boldsymbol{\theta}}_n$ and replacing the expectation by queries:

$$\widehat{\mathbf{A}}_{\hat{\boldsymbol{\theta}}_n}(\mathcal{S}) = \sum_{\mathbf{x}_i\in\mathcal{S}}\int p_{\hat{\boldsymbol{\theta}}_n}(y|\mathbf{x}_i)\partial_{\boldsymbol{\theta}}\boldsymbol{\psi}(y|\mathbf{x}_i;\hat{\boldsymbol{\theta}}_n)\mathrm{d}y, \tag{4.18}$$

$$\widehat{\mathbf{M}}_{\hat{\boldsymbol{\theta}}_n}(\mathcal{S}) = \sum_{\mathbf{x}_i\in\mathcal{S}}\int p_{\hat{\boldsymbol{\theta}}_n}(y|\mathbf{x}_i)\boldsymbol{\psi}(y|\mathbf{x}_i;\hat{\boldsymbol{\theta}}_n)\boldsymbol{\psi}(y|\mathbf{x}_i;\hat{\boldsymbol{\theta}}_n)^\top\mathrm{d}y. \tag{4.19}$$

where $\mathcal{S}$ is a set of the queries selected at Step 2 in Algorithm 5.

If a model consists of an unique parameter, the above equations are scalars. In this case, the variance of the parameter given as the product of Eqs. (4.18) and (4.19) is also scalar. Therefore, by selecting a set of queries $\mathcal{S}$, we minimize the scalar value of the variance. However, in our case, the regression model has not less than two parameters (the coefficient $\mathbf{w}$ and the variance σ^2) and therefore our querying measure needs to be optimized over a matrix. This optimization for the matrix is not trivial. In this study, we take the trace norm of the matrix and derive the querying measure as follows:

$$\mathcal{S}^* = \underset{\mathcal{S}\subseteq\mathcal{U}\wedge|\mathcal{S}|=Q}{\operatorname{argmin}} Z(\mathcal{S};\hat{\boldsymbol{\theta}}_n),$$

where $|\cdot|$ is a cardinality of a set, Q is the number of queries and

$$Z(\mathcal{S};\hat{\boldsymbol{\theta}}_n) = \frac{1}{Q}\mathrm{tr}\left\{\widehat{\mathbf{A}}_{\hat{\boldsymbol{\theta}}_n}(\mathcal{S})^{-1}\widehat{\mathbf{M}}_{\hat{\boldsymbol{\theta}}_n}(\mathcal{S})(\widehat{\mathbf{A}}_{\hat{\boldsymbol{\theta}}_n}(\mathcal{S})^{-1})^\top\right\}. \tag{4.20}$$

The trace norm of the matrix stands for the sum of the diagonal element of the matrix, *i.e.*, the variance of each parameter. Therefore, this empirical measure minimizes the sum of the differences between the estimated parameter and the true parameter. The procedure of taking a trace norm is known as *A-optimality* and is popular in active learning [21, 41].

Unfortunately, the empirical querying measure based on the γ-divergence still cannot be calculated directly because of the integration in it. Therefore, we employ Monte Carlo integration method to compute the querying measure.

4.4.2 Optimization of Querying Measure

In the case with the number of the queries $|\mathcal{S}| = 1$, this active learning algorithm is a simple one-by-one active learning method. For the online method, it requires many iterations of querying and estimating step (Step 2, Algorithm 5) to obtain an accurate model. In applications, such the iteration is time-consuming and bothersome since the oracle has to give a label in each iteration. Hence, in practice, we prefer to use batch-mode algorithms, *i.e.*, $|\mathcal{S}| \geq 2$ [21]. However, optimizing $\mathcal{S}$ is a combinatorial problem which is difficult to solve. In this case, the greedy algorithms are usually used to solve the combinatorial problems. Therefore, we naively employ the greedy algorithms to optimize a set of queries as shown in Algorithm 6.

Algorithm 6 Greedy algorithm for selecting queries.

```
1. Given a set of unlabeled samples $\mathcal{L}$, the estimated model parameter $\hat{\boldsymbol{\theta}}_n$ and the number of queries $Q$, and initialize a set of queries $\mathcal{S} := \emptyset$.
2. Repeat until $Q$ queries are added to $\mathcal{S}$.
    (a) Select a query that minimize the querying measure of Eq. (4.20):
        $\mathbf{x}^{(m)} = \operatorname*{argmin}_{\mathbf{x}_i \in \mathcal{U} \backslash \mathcal{S}} Z(\mathcal{S} \cup \mathbf{x}_i; \hat{\boldsymbol{\theta}}_n),$
        where $\mathbf{x}^{(m)}$ is an unlabeled sample selected as the query.
    (b) Let $\mathcal{S} := \mathcal{S} \cup \mathbf{x}^{(m)}$.
3. Obtain the set $\mathcal{S}$ containing $Q$ queries.
```

4.5 Experiments

In this section, we show some experimental results to illustrate the robustness of our active learning method by using artificial and real-world datasets. In these experiments, we compared the following six methods.

