Kernel-based Approach for Learning Causal Graphs from Mixed Data

Teny Handhayani[1] TH1075@YORK.AC.UK

James Cussens[2] JAMES.CUSSENS@BRISTOL.AC.UK

[1] *Department of Computer Science, University of York, United Kingdom*

[2] *Department of Computer Science, University of Bristol, United Kingdom*

Abstract

A causal graph can be generated from a dataset using a particular causal algorithm, for instance, the PC algorithm or Fast Causal Inference (FCI). This paper provides two contributions in learning causal graphs: an easy way to handle mixed data so that it can be used to learn causal graphs using the PC algorithm/FCI and a method to evaluate the learned graph structure when the true graph is unknown. This research proposes using kernel functions and Kernel Alignment to handle mixed data. The two main steps of this approach are computing a kernel matrix for each variable and calculating a pseudo-correlation matrix using Kernel Alignment. The Kernel Alignment matrix is used as a substitute for the correlation matrix that is the main component used in computing a partial correlation for the conditional independence test for Gaussian data in the PC Algorithm and FCI. The advantage of this idea is that is possible to handle more data types when there is a suitable kernel function to compute a kernel matrix for an observed variable. The proposed method is successfully applied to learn a causal graph from mixed data containing categorical, binary, ordinal, and continuous variables. We also introduce the Modal Value of Edges Existence (MVEE) method, a new method to evaluate the structure of learned graphs represented by Partial Ancestral Graph (PAG) when the true graph is unknown. MVEE produces an agreement graph as a proxy to the true graph to evaluate the structure of the learned graph. MVEE is successfully used to choose the best-learned graph when the true graph is unknown.

Keywords: Causal learning; Kernel Alignment; Mixed data; graph structure evaluation.

1. Introduction

A causal graph is a graphical model used to describe the cause-effect relationship between variables. Algorithms for learning causal graphs from data include the PC algorithm (Spirtes et al., 2001), Fast Causal Inference (FCI) (Spirtes et al., 2001) and Really Fast Causal Inference (RFCI) (Colombo et al., 2012). Data that includes both discrete and continuous is called *mixed data*. The PC and FCI algorithms use conditional independence tests to generate a causal graph from a dataset (Spirtes et al., 2001; Kalisch and Buhlmann, 2007). Testing for conditional independence is more complex when data is mixed than when it is either entirely discrete or entirely continuous.

Methods have been developed to handle mixed data for learning causal graphs. Tsagris et al. (2018) proposed the likelihood-ratio test based on an appropriate regression model then derived symmetric conditional independence tests. This test only works well when certain conditions hold. Raghu et al. (2018) proposed a conditional independence test based on linear and logistic regression to handle mixed data for causal discovery. It was implemented together with the modification of FCI for the causal discovery of latent variables from mixed data. Raghu et al. (2018) assumed linear and logistic regression were accurate models of the interactions between continuous and discrete variables, but it is not clear how well these assumptions will hold in certain continuous

nonlinear cases. The Copula PC algorithm is a method for causal discovery from mixed data with an assumption that the data is drawn from a Gaussian copula model (Cui et al., 2016). The Copula PC algorithm implements Gibbs sampling based on the extended rank likelihood, then estimates a scale matrix and degrees of freedom from the Gibbs samples. The scale matrix substitutes for a correlation matrix and the degrees of freedom acts as the effective number of data points for the conditional independence test. The Copula PC algorithm can be used to learn causal graphs from mixed data containing binary, ordinal, and continuous variables but is not appropriate for non-binary categorical variables whose values cannot be ranked, such as blood type.

Kernel methods have been used to measure conditional (in)dependence. Bach and Jordan (2002a) proposed Kernel Generalized Variance (KGV) to measure conditional dependence and it can be used to learn a hybrid network from discrete and continuous variables. KGV allows discrete and continuous variables to be treated as Gaussians obtained from Mercer kernels in a feature space. Kernel Canonical Correlation (KCC) was proposed as a measure of independence (Bach and Jordan, 2002b). Gretton et al. (2005) proposed Kernel Mutual Information (KMI) to measure the degree of independence of random variables. Sun et al. (2007) developed a causal learning method for discrete and continuous variables by measuring the strength of statistical dependencies in terms of the Hilbert-Schmidt norm of kernel-based cross-covariance operators and used incomplete Cholesky decomposition. Fukumizu et al. (2007) proposed a method to measure conditional dependence using kernels based on a normalized cross-covariance operator on reproducing kernel Hilbert spaces. The Kernel PC algorithm was successfully developed by Tillman et al. (2009). KGV, KCC, KMI, and the Kernel PC algorithm use incomplete Cholesky factorizations for efficient implementation but this is not numerically stable.

In this study, there are two research questions: (i) how to handle mixed data in learning causal graphs? (ii) how to evaluate the learned graph structure when the true graph is unknown? We propose kernel functions and Kernel Alignment to handle mixed data in a way which allows existing algorithms (e.g. PC, FCI) to be used. In this research, we use the PC and FCI implementation from *pcalg* (Kalisch et al., 2012). The second goal is evaluating the structure of a learned graph when the ground truth is unknown. We introduce a new method called Modal Value of Edges Existence (MVEE) to evaluate different learned PAGs (partial ancestral graphs). We implement kernel functions to compute kernel matrices where we have a choice of kernel parameters. Different kernel parameter choices can produce different learned graphs. We use our MVEE method to choose between these various learned graphs, since in real applications we do not have the ground truth to help us make this choice.

