Respecting Markov Equivalence in Computing Posterior Probabilities of Causal Graphical Features

Eun Yong Kang
Department of Computer Science
University of California, Los Angeles
Los Angeles, CA 90095 USA
ekang@cs.ucla.edu

Ilya Shpitser
Department of Epidemiology
Harvard School of Public Health
Boston, MA 02115 USA
ishpitse@hsph.harvard.edu

Eleazar Eskin
Department of Computer Science
University of California, Los Angeles
Los Angeles, CA 90095 USA
eeskin@cs.ucla.edu

Abstract

There have been many efforts to identify causal graphical features such as directed edges between random variables from observational data. Recently, Tian et al. proposed a new dynamic programming algorithm which computes marginalized posterior probabilities of directed edge features over all the possible structures in $O(n3^n)$ time when the number of parents per node is bounded by a constant, where n is the number of variables of interest. However the main drawback of this approach is that deciding a single appropriate threshold for the existence of the directed edge feature is difficult due to the scale difference of the posterior probabilities between the directed edges forming v-structures and the directed edges not forming v-structures. We claim that computing posterior probabilities of both adjacencies and v-structures is necessary and more effective for discovering causal graphical features, since it allows us to find a single appropriate decision threshold for the existence of the feature that we are testing. For efficient computation, we provide a novel dynamic programming algorithm which computes the posterior probabilities of all of $\frac{n(n-1)}{2}$ adjacency and $n\binom{n-1}{2}$ v-structure features in $O(n^3 3^n)$ time.

Introduction

Background

In many scientific problems, identifying causal relationships is an important part of the solution. The gold standard for identifying causal relationships is a randomized experiment. However in many real world situations, the randomized experiment cannot be performed due to various reasons such as ethical, practical or financial issues. Therefore identifying causal relationships from observational data is an unavoidable and important step in understanding and solving many scientific problems.

A popular tool to represent causal relationships is a graphical model called a causal diagram (Pearl 1988), (Pearl 2000). A causal diagram is a directed acyclic graph (DAG), which consists of nodes and directed edges. Nodes represent variables of interest while directed edges between nodes represent directed causal influence between two end nodes of those edges. A causal diagram is a data generating model, with a bundle of directed arrows pointing to each node representing the causal mechanism which determines values of that node in terms of values of that node's direct causes.

The ideal goal of inferring causal relationships from observational data is to identify the exact data generating model. However inferring causal relationships has a fundamental limitation, if we only use observational data. That is, the best we can infer with observational data is the Markov equivalence class (Verma and Pearl 1990) of the data generating model.

The Markov equivalence class is the set of graphical models which all represent the same set of conditional independence assertions among observable variables. All graphical models in a Markov equivalence class share the same set of adjacencies and v-structures. Here a v-structure represents a node triple where two nodes are non-adjacent parents of another node. If we find the correct Markov equivalence class of the data generating model, then the Markov equivalence class contains the true data generating model as its member. The true generating model and other models in its Markov equivalence class must agree on directions of certain edges, while possibly disagreeing on others. Since all models in the Markov equivalence class share v-structures, all edges taking part in v-structures must have the same direction in all models in the class. Furthermore, certain other edges must have the same direction in the entire class as well. These edges have the property that reversing their direction would entail the creation of a v-structure which is not present in the class.

Previous Approaches and Their Limitations

Computing marginal posterior probabilities of graphical features considering all the possible causal network structures is computationally infeasible due to the exponential number of possible network structures, unless the number of variables is small. Due to this problem, approximation methods for computing marginal posterior probabilities of graphical features such as directed or undirected edges have been proposed. Madigan and York (1995) applied a Markov chain Monte Carlo (MCMC) method in the space of all the network structures. Friedman and Koller (2003) developed a more efficient MCMC procedure which applies to the space of all orderings of variables. The problem with these approximation methods is that inference accuracy is not guaranteed in the finite runs of MCMC.

Several methods have been developed to compute the exact posterior probability of graphical features using dynamic programming algorithms. Koivisto and Sood (2004) proposed a DP algorithm to compute the marginal posterior probability of a single undirected edge in $O(n2^n)$ time, where n is the number of variables. Koivisto (2006) also developed a DP algorithm for computing all $n(n-1)$ undirected edges in $O(n2^n)$ time and space. In the above two DP algorithms, the number of adjacent node is bounded by a constant. These methods compute the posterior probability by marginalizing over all the possible orderings of variables in the graphical models. These DP approaches and order MCMC approaches require a special prior for the graphical structure when summing over all the possible orders of the variables. The drawback of these structure priors is that due to averaging DAGs over variable ordering space instead of over all possible structures, this structure prior results in biased posterior probability of features and leads to incorrect inferences (Ellis and Wong 2008). To fix this bias problem, new MCMC algorithms were proposed recently (Eaton and Murphy 2007), (Ellis and Wong 2008). But these sampling-based algorithms still cannot compute the exact posterior probability.