KL-RAND A standard algorithm which learns by the KL-divergence and selects queries randomly.

β-RAND A algorithm which learns by the β-divergence and selects queries randomly.

γ-RAND A algorithm which learns by the γ-divergence and selects queries randomly.

KL-AL The conventional active learning algorithm based on the KL-divergence ([62] applied in the linear regression model).

β-AL Our proposed active learning method based on the β-divergence.

γ-AL The other proposed active learning method based on the γ-divergence.

As for KL-AL and KL-RAND, the parameters were estimated by solving Eq. (4.7) analytically. On the other hand, for the other methods, we solved nonlinear equations Eqs. (4.13) and (4.16) by quasi-Newton's method [26] which is one of the gradient-based methods. In contrast to the naive Newton's method, the quasi-Newton's method need not to compute the inversion of the gradient matrix (Hessian matrix). Without the time-consuming computation of the inversion matrix, the method is expected to estimate the model parameter rapidly.

Moreover, in these experiments, the sample size for the Monte Carlo integration in γ-AL was set to be 250 to maintain the tractable computational time for comparison with the other methods. Furthermore, the parameter values β and γ were set to be 0.1 based on the preceding work on the parameter analysis [6, 16].

4.5.1 Evaluation of Robustness

In the first experiment, we investigated the robustness of the proposed methods using artificial datasets. The procedure for generating the datasets is as follows: First, we randomly generated samples $\mathbf{x}_i$ from a uniform distribution in the range of [-1, 1], where the dimensionality and the number of samples are respectively 5 and 300. Next, we randomly generated five-dimensional coefficient vector $\mathbf{w}$ and the constant term w_0 from a uniform distribution in the range of [-2.5, 2.5]. Moreover, noises ϵ_i in the linear regression model were randomly generated from a Gaussian distribution with zero mean and unit variance. Finally, we determined labels y_i as $y_i = \mathbf{w}^\mathrm{T}\mathbf{x}_i + w_0 + \epsilon_i$.

Each of the generated datasets is randomly partitioned into the training set $\mathcal{T}_{train}$ with 80% samples and the test set $\mathcal{T}_{test}$ with 20% samples. 10 samples were randomly selected from the training set $\mathcal{T}_{train}$ as initial labeled samples $\mathcal{L}$. Then, noises ± 5 are added to $\eta\%(\eta = 0, 0.2, \cdots, 5)$ of randomly selected labels in the remaining samples $\mathcal{U}$ as noisy labels given by the noisy oracle. In this experiment, the number of iterations for querying R in Algorithm 5 was set to be 2, and the number of queries Q in Algorithm 6 is set to be 5. We evaluate a mean-squared error (MSE) between the true label and the

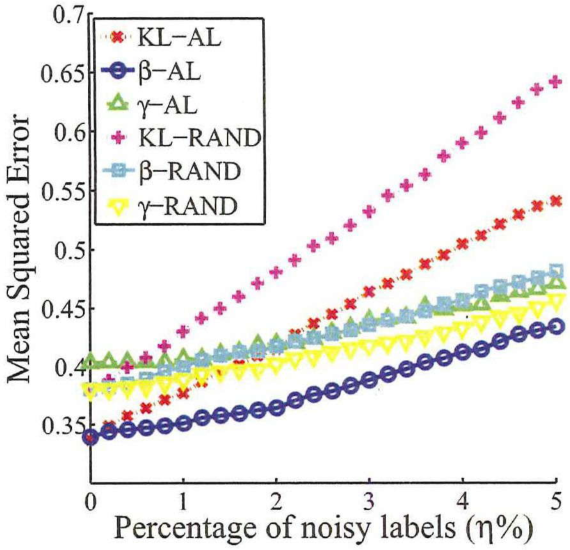


Figure 4.2: Difference among the five methods under various amounts of noisy labels.

Table 4.1: Specifications of Datasets

Dataset	# of dimension p	# of samples
concrete	8	1030
forestfires	8	517
imports	14	160
machine	7	209
elevator	6	9517
stockvalues	159	1813

estimated label in the test set $\mathcal{T}_{test}$ as follows:

$$\text{MSE} = \frac{1}{|\mathcal{T}_{test}|} \sum_{(\mathbf{x}_i, y_i) \in \mathcal{T}_{test}} (y_i - \hat{\mathbf{w}}^\top \mathbf{x}_i - \hat{w}_0)^2,$$

where $\hat{\mathbf{w}}$ is the p-dimensional estimated coefficient vector and $\hat{w}_0$ is the estimated constant term.