2. PC algorithm and Fast Causal Inference

PC and FCI consist of two main stages: generating the skeleton and orienting the edges (Spirtes et al., 2001). A graph's skeleton is an undirected graph. The output of the PC algorithm is represented by Completed Partially Directed Acyclic Graph (CPDAG) (Kalisch and Buhlmann, 2007). Assume the distribution *P* of the random variables *X* is multivariate normal. For $i \neq j \in \{1, ..., p\}, k \subseteq \{1, ..., p\} \backslash \{i, j\}$ the partial correlation between X_i and X_j given $\{X_r; r \in k\}$ is denoted by $\rho_{i,j|k}$. $\rho_{i,j|k} = 0$ if and only if X_i and X_j are conditionally independent given $\{X_r; r \in k\}$. The estimated partial correlation $\hat{\rho}_{i,j|k}$ can be computed via regression, inversion of part of the covariance matrix or recursively. Fisher's z-transform is used to test whether the partial correlation is equal to zero or not. We reject the null hypothesis $H_0(i, j|k) : \rho_{i,j|k} = 0$ against the two-sided

alternative $H_A(i,j|k) : \rho_{i,j|k} \neq 0$ if $\sqrt{n-|k|-3}|Z(i,j|k)| > \Phi^{-1}(1-\alpha/2)$ (Kalisch and Buhlmann, 2007). PC is a causal algorithm that can be applied to a dataset assuming there are no latent variables. A latent variable is a variable that is not measured or not recorded (Colombo et al., 2012). FCI is a causal algorithm that allows the presence of latent variables (Spirtes et al., 2001). The early steps of FCI generate a skeleton graph then the edges are oriented. The output of FCI is as a Partial Ancestral Graph (PAG) (Colombo et al., 2012). PC and FCI apply conditional independence tests to generate a graph from a dataset.

3. Kernel Functions and Kernel Alignment for Mixed Data

In this paper we use the standard conditional independence test for Gaussian data which requires *partial correlations* Kalisch and Buhlmann (2007). Partial correlations can be computed from a correlation matrix Meloun and Militký (2011). We propose an easy way to compute a correlation matrix from mixed data using kernels. This approach is inspired by learning graphical models using KGV (Bach and Jordan, 2002a). The method was proposed by Bach and Jordan (2002a) who map data into a feature space using a set of Mercer kernels, with different kernels for different data types. They then treat all data equally as Gaussian in the feature space. Suppose $x_1, ..., x_m$ are m random variables with values in space $X_1, ..., X_m$. A Mercer kernel k_i is assigned to each input space X_i, with feature space F_i and feature map Φ_i. The random vectors of feature images $\phi = (\phi_1, ..., \phi_m) \triangleq ((\Phi_1(x_1), ..., \Phi_m(x_m)))$ has a covariance matrix C defined by blocks. Block C_{ij} is the covariance matrix between $\phi_i = \Phi_i(x_i)$ and $\phi_i = \Phi_j(x_j)$. Suppose $\phi^G = (\phi_1^G, ..., \phi_m^G)$ denotes a jointly Gaussian vector with the same mean and covariance as $\phi = (\phi_1, ..., \phi_m)$. The vector ϕ^G will be used as the random vector on which the learning of graphical model structure is based. Bach and Jordan (2002a) proposed a general framework that can be applied to any type of variable.

Computing sample covariances using the kernel trick needs high computation, and for efficient implementation it uses incomplete Cholesky decomposition. However, applying incomplete Cholesky decomposition to find a low-rank decomposition matrix might lead to numerically instability. Our idea is to transform each variable in the mixed data using a suitable kernel function for each data type into a *single* Gaussian variable in the feature space. We then use *kernel alignment* to compute pairwise 'covariances' and thus construct a pseudo-covariance matrix. In this way we obtain a pseudo-correlation matrix which can be used for conditional independence tests in the normal way. The prefix 'pseudo-' is used to emphasize that the resulting matrix is not a correlation matrix for the original variables.

3.1 Kernel Functions

A kernel is a function κ that for all $x, z \in X$ satisfies $\kappa(x,z) = \langle \phi(x), \phi(z) \rangle$, where ϕ is a mapping from X to an (inner product) feature space F, $\phi : x \to \phi(x) \in F$ (Shawe-Taylor and Cristianini, 2004). Given a set of vectors $S = \{x_1, ..., x_\ell\}$, the Gram matrix is an $\ell \times \ell$ matrix G whose entries are $G_{ij} = \langle x_i, x_j \rangle$. Using a kernel function κ to evaluate the inner products in a feature space with feature map ϕ, the associated Gram matrix has entries $G_{ij} = \langle \phi(x_i), \phi(x_j) \rangle = \kappa(x_i, x_j)$. In this case the matrix is often referred to as the kernel matrix. This paper uses the RBF kernel for continuous variables and the *Categorical kernel* Belanche and Villegas (2013) for discrete variables. The RBF Kernel is $\kappa(x_i, x_j) = \exp\left(-\frac{\|x_i - x_j\|^2}{2\sigma^2}\right), \sigma > 0$. The Categorical kernel is $\kappa(z_i, z_j) =$

$\begin{cases} h_\theta\left(P_Z\left(z_i\right)\right) & \text{if } z_i = z_j \\ 0 & \text{if } z_i \neq z_j. \end{cases}$. P is probability and $h_\theta(.)$ is a function that depends on the parameter θ. It is defined as $h_\theta(z) = \left(1 - z^\theta\right)^{1/\theta}, \theta > 0$. Note that our method is not restricted to these particular choices of kernel.