Tian and He (2009) proposed a new dynamic programming algorithm which computes marginalized posterior probabilities of directed edges over all the possible structures in $O(n3^n)$ total time when the number of parents per node is bounded by a constant. This algorithm requires longer running time than Koivisto's approach (2006) but it can compute exact or unbiased posterior probabilities since it marginalizes over all the possible structures instead of all the possible orders of variables.

However there is a problem with the approach of predicting only directed edges for the two following reasons. First, if we predict a directed edge for two fixed nodes a and b, there are only three possible predictions: $a \rightarrow b$, $a \leftarrow b$, or $a \quad b$, where $a \quad b$ means no edge between a and b. If the presence of an edge is predicted, then there are two possibilities. If the edge is constrained to always point in the same direction in every model in the Markov equivalence class, then, in the limit, its posterior should approach 1. On the other hand, if an edge can point in different directions in different models in the Markov equivalence class, then, in the limit, the posterior probability of a particular orientation should approach the fraction of the models in the class which agree on this orientation. For example, in the class showed in Figure 1, in the limit, $P(c \rightarrow d) = \frac{1}{3}$ while, $P(c \leftarrow d) = \frac{2}{3}$. Since the scale of posterior probabilities of these two kinds of edge features is different, this makes it difficult to decide a single appropriate posterior threshold. However one can say that scale difference problem in the posterior computation can be addressed by direct scale adjustment. But direct scale adjustment is not possible, since we do not know in advance which edges are compelled to point in the same direction across the equivalence class, and which are not. Second, we usually have only limited samples available in real world situations. This can lead to inaccurate posterior probability computation. This small sample error can make the scale difference problem worse when considering only directed edges for posterior computation, which leads to determining an inappropriate decision threshold.

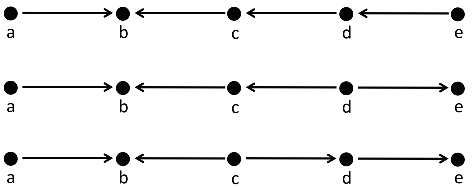

Figure 1: Three graphical models from a Markov equivalent class

Proposed Approach

The scale difference of the posterior probabilities of directed edge features between different kinds of edges makes it difficult to determine a single appropriate threshold. Therefore to correctly recover the model, inferring only directed edge features is not sufficient.

The following well-known theorem (Verma and Pearl 1990) will guide us in the choice of features.

Theorem 1 *Any two Markov-equivalent graphs have the same set of adjacencies and the same set of v-structures, where a v-structure is a node triple $\langle A, B, C\rangle$, where A, C are non-adjacent parents of B.*

Theorem 1 gives us a simple set of features to test in order to completely determine the Markov equivalence class of the data generating model: adjacencies and v-structures. If the posterior probabilities of adjacency features and v-structure features are computed by marginalizing over all possible DAGs, the scale of the posterior probabilities of these two graphical features is the same unlike the posterior probabilities of the directed edge features. Therefore we propose that posteriors of both adjacency and v-structure features must be computed, if one wants to recover the Markov equivalence class of true data generating model by computing posteriors of the graphical features. For a graph of size n, our feature vector consists of $\frac{n(n-1)}{2}$ binary features for adjacencies and $n\binom{n-1}{2}$ binary features for v-structures. For the efficient posterior computation, we provide a novel dynamic programming algorithm which computes the posterior probabilities of all of $\frac{n(n-1)}{2}$ adjacency and $n\binom{n-1}{2}$ v-structure features in $O(n^3 3^n)$ time.

Posterior Computation of Graphical Features

To learn the causal structure, we must compute the posteriors of causal graphical features such as directed edges. The posterior probability of a graphical feature can be computed by averaging over all the possible DAG structures.

The posterior probability of a certain DAG G given observational data D can be represented as follows:

$$P(G|D) \propto \mathcal{L}(G)\pi(G), \tag{1}$$

where $\mathcal{L}(G)$ is the likelihood score of the structure given the observation, and $\pi(G)$ is the prior of this DAG structure. The likelihood score computation can be performed based on the underlying model representing the causal relationships. We will talk more about the model that we used for our experiments in the experiment section.