Fig. 4.2 shows MSE between the true labels and the estimated labels by the compared methods. The values in the graph are averaged over 2000 random trials for numerical stabilization. As can be seen in Fig. 4.2, with the increasing of noisy labels, the average errors by the KL-divergence-based methods grow more rapidly than the β/γ-divergence-based methods. Also, in most cases, the active learning methods seem to perform better than the random-query methods. However, the performance of γ-AL was worse than β-AL and the other random-query methods. This would be because the querying measure of γ-AL selects less informative samples by an approximation of the Monte Carlo integration. Although one could improve the approximation by increasing the number of samples, it usually leads severe increase of computational costs. Thus, these results seem to show that β-AL practically achieves the robust estimation of the regression model.

4.5.2 Evaluation with Real-world data

Next, we conducted experiments with six real-world datasets provided from [3, 58, 36] to examine the robustness of our proposed methods. The summaries of the datasets are given in Table 4.1. First, each of the datasets is partitioned into an initial set $\mathcal{L}$, an unlabeled set $\mathcal{U}$ and a test set $\mathcal{T}_{test}$ in the same manner with the previous experiment. Then, noisy labels were generated by adding ± 5 to $\eta\%(\eta = 0, 5)$ samples randomly selected from $\mathcal{U}$. For this experiment, if the cardinality of $\mathcal{U}$ is more than 300, 300 samples were subsampled from $\mathcal{U}$ as candidates for the unlabeled samples in selecting queries[1]. In this experiment, we set the number of learning iteration R and queries Q to 5, respectively. Similar to the previous experiment, we evaluated the average MSE of 1000 random trials by using the test set $\mathcal{T}_{test}$.

The graphs in Fig. 4.3 show the errors at each learning step of the methods. Note that the result of γ-AL with the "stockvalues" dataset could not be obtained because of its high computational cost of the numerical integration. As can be seen in Fig. 4.3, the error of β-AL is comparable with KL-AL without the noisy labels. On the other hand, in cases where the noisy labels exist, the errors of KL-AL become larger than β-AL. Especially in the case of the high-dimensional "stockvalues" dataset with the noisy

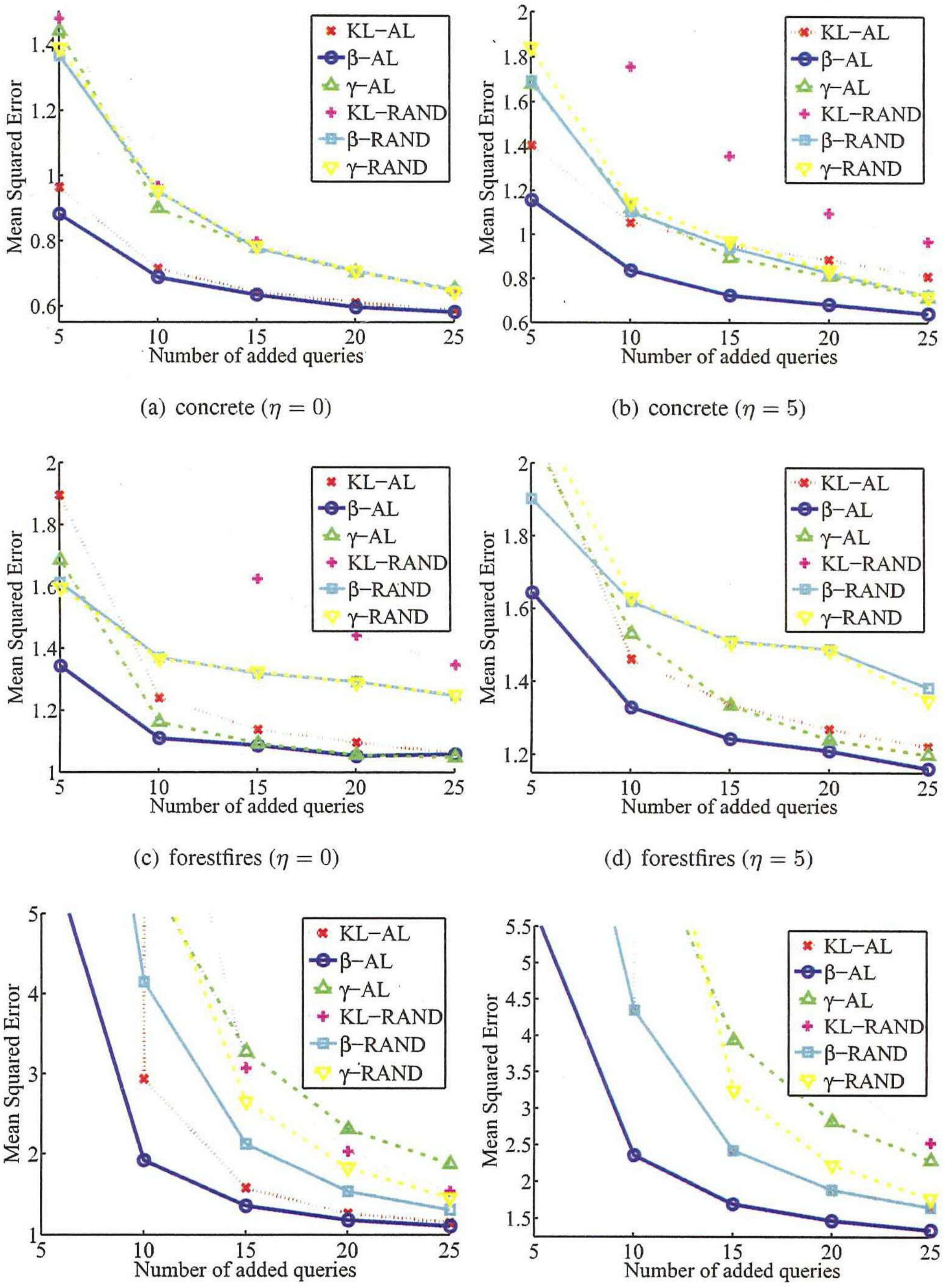
[1] Although it would have been better to utilize all samples in practice, we subsampled to compare all the methods in this experiment.