3.2 Kernel Alignment

Given a sample $S = \{x_1, ..., x_m\}$, the inner product between two kernel matrices is $\langle K_1, K_2\rangle = \sum_{i,j=1}^{m} K_1(x_i, x_j)K_2(x_i, x_j)$. The alignment between a kernel k_1 and k_2 of the sample S is defined as $\hat{A}(S, k_1, k_2) = \frac{\langle K_1, K_2\rangle}{\sqrt{\langle K_1, K_1\rangle\langle K_2, K_2\rangle}}$, where K_i is the kernel matrix of the sample S using kernel function k_i (Cristianini et al., 2001). We use Kernel Alignment as follows. Suppose, a mixed dataset consists of two variables (X and Y) and ℓ data points, i.e $X = \{x_1, x_2, ..., x_\ell\}$ and $Y = \{y_1, y_2, ..., y_\ell\}$. We compute kernel matrices K_X and K_Y using kernel functions k_1 and k_2, respectively. Kernel matrices K_X and K_Y correspond to variables X and Y, respectively. $K_X(i, j)$ can be thought of as the similarity between x_i and x_j, the ith and jth observed values of X. It is also the inner product of the x_i and x_j in the feature space. In the same way K_Y encodes similarities between observed values of Y and contains inner products in the feature space for Y.

The alignment $A(K_X, K_Y)$ is built from the inner product between K_X and K_Y (and so is a (normalised) inner product of inner products). Equation 1 defines Kernel Alignment.

$$A(K_X, K_Y) = \frac{\langle K_X, K_Y\rangle}{\sqrt{\langle K_X, K_X\rangle\langle K_Y, K_Y\rangle}} = \frac{\langle K_X, K_Y\rangle}{\|K_X\|\|K_Y\|} = \frac{\sum_{i,j=1}^{n} k_1(x_i, x_j)k_2(y_i, y_j)}{\sqrt{(\sum k_1(x_i, x_j)k_1(x_i, x_j))(\sum k_2(y_i, y_j)k_2(y_i, y_j))}} \quad (1)$$

Suppose a dataset $D = \{V_1, .., V_p\}$ consists of p variables and a set of kernel matrices $K = \{K_1, ..., K_p\}$ is computed from those variables. The kernel alignment matrix A is a $p \times p$ Gram matrix where each entry $A(s, t)$ is an inner product of the two vectors produced by flattening the kernel matrices K_s and K_t. A Gram matrix is a positive semi-definite matrix Shawe-Taylor and Cristianini (2004). Every positive semi-definite matrix is a covariance matrix, so a kernel alignment matrix A is a covariance matrix. Given any covariance matrix, it is possible to construct a Gaussian distribution with that covariance matrix. Thus the use of kernels in this proposed method can be viewed as a way of (implicitly) generating a Gaussian distribution whose covariance matrix is the kernel alignment matrix. This proposed method differs from that Bach and Jordan (2002a) in that the (implicitly constructed) Gaussian always has the same dimension as the original data (since unlike them we use kernel alignment), but in common with Bach and Jordan there is a separate kernel for each of the original variables. Like Bach and Jordan (2002a), we do not view this treating-variables-separately approach as unduly restrictive. (Note that for Copula PC the mapping is in the other direction: the observed variables are viewed as the result of a mapping from some latent Gaussian distribution.) The alignment between a kernel matrix and itself is $A(s, s) = A(K_s, K_s) = \frac{\langle K_s, K_s\rangle}{\sqrt{\langle K_s, K_s\rangle\langle K_s, K_s\rangle}} = 1$, so that $\sigma_s = \sqrt{1} = 1$. $A(s, s)$ and $A(t, t)$ can be viewed as variance σ_s^2 and σ_t^2, respectively. Thus, the entry of the correlation matrix can be defined as $\rho_{st} = \frac{\sigma_{st}}{\sigma_s\sigma_t} = \frac{A(K_s, K_t)}{A(K_s, K_s)A(K_t, K_t)} = \frac{A(K_s, K_t)}{(1)(1)} = A(K_s, K_t)$. Hereafter, this correlation matrix is called a pseudo-correlation matrix A. Note that the entries in any Kernel Alignment matrix are in the range $[0, 1]$ (Shawe-Taylor and Cristianini, 2004).

It is useful to compare Kernel Alignment to Distance correlation (Székely et al., 2007). Kernel Alignment values have the same range as Distance correlation values and both can be used for

mixed data. However, Kernel Alignment values are connected to the angle between pairs of vectors resulting from flattening kernel matrices (see equation 1). In contrast, the Distance correlation between two variables is related their Euclidean distance apart.

4. Modal Value of Edges Existence (MVEE)

Intersection-validation (InterVal) (Viinikka et al., 2018) is a method for evaluating CPDAG learning algorithms when the ground truth is unknown. The basic idea is to generate an *agreement graph* from the CPDAGs learned by different algorithms from the same dataset. The agreement graph is then used as a proxy for the ground truth.

Let $G = (V, E)$ be a graph with node set V and edge set $E \subseteq V \times V$. Denote a node pair (u, v) by uv and say that its type in G (or, in E) is bidirected, forward, backward, or nonadjacent if, respectively, both uv and vu, only uv, only vu, or neither belongs to E. Agreement graphs are *partial graphs* where a *partial graph* on a set of node pairs $S \subseteq V \times V$ is a pair (S, E) where $E \subseteq S$. An ordinary graph on V is obtained as a special case with $S = V \times V$.

Suppose, a set of CPDAGs $G = \{G_1, G_2, ..., G_k\}$ are learned from a dataset using algorithms $Alg = \{A_1, A_2, ..., A_k\}$ as input graphs. Figure 1 (a) shows the original InterVal method to generate an agreement graph, which is a partial graph, from two CPDAGs (dashed lines connect excluded node pairs not in S) (Viinikka et al., 2018). In this example, the agreement graph has the following node pairs $S = \{(A, B), (A, C), (B, C)\}$ and only one edge $E = \{(A, C)\}$.

InterVal applies a strict rule where it only includes a node pair in the agreement graph when all input graphs agree on that node pair. InterVal might produce an agreement graph containing no node pairs when there is no exact matching node pair type from all input graphs, especially when we create an agreement graph form PAGs. A PAG has three kinds of marks that form six different types of edges ($\circ\!\!\rightarrow$, $\leftrightarrow$, $\circ\!\!-\!\!\circ$, $\rightarrow$, $\circ\!\!-$, and $-$), so to get an exact match for all input graphs is difficult. Figure 1 (b) shows the agreement graph of two PAGs has two node pairs (A, D) and (B, C).

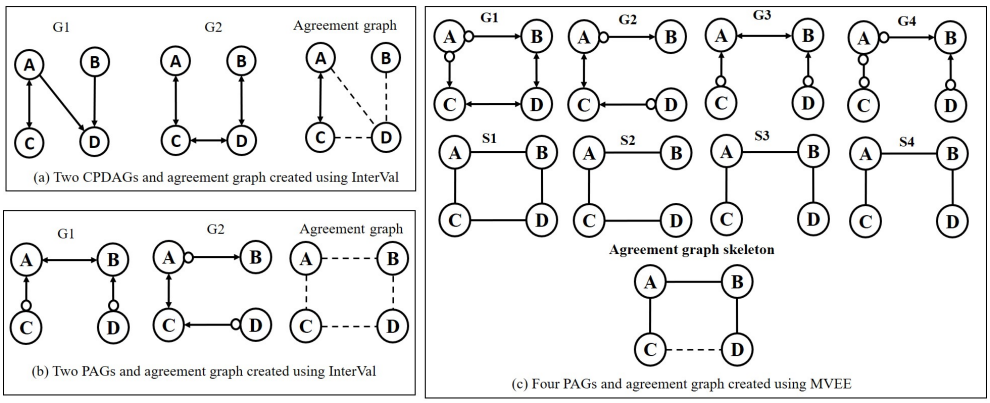

Figure 1: Generating agreement graph using InterVal and MVEE

Suppose we have competing algorithms that are used to generate learned graphs from the same dataset and they output the same type of graph (e.g. PAG). It is difficult to choose the best-learned graph when the true graph is unknown. In this situation, we assume that an edge (resp. non-edge) that exists in *most* of the learned graphs has a high possibility of being a true edge (resp. non-edge).

Based on this assumption, we propose a method based on the modal value of the edge type for any node pair to create a proxy to the true graph when the true graph is unknown.

Modal Value of Edges Existence (MVEE) is modified from the original InterVal concept proposed by Viinikka et al. (2018). Like InterVal MVEE adopts the idea of an agreement graph that is generated from input graphs and uses it as a proxy for the true graph. Specifically, we consider the PAG output from FCI algorithm. For a given pair of a graph $\mathcal{G}$ and a distribution P faithful to it, there may be two different FCI-PAGs that represent graph $\mathcal{G}$ but they will have the same skeleton (Colombo et al., 2012), so 'true' PAGs have the same skeleton. Since they will all share the same skeleton, it is useful to use a skeleton agreement graph as a proxy for the skeleton of the true PAG and use it to choose the 'best' graph from the output of competing PAG-learning algorithms. MVEE finds a skeleton agreement graph using a majority vote from the skeletons of a set of input graphs. This differs from InterVal where (i) all graphs must agree on a node pair for that node pair to be included in the agreement graph and where (ii) the agreement graph is not just a skeleton (see Fig 1(a)).

InterVal was designed for choosing between CPDAGs and MVEE for choosing between PAGs which contain more edge types than CPDAGs. Fig 1 (c) shows the input graphs represented by PAGs, skeleton of input graphs and the skeleton agreement graph generated using MVEE. In this example, a tie happens: two graphs have an edge between node C and D and two other graphs do not. In this situation, MVEE decides to abstain and so does not include (C, D) in the node pairs for MVEE's skeleton agreement graph (dashed line connects excluded node pair). If every node pair on input graphs results in a tie, the MVEE agreement graph has no node pairs. If there are no ties then S, the set of node pairs in the MVEE agreement graph is all possible pairs. If all input graphs were empty the MVEE agreement graph is (S, E) where S is $V \times V$ and E is the empty set.