labels, KL-AL completely fails to estimate the regression model. Similar to the result of the previous experiment, γ-AL seems to work worse than the other methods in most cases because of the numerical integration. Thus, our proposed method β-AL seems to work more robustly than the other methods in this experiment.

4.6 Conclusion

We proposed the robust active learning methods for the linear regression model. Our querying measures were obtained by extending the conventional measure through the asymptotic analysis of the M-estimator and incorporating the β- and γ-divergence-based estimators into the extended measure. The proposed methods can achieve robust results under the situation with the noisy oracle because of the characteristics of the robust divergences. We investigated the performance of our methods by the experiments with the artificial datasets and the real-world datasets. From these experiments, we confirmed that it could estimate the regression model accurately and robustly from the small labeled samples even under the situation with the noisy oracle.

In this chapter, we applied our robust active learning measure to a linear regression model and achieved the robust estimation of the relation between the variables. However, our framework is not restricted to the linear regression model. We can also apply it to the other models such as a logistic model for discovering a relation between a binary variable and the other continuous variables. Therefore, one of our future works is to apply our robust active learning framework to the other models and investigate their behaviors. In addition, the procedure for optimizing the queries can be improved by using discrete optimization techniques. Recently, a study on the conventional KL-based active learning method using VRA [21] showed that its querying measure have a property called submodularity [14]. This property provides an efficient algorithm to solve the discrete optimization problem. Because of this property, we expect that the optimization of the queries can be performed more efficiently [29].



(a) concrete ($\eta = 0$)

(b) concrete ($\eta = 5$)