5. Experimental Result and Discussion

5.1 Datasets and Experimental Design

In our experiments, mixed datasets are produced using forward sampling from a hybrid network. We use Conditional Linear Gaussian (CLG) models (Koller and Friedman, 2009). First we generate random DAGs containing 10, 20 and 30 nodes. Each variable is given a data type. For each different group of nodes, there are produced 10 different graphs and each graph produces five different datasets. We generate two groups of datasets according to the kind of data types: dataset 1 (binary, ordinal and continuous variables) and dataset 2 (binary, ordinal, categorical and continuous variables). The generated datasets are used in the simulation. The experiment is divided into several groups according to the algorithms and datasets. Experiment 1 uses dataset 1. The purpose of Experiment 1 is to compare the performance of our Kernel Alignment PC algorithm (KAPC) to the Copula PC algorithm. According to previous research by Cui et al. (2016), the Copula PC algorithm can be used to learn causal graphs from mixed data containing binary, ordinal, and continuous variables. For a fair comparison, we do not use categorical variables. Experiment 2 and experiment 3 use dataset 2. We only run KAPC and KAFCI for experiment 2 and experiment 3, respectively. The first step is computing the kernel matrix from each variable using a suitable kernel function for each data type. The RBF kernel and Categorical kernel are applied to compute kernel matrices from continuous and discrete variables, respectively. Let $P_i(\sigma, \theta)$ be the kernel parameters for the RBF and Categorical kernel, respectively. We apply centring to all kernel matrices. The detailed explanation of centering kernel matrices had been studied by Meila (2003). The second step is

computing a Kernel Alignment matrix from a set of centred kernel matrices. The third step is applying the Kernel Alignment matrix as input for conditional independence tests for learning a causal graph using PC/FCI. The quality of the structure of the learned graph is measured using Structural Hamming Distance (SHD) score (Tsamardinos et al., 2006). The SHD score is normalized using a method proposed by Malone et al. (2015). Lower SHD score means that the learned graph has less mismatch to the true graph and has a high similarity to the true graph. The experiments use generated datasets. The experiments are run for $N = \{1000, 2000, 3000, 5000\}$ data points. There are no missing values in the dataset.

5.2 Experiment using Benchmark Datasets

To further investigate the performance of the proposed method, we also ran experiments using the datasets *gaussian.test* (sampled from a Gaussian) and *clgaussian.test* from the *R* package bnlearn and *gmD* (discrete variables) from the pcalg R package. The distribution from which *clgaussian.test* was sampled contains one Gaussian variable, 4 discrete variables (2 binary and 2 categorical variables) and 3 conditional Gaussian variables.

In our analysis of experiments we use the following terms. A *true edge* is an edge in the learned graph that has exactly the same position and marks as the edge on the true graph. A *missing edge* is an edge that not appear in the learned graph but exists in the true graph. An *extra edge* is an edge in the learned graph that does not exist in the true graph. A *wrongly marked edge* is an edge in the learned graph that exists in the true graph but has a different mark on one or both sides.

KAPC uses the RBF kernel for continuous variables and the Categorical kernel for discrete variables. Figure 2 (a) shows the learned graphs generated by the original PC algorithm and KAPC using *gaussian.test* data at $\alpha = 0.05$. KAPC successfully generates all true edges but it produces one extra edge (red edge). Figure 2 (b) shows the true graph and learned graphs generated from the *gmD* dataset at $\alpha = 0.1$. The original PC algorithm and KAPC produce the same graph skeleton as the true graph skeleton but they have one wrongly marked edge. Figure 2 (c) shows the true graph and learned graph generated from *clgaussian.test* data using KAPC with $\alpha = 0.05$. KAPC generates a learned graph containing 10 true edges and 1 extra edge.

5.3 Experiment using Generated Datasets

We run KAPC and Copula PC (Cui et al., 2016). The first experiment uses kernel parameters for RBF kernel σ and Categorical kernel θ, $P = \{P_1(\sigma = 0.001, \theta = 0.5), P_2(\sigma = 0.01, \theta = 1), P_3(\sigma = 0.001, \theta = 1), P_4(\sigma = 0.01, \theta = 1.5), P_5(\sigma = 0.001, \theta = 1.5)\}$. The experiment using generated datasets uses $\alpha = 0.1$ and $\alpha = 0.05$. The experiment results in this section shows that different values of the kernel parameters produce different learned graphs. Figure 3 **A** shows the result for KAPC using different kernel parameters (P1-P5) and Copula PC algorithm (C). Copula PC produces lower SHD scores for graphs with 10 nodes and KAPC using kernel parameter P1, P3, and P5 slightly outperforms Copula PC for graphs with 20 and 30 nodes. In the second and third experiments, we only run KAPC and KAFCI to learn causal graphs from mixed datasets containing categorical, binary, ordinal and continuous variables. The second and third experiment uses kernel parameters for RBF kernel $\sigma = \{1, 0.1, 0.01, 0.001, 0.0001\}$ and Categorical kernel $\theta = 2$. The general result of the second experiment is similar to the result of the first and third experiments, KAPC produces a lower SHD score if it is given the proper kernel parameter. In the third experiment, we randomly delete 1, 4, and 8 variables from graph 10, 20, and 30 nodes, respectively. Those variables represent