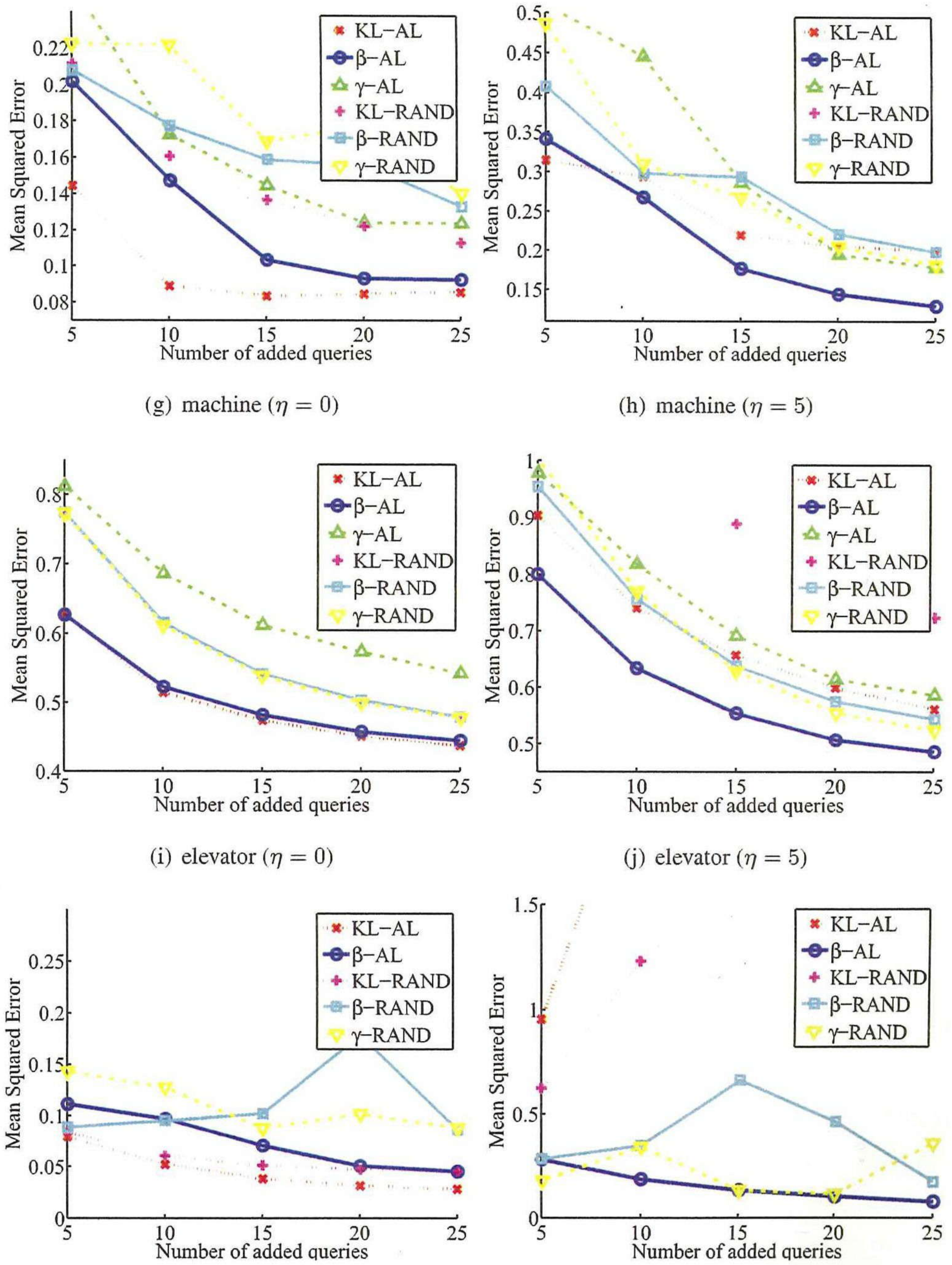
(c) forestfires ($\eta = 0$)

(d) forestfires ($\eta = 5$)

(e) imports ($\eta = 0$)

(f) imports ($\eta = 5$)

(To be continued)



(g) machine ($\eta = 0$)

(h) machine ($\eta = 5$)

(i) elevator ($\eta = 0$)

(j) elevator ($\eta = 5$)

(k) stockvalues ($\eta = 0$)

(l) stockvalues ($\eta = 5$)

Figure 4.3: Comparisons of the means-squared error among six methods at each learning step. The left figures are for the MSE without noisy labels and the right are for the MSE with 5% noisy labels.

Chapter 5

Conclusion

In this dissertation, we presented three methods to estimate the variable relations. They are categorized into main two branches. One is the technique to obtain some knowledge on the directed network representing the ordering of the effects among the variables. A basis of this technique is the linear non-Gaussian acyclic model called LiNGAM model. The other is the technique to estimate the relation between the label variable and the other explanatory variables, which is known as the linear regression. The past methods for estimating these variable relations cannot achieve the good performance under real-world datasets consisting of small samples and/or noisy samples. In this dissertation, we tackled this problem.

First, we attacked the problem that the past LiNGAM-model-based methods cannot obtain sufficient knowledge on the network in the analysis with high-dimensional and small sample data. To solve this problem, we proposed a variant of the linear non-Gaussian acyclic model based on some realistic assumptions. Subsequently, we proposed a method called EggFinder to estimate the exogenous variable in the network. With the numerical experiments, we confirmed that our proposed algorithm can estimate the exogenous variables from the high-dimensional and small sample data. Further, we investigated the applicability of our method by using the gene microarray data with small samples. The genes found by EggFinder is likely to be exogenous in the gene network according to the domain knowledge in bioinformatics.

Second, we proposed the LiNGAM-model-based method which can estimate the entire network more accurately and robustly from noisy and small sample data than the past non-Gaussianity-based methods. To estimate a correct network, the evaluation of the independence between the variables and the solution search algorithm are important. Thus, we presented the two principles to modify the past methods on these points. One is to incorporate kernel based independence measure for enhancing the robustness and the accuracy of the network estimation. The other is to employ the beam search algorithm to avoid the local optima. Based on these modifications, the four variants of the past methods were proposed. Then, we investigated their robustness and accuracy through the numerical experiments. Furthermore, we discussed the trade off between the accuracy and the computational cost. From the results, we concluded that the method, Beam-Kernel-DirectLiNGAM is the best in terms of accuracy, robustness and tractability for the network estimation even under the noisy and small sample data.

Third, we focused on the linear regression model and its active learning by using small labeled samples and large unlabeled samples. Conventional active learning methods cannot estimate the model under the situation with the noisy oracle giving noisy labeled samples. Therefore, we proposed a more robust active learning method for estimating the regression model. Firstly, we extended the conventional querying measure based on M-estimator. Subsequently, we incorporated the robust divergences into the extended querying measure. Then, our proposed methods and the conventional method are compared by the numerical experiments with the artificial datasets and the real-world datasets. From the results of these experiments, we confirmed that our proposed method can estimate the linear regression model robustly from the small noisy labeled samples.

This dissertation aims to close some gaps between the real-world problems and the techniques for estimating the variable relations. This objective was achieved by our three proposal. However, more extensions may be possible. One is the extension to incorporate non-linearity of the relations in both the LiNGAM model and the regression model. It is worth for the real-world problems where many relations are non-linear. Second is introducing the more sophisticated independence measure and rapid search algorithm into the network estimation method. Third is to consider efficient optimization algorithm so that the robust active learning method for the regression model can deal with much higher-dimensional datasets. These extensions will enhance the applicability of our proposed methods to wider practical problems.

References

[1] S. Akaho. A kernel method for canonical correlation analysis. In *In Proceedings of the International Meeting of the Psychometric Society (IMPS2001)*. Springer-Verlag, 2001.

[2] D. Angluin. Queries and concept learning. *Machine Learning*, 2(4):319–342, 1988.

[3] A. Asuncion and D.J. Newman. UCI machine learning repository, 2007.

[4] L. Atlas, D. Cohn, R. Ladner, M. A. El-Sharkawi, and R. J. Marks, II. Training connectionist networks with queries and selective sampling. In *Advances in Neural Information Processing Systems 2*, pages 566–573, 1990.

[5] F. R. Bach and M. I. Jordan. Kernel independent component analysis. *Journal of Machine Learning Research*, 3:1–48, 2002.

[6] A. Basu, I.R. Harris, N.L. Hjort, and MC Jones. Robust and efficient estimation by minimising a density power divergence. *Biometrika*, 85(3):549–559, 1998.

[7] Y. Benjamini and Y. Hochberg. Controlling the false discovery rate: a practical and powerful approach to multiple testing. *Journal of Royal Statistical Society: Series B*, 57:289–300, 1995.

[8] P. Billingsley. *Probability and Measure*. Wiley Series in Probability and Statistics, 1986.

[9] A.C. Burns and R.F. Bush. Marketing research. *Globalization*, 1:7, 2000.

[10] C. Campbell, N. Cristianini, and A. Smola. Query learning with large margin classifiers. In *Proceedings of the 17th International Conference on Machine Learning*, pages 111–118, 2000.

[11] P. Comon. Independent component analysis, a new concept? *Signal Processing*, 36:62–83, 1994.

[12] N. Delfosse and P. Loubaton. Adaptive blind separation of independent sources: a deflation approach. *Signal Processing*, 45:59–83, 1995.

[13] D. di Bernardo, M.J. Thompson, T.S. Gardner, S.E. Chobot, E.L. Eastwood, A.P. Wojtovich, S.J. Elliot, S.E. Schaus, and J.J Collins. Chemogenomic profiling on a genome-wide scale using reverse-engineered gene networks. *Nature Biotech.*, 23:377–383, 2005.

[14] J. Edmonds. Submodular functions, matroids, and certain polyhedra. *Combinatorial Optimization?Eureka, You Shrink!*, pages 11–26, 2003.

[15] K. Elenius, S. Paul, G. Allison, J. Sun, and M. Klagsbrun. Activation of HER4 by heparin-binding EGF-like growth factor stimulates chemotaxis but not proliferation. *The EMBO journal*, 16(6):1268–1278, 1997.

[16] H. Fujisawa and S. Eguchi. Robust parameter estimation with a small bias against heavy contamination. *Journal of Multivariate Analysis*, 99(9):2053–2081, 2008.

[17] A.S. Goldberger. Structural equation methods in the social sciences. *Econometrica: Journal of the Econometric Society*, pages 979–1001, 1972.

[18] D. Graus-Porta, R.R. Beerli, J.M. Daly, and N.E. Hynes. ErbB-2, the preferred heterodimerization partner of all ErbB receptors, is a mediator of lateral signaling. *The EMBO journal*, 16(7):1647–1655, 1997.

[19] D. Hakkani-Tur, G. Riccardi, and A. Gorin. Active learning for automatic speech recognition. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, volume 4, pages 3904–39078, 2002.

[20] J.P. Hoffmann. *Generalized linear models.* Allyn & Bacon Boston, MA, 2003.