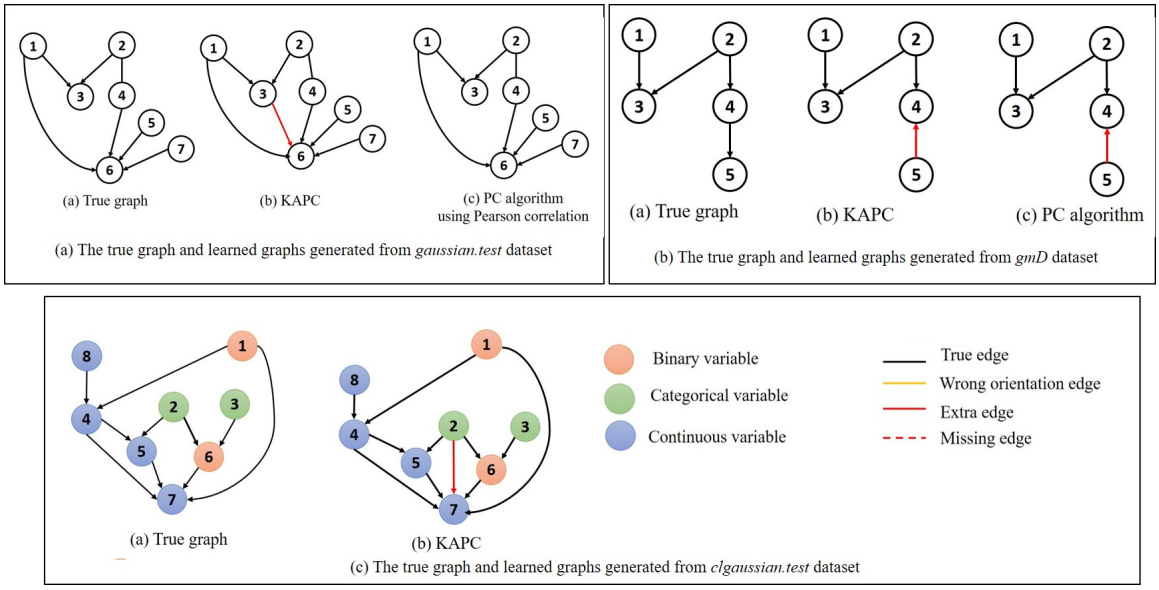

Figure 2: The true graph and learned graphs generated from Benchmark datasets

latent variables. Kernel matrices and kernel alignment for conditional independence test for FCI are computed from the remaining variables. Figure 3 **B** shows the SHD scores of KAFCI using different kernel parameters. The learned graphs generated using KAFCI are used as input for MVEE.

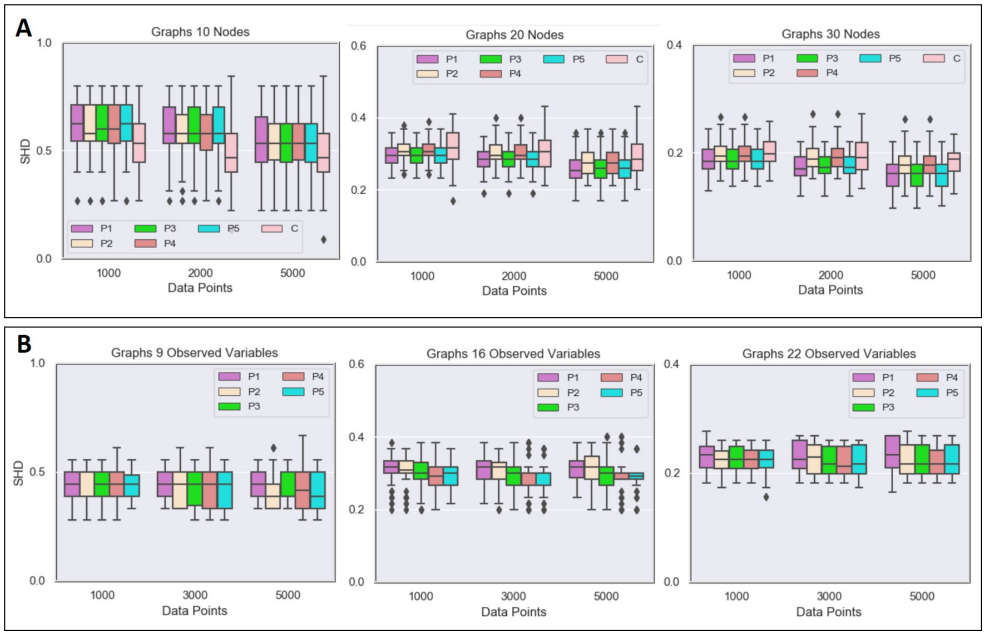

Figure 3: (A) Experiment Results KAPC and (B) Experiment Result KAFCI

The kernel-based approach and Copula approach (Cui et al., 2016) compute a matrix, which is used as a correlation matrix, from mixed data, it then uses this matrix as input for conditional

independence tests in the PC algorithm. The main reason to develop a kernel-based approach is that it provides a procedure to treat categorical variables similarly to the binary, ordinal and continuous variable in learning a causal graph from mixed data using PC algorithm and FCI. The kernel-based approach uses n data points for conditional independence tests, where n is the number of data points in the dataset. The Copula PC algorithm computes the scale matrix and degree of freedom and uses them as input for conditional independence test (Cui et al., 2016). The Copula PC algorithm uses as the number of data points around 80% of the actual number of data points. PC Algorithm and FCI work well if they are given sufficient data. Having more data typically leads to more accurate learned graphs.

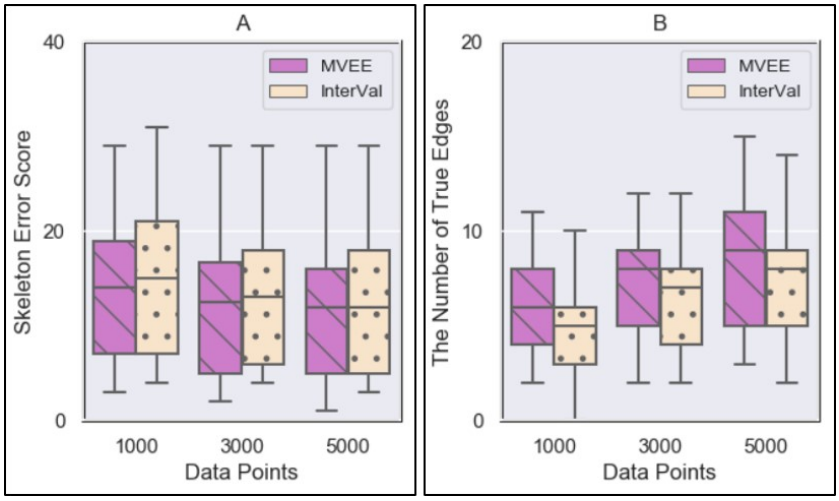

Figure 4: The evaluation of agreement graph

The advantage of KAPC is that, unlike Copula PC, it works for categorical variables as well as binary, ordinal, and continuous variables. Generally, the kernel-based approach runs faster than the Copula model. The Copula PC algorithm does not work for 48 datasets due to failure to generate the scale matrices. We do not investigate the reason for failure because it is not our main concern. Meanwhile the proposed method succeeds in generating the kernel alignment matrix for all datasets used in the simulation. The issue in applying the kernel-based approach is how to choose the best value of kernel parameter for the kernel function. The kernel-based approach can be implemented to learn a causal graph from mixed data as long as there is a suitable kernel function for particular variables. The kernel-based approach gives a simple solution for learning causal graphs from mixed data using the PC algorithm and FCI. The experiments using different values of parameters show that KAPC and KAFCI produce graphs with lower SHD score when it is given proper kernel parameter values. This proposed method might generate the wrong graph when we use the wrong kernel parameter.

We use MVEE to estimate the best learned graph when the true graph is unknown. First, we generate learned graphs from the same dataset using KAFCI with different kernel parameters. KAFCI with different kernel parameters can be viewed as different algorithms. An agreement graph is generated from the skeletons of the learned graphs from the same dataset. In this experiment, we use learned graphs from the third experiment.

The InterVal approach uses subsamples of the data to analyze how close the resulting graph is to the agreement graph. Our goal is different from InterVal, so we do not use exactly the same approach. Our goal is to use MVEE to estimate the best learned graph produced by KAFCI with different kernel parameters when the true graph is unknown. Suppose, $G = \{G_1, G_2, ..., G_n\}$ is a set of learned graphs generated from the same dataset using different algorithms $Alg = \{A_1, A_2, ..., A_n\}$. These learned graphs are used as input for MVEE to produce an agreement graph S_0. We apply *partial Hamming distance (PHD)* from Viinikka et al. (2018) to compute the mismatch between the skeleton agreement graph and the skeleton of a learned graph. We call the score as Partial Skeleton Error (PSE) score. The learned graph with the lowest PSE score is considered the best learned graph.

To evaluate MVEE, we first find which learned graphs have minimal (i.e. best) SHD when compared to the true graph: this is the set of optimal learned graphs. We generate an MVEE agreement graph from the learned graphs generated using KAFCI with different values of kernel parameters from the same dataset. We estimate a best learned graph by computing the PSE score between the learned graphs and the agreement graph; the estimated best learned graph is some learned graph with minimal PSE score. In our evaluation of MVEE we have found that MVEE identifies an optimal graph 91.56% of the time.

We analyze the agreement graphs to understand their performance as a proxy for the true graph. First, we compute the mismatch between the skeleton of the agreement graph and the skeleton of the true graph. This score is named the skeleton error score. Figure 4 (**A**) shows the MVEE skeleton agreement graph has less mismatch than InterVal skeleton agreement graph. Figure 4 (**B**) shows MVEE skeleton agreement graphs have more true edges than Interval skeleton agreement graphs. Unsurprisingly both methods are more accurate as the number of datapoints increases.

6. Conclusion

In conclusion, our kernel-based approach is a promising method for learning a causal graph from mixed data containing categorical, binary, ordinal, and continuous variables using PC and FCI. The kernel-based approach offers better treatment for the categorical variable in mixed data which cannot be handled properly using the Copula PC developed by Cui et al. (2016). MVEE is a method that successfully chooses the best learned graph when the true graph is unknown. In the experiment to choose the best learned graph, MVEE produces accuracy 91.56%. For future research, we will use kernel functions for missing data (i.e kernel extensions for missing data proposed by Nebot-Troyano and Belanche-Munoz (2009)) to substitute the regular kernel functions in this paper, so KAPC/KAFCI can be used to learn a causal graph from mixed data containing missing values. MVEE is only based on the structure of the graph without analyzing its statistical meaning. For future work, it is possible to combine MVEE with statistical based evaluation method to develop a sophisticated evaluation method when the true graph is unknown.