[21] S.C.H. Hoi, R. Jin, J. Zhu, and M.R. Lyu. Batch mode active learning and its application to medical image classification. In *Proceedings of the 23rd International Conference on Machine Learning*, pages 417–424, 2006.

[22] P. J. Huber and E. M. Ronchetti. *Robust Statistics*. John Wiley and Sons, 2009.

[23] A. Hyvärinen. Fast and robust fixed-point algorithms for independent component analysis. *IEEE Transactions on Neural Networks*, 10:626–634, 1999.

[24] A. Hyvärinen, J. Karhunen, and E. Oja. Independent component analysis. *John Willey and Sons*, 2001.

[25] A. Hyvärinen and E. Oja. A fast fixed-point algorithm for independent component analysis. *Neural Computation*, 9:1483–1492, 1997.

[26] C.T. Kelley. *Solving nonlinear equations with Newton's method*, volume 1. Society for Industrial Mathematics, 1987.

[27] J. Kim, W.J. Jahng, D. Di Vizio, J.S. Lee, R. Jhaveri, M.A. Rubin, A. Shisheva, and M.R. Freeman. The phosphoinositide kinase PIKfyve mediates epidermal growth factor receptor trafficking to the nucleus. *Cancer Research*, 67(19):9229–9237, 2007.

[28] K.W. Kohn. Molecular interaction map of the mammalian cell cycle control and DNA repair systems. *Molecular Biology of the Cell*, 10(8):2703–2734, 1999.

[29] A. Krause. Sfo: A toolbox for submodular function optimization. *Journal of Machine Learning Research*, 11:1141–1144, 2010.

[30] J. Lazarou, B.H. Pomeranz, and P.N. Corey. Incidence of adverse drug reactions in hospitalized patients. *JAMA: the journal of the American Medical Association*, 279(15):1200–1205, 1998.

[31] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[32] E.L. Lehmann and J.P. Romano. *Testing Statistical Hypotheses.* Springer, 2005.

[33] D. D. Lewis and W. A. Gale. A sequential algorithm for training text classifiers. In *Proceedings of the 17th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 3–12, 1994.

[34] A. Londei, A. D'Ausilio, D. Basso, and M. O. Belardinelli. A new method for detecting causality in fMRI data of cognitive processing. *Cognitive Processing*, 7(1):42–52, 2006.

[35] A. McCallum and K. Nigam. Employing EM in pool-based active learning for text classification. In *Proceedings of the 15th International Conference on Machine Learning*, pages 350–358, 1998.

[36] mldata.org, 2012. `http://mldata.org`.

[37] T. Nagashima, H. Shimodaira, K. Ide, T. Nakakuki, Y. Tani, K. Takahashi, N. Yumoto, and M. Hatakeyama. Quantitative transcriptional control of ErbB receptor signaling undergoes graded to biphasic response for cell differentiation. *Journal of Biological Chemistry*, 282(6):4045, 2007.

[38] H. Nishi, K.H. Nishi, and A.C. Johnson. Early growth response-1 gene mediates up-regulation of epidermal growth factor receptor expression during hypoxia. *Cancer Research*, 62(3):827–834, 2002.

[39] J. Pearl. *Causality: Models, Reasoning, and Inference.* Cambridge University Press, 2000.

[40] C. R. Rao. *Linear Statistical Inference and its Applications, 2nd ed.* Wiley-Interscience, New York, 2002.

[41] B. Settles. Active learning literature survey. Technical Report Computer Science Technical Report 1648, University of Wisconsin-Madison, 2010.

[42] B. Settles and M. Craven. An analysis of active learning strategies for sequence labeling tasks. In *Proceedings of the Conference on Empirical Methods in Natural*

Language Processing, pages 1070–1079. Association for Computational Linguistics, 2008.

[43] H. Shen, S. Jegelka, and A. Gretton. Fast kernel ICA using an approximate newton method. In *Proceedings of the 11th International Conference on Artificial Intelligence and Statistics*, pages 476–483, 2007.

[44] S. Shimizu, P. O. Hoyer, A. Hyvärinen, and A. Kerminen. A linear non-gaussian acyclic model for causal discovery. *Journal of Machine Learning Research*, 7:2003–2030, 2006.

[45] S. Shimizu, A. Hyvärinen, Y. Kawahara, and T. Washio. A direct method for estimating a causal ordering in a linear non-Gaussian acyclic model. In *Proceedings of 25th Conference on Uncertainty in Artificial Intelligence (UAI2009)*, 2009.

[46] P. Singhirunnusorn, Y. Ueno, M. Matsuo, S. Suzuki, I. Saiki, and H. Sakurai. Transient suppression of ligand-mediated activation of epidermal growth factor receptor by tumor necrosis factor-α through the TAK1-p38 signaling pathway. *Journal of Biological Chemistry*, 282(17):12698, 2007.