Acknowledgments

This work was supported through a scholarship managed by Lembaga Pengelola Dana Pendidikan Indonesia (Indonesia Endowment Fund for Education).

References

F. R. Bach and M. I. Jordan. Learning graphical models with mercer kernels. In *Proceedings of the 15th International Conference on Neural Information Processing Systems*, pages 1033–1040, 2002a.

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

L. A. Belanche and M. A. Villegas. Kernel functions for categorical variables with application to problems in the life sciences. In *The 16 International Conference of the Catalan Association of Artificial Intelligence*, pages 171–180, 2013.

D. Colombo, M. H. Maathuis, M. Kalisch, and T. S. Richardson. Learning high-dimensional directed acyclic graphs with latent and selection variables. *The Annals of Statistics*, 40:294–321, 2012.

N. Cristianini, J. Shawe-Taylor, A. Elisseeff, and J. Kandola. On kernel-target alignment. In T. G. Dietterich, S. Becker, and Z. Ghahramani, editors, *Neural Information Processing Systems 14 (NIPS 2001)*, pages 367–373. MIT Press, 2001.

R. Cui, P. Groot, and T. Heskes. Copula PC algorithm for causal discovery from mixed data. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 377–392, Riva del Garda, 2016. Springer.

F. Fukumizu, A. Gretton, X. Sun, and B. H. Schölkopf. Kernel measures of conditional dependence. In *Proceedings of the 20th International Conference on Neural Information Processing Systems*, pages 489–496, 2007.

A. Gretton, R. Herbrich, A. Smola, O. Bousquet, and B. Schölkopf. Kernel methods for measuring independence. *Journal of Machine Learning Research*, 6:2075–2129, 2005.

M. Kalisch and P. Buhlmann. Estimating high-dimensional directed acyclic graphs with the PC algorithm. *Journal of Machine Learning Research*, 8:613–636, 2007.

M. Kalisch, M. Machler, D. Colombo, M. H. Maathuis, and P. Buhlmann. Causal inference using graphical models with the R package pcalg. *Journal of Statistical Software*, 47:1–26, 2012.

D. Koller and N. Friedman. *Probabilistic Graphical Models Prinsiples and Techniques*. The MIT Press, 2009.

B. Malone, M. Järvisalo, and P. Myllymäki. Impact of learning strategies on the quality of bayesian networks: an empirical evaluation. In *Proceedings of the Thirty-First Conference on Uncertainty in Artificial Intelligence*, pages 562–571, Amsterdam, Netherlands, 2015. AUAI Press.

M. Meila. Data centering in feature space. In *AISTATS*, 2003.

M. Meloun and J. Militký. *Statistical Data Analysis A Practical Guide*. India PVT. LTD., 2011.

G. Nebot-Troyano and L. Belanche-Munoz. A kernel extension to handle missing data. In M. Bramer, R. Ellis, and M. Petridis, editors, *The Twenty-ninth SGAI International Conference on Innovative Techniques and Applications of Artificial Intelligence*, pages 165–178. Springer, 2009.

V. K. Raghu, J. D. Ramsey, A. Morris, D. V. Manatakis, P. Sprites, P. K. Chrysanthis, C. Glymour, and P. V. Benos. Comparison of strategies for scalable causal discovery of latent variable models from mixed data. *International Journal of Data Science and Analytics*, 6:33–45, 2018.

J. Shawe-Taylor and N. Cristianini. *Kernel Methods for Pattern Analysis*. Cambridge University Press, New York, NY, USA, 2004.

P. Spirtes, C. Glymour, and R. Scheines. *Causation, Prediction, and Search*. MIT-Press, New York, USA, 2001.

X. Sun, D. Janzing, B. H. Scholkopf, and K. Fukumizu. A kernel-based causal learning algorithm. In *The 24th international conference on Machine learning*, pages 855–862, 2007.

G. J. Székely, M. L. Rizzo, and N. K. Bakirov. Measuring and testing dependence by correlation of distances. *The Annals of Statistics*, 35(6):2769–2794, 2007.

R. E. Tillman, A. Gretton, and P. Spirtes. Nonlinear directed acyclic structure learning with weakly additive noise models. In *Proceedings of the 22nd International Conference on Neural Information Processing Systems*, pages 1847–1855, 2009.

M. Tsagris, G. Borboudakis, V. Lagani, and I. Tsamardinos. Constraint-based causal discovery with mixed data. *International Journal of Data Science and Analytics*, 6(1):19–30, 2018.

I. Tsamardinos, L. E. Brown, and C. f. Aliferis. The max-min hill-climbing Bayesian Network structure learning algorithm. *Machine Learning*, pages 31–78, 2006.

J. Viinikka, R. Eggeling, and M. Koivisto. Intersection-validation: A Method for Evaluating Structure Learning without Ground Truth. In *The 21st International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 1–5, Lanzarote, Spain, 2018. PMLR.