[47] Y. Sogawa, S. Shimizu, A. Hyvärinen, T. Washio, T. Shimamura, and S. Imoto. Discovery of exogenous variables in data with more variables than observations. In *Proceedings of International Conference on Artificial Neural Networks (ICANN2010)*, pages 67–76, 2010.

[48] Y. Sogawa, S. Shimizu, Y. Kawahara, and T. Washio. An experimental comparison of linear non-gaussian causal discovery methods and their variants. In *Proceedings of International Joint Conference on Neural Networks (IJCNN2010)*, pages 1–8, 2010.

[49] Y. Sogawa, S. Shimizu, T. Shimamura, A. Hyvärinen, T. Washio, and S. Imoto. Estimating exogenous variables in data with more variables than observations. *Neural Networks*, 24(8):875–880, 2011.

[50] Y. Sogawa, S. Shimizu, T. Washio, and S. Imoto. Identification of exogenously expressed genes by applying independent component analysis. In *Proceedings of the 23th Annual Conference of the Japanese Society for Artificial Intelligence*, 2009. In Japanese.

[51] Y. Sogawa, T. Ueno, Y. Kawahara, and T. Washio. Robust active learning for linear regression via density power divergence. In *Proceedings of International Conference on Neural Information Processing (ICONIP2012)*, pages 594–602, 2012.

[52] Y. Sogawa, T. Ueno, Y. Kawahara, and T. Washio. Active learning for regression via density power divergence. *Transactions of the Japanese Society for Artificial Intelligence*, 28(1):13–21, 2013. In Japanese.

[53] P. Spirtes, C. Glymour, and R. Scheines. *Causation, Prediction, and Search.* Springer Verlag, 1993.

[54] R. Srinivasan, G.M. Mager, R.M. Ward, J. Mayer, and J. Svaren. NAB2 represses transcription by interacting with the CHD4 subunit of the nucleosome remodeling and deacetylase (NuRD) complex. *Journal of Biological Chemistry*, 281(22):15129–15137, 2006.

[55] J.P. Stevens. *Applied multivariate statistics for the social sciences.* Lawrence Erlbaum, 2001.

[56] M. Sugiyama. Active learning in approximately linear regression based on conditional expectation of generalization error. *Journal of Machine Learning Research*, 7:141–166, 2006.

[57] The tetrad project. `http://www.phil.cmu.edu/projects/tetrad/`, 2012.

[58] L. Torgo. Regression datasets. `http://www.liaad.up.pt/~ltorgo/Regression/DataSets.html`, 2012.

[59] A.W. Van der Vaart. *Asymptotic Statistics.* Cambridge University Press, 2000.

[60] R.M. Wachter, P. Katz, J. Showstack, A.B. Bindman, and L. Goldman. Reorganizing an academic medical service. *JAMA: the journal of the American Medical Association*, 279(19):1560–1565, 1998.

[61] Y. Xu and A. Fern. On learning linear ranking functions for beam search. In *Proceedings of the 24th international conference on Machine learning*, pages 1047–1054, 2007.

[62] T. Zhang and F. Oles. The value of unlabeled data for classification problems. In *Proceedings of the 17th International Conference on Machine Learning*, pages 1191–1198, 2000.

Publication

Journal Articles

1. Y. Sogawa, S. Shimizu, T. Shimamura, A. Hyvärinen, T. Washio, and S. Imoto. Estimating Exogenous Variables in Data with More Variables than Observations. *Neural Networks*, 24:875–880, 2011.

2. Y. Sogawa, T. Ueno, Y. Kawahara, and T. Washio. Active Learning for Regression via Density Power Divergence. *Transactions of the Japanese Society for Artificial Intelligence*, 28(1):13–21, 2013. (In Japanese)

International Conferences

1. Y. Sogawa, S. Shimizu, Y. Kawahara, and T. Washio. An experimental comparison of linear non-Gaussian causal discovery methods and their variants. In *Proceedings of the 2010 International Joint Conference on Neural Networks (IJCNN2010)*, pages 768–775, 2010.

2. Y. Sogawa, S. Shimizu, A. Hyvärinen, T. Washio, S. Teppei, and S. Imoto. Discovery of Exogenous Variables in Data with More Variables than Observations. In *Proceedings of the 20th International Conference on Artificial Neural Networks (ICANN2010)*, pages 67–76, 2010.

3. Y. Sogawa, T. Ueno, Y. Kawahara, and T. Washio. Robust Active Learning for Linear Regression via Density Power Divergence. In *Proceedings of the 19th International Conference on Neural Information Processing (ICONIP2012)*, pages 592–602, 2012