In this paper, we take widely accepted assumptions including *parameter independence*, *parameter modularity* (Geiger and Heckerman 1994), and *prior modularity* which simplifies our learning task. Parameter independence means that each parameter in our network model is independent. Parameter modularity means that if a node has the same set of parents in two distinct network structures, the probability distribution for the parameters associated with this node is identical in both networks. Prior modularity means that the structural prior can be factorized into the product of the local structure priors. Now if we assume global and local parameter independence, parameter modularity, and prior modularity, $\mathcal{L}(G)$ and $\pi(G)$ can be factorized into the product of local marginal likelihoods and local structural prior scores respectively.

$$P(G|D) \propto \prod_{i=1}^{n} \mathcal{L}(V_i|Pa_{V_i})\pi_i(Pa_i), \tag{2}$$

where n is the number of variables, V_i is the i^{th} node, Pa_{V_i} is the parent set of node V_i, and $\pi_i(Pa_i)$ is the structure prior for the local structure which consists of node i and its parent set Pa_i. This factorization implies that conditional independencies in the distribution of observable variables are mirrored by a graphical notion of "path blocking" known as d-separation (Pearl 1988), in a sense that any time two variable sets $\mathbf{X}, \mathbf{Y}$ are d-separated by $\mathbf{Z}$ in a graph G induced by the underlying causal model , then $\mathbf{X}$ is conditionally independent from $\mathbf{Y}$ given $\mathbf{Z}$ in the distribution of variables in underlying causal model.

In stable (Pearl and Verma 1991) or faithful (Spirtes, Glymour, and Scheines 1993) models, the converse is also true: conditional independence implies d-separation. Informally, in faithful models all probabilistic structure embodied by conditional independencies is precisely characterized by graphical structure, there are no extraneous independencies due to "numerical coincidences" in the observable joint distribution. Faithfulness is typically assumed by causal discovery algorithms since it allows us to make conclusions about the graph based on the outcome of conditional independence tests. In addition, we assume that unobserved variables which are involved in the data generating process are jointly independent. Finally, since we are interested in learning causal, not just statistical models, we make the causal Markov assumption; that is, each variable V_i in our model is independent of all its non-effects given its direct causes (Pearl and Verma 1991).

Now the posterior probability for a feature f of interest can be computed by averaging over all the possible DAG structures which can be expressed as:

$$P(f|D) = \frac{P(f,D)}{P(D)} = \frac{\sum_{G_j \in G_{f+}} \prod_{i \in G_j}^{n} \mathcal{L}(V_i|Pa_{V_i})\pi_i(Pa_i)}{\sum_{G_j \in G} \prod_{i \in G_j}^{n} \mathcal{L}(V_i|Pa_{V_i})\pi_i(Pa_i)} \tag{3}$$

where G_{f+} is the set of DAG structures where f occurs, G is the set of all the possible DAG structures, from i to n is nodes in G_j, and $\pi_i(Pa_i)$ is a prior over local structures. If we introduce an indicator function to equation (3), $P(f, D)$ can be expressed as follows:

$$P(f,D) = \sum_{G_j \in G} \prod_{i \in G_j}^{n} f_i(Pa_i)\mathcal{L}(V_i|Pa_{V_i})\pi_i(Pa_i), \tag{4}$$

where $f_i(Pa_i)$ is an indicator function with the value of 1 or 0. The function $f_i(Pa_i)$ can have the value 1, if the G_j has the feature f of interest (directed edge, adjacency or v-structure), or 0 otherwise. For the directed edge feature $j \rightarrow i$, if $j \in Pa_i$ then $f_i(Pa_i)$ will equal 1. For the v-structure feature $a \rightarrow i \leftarrow b$, if $a \in Pa_i$, $b \in Pa_i$, $a \notin Pa_b$, and $b \notin Pa_a$ then $f_i(Pa_i)$ will have 1. We can compute (3), if we know $P(f, D) = \sum_{G_j \in G} \prod_{i \in G_j}^{n} f_i(Pa_i)\mathcal{L}(V_i|Pa_{V_i})\pi_i(Pa_i)$, since $P(f, D)$ computation with $f = 1$ will gives us $P(D)$. Since the number of possible DAGs is super-exponential $O(n!2^{\frac{n(n-1)}{2}})$, the brute-force approach of posterior computation of all the possible graphical features takes $O(n(n-1)n!2^{\frac{n(n-1)}{2}})$ for all $n(n-1)$ directed edge features and $O(n\binom{n-1}{2}n!2^{\frac{n(n-1)}{2}})$ for all $n\binom{n-1}{2}$ v-structure features, where n is the number of nodes in DAG. In the following sections, we will explain how to compute $\sum_{G_j \in G} \prod_{i \in G_j}^{n} f_i(Pa_i)\mathcal{L}(V_i|Pa_{V_i})\pi_i(Pa_i)$ by dynamic programming more efficiently. For simplicity, we let $B_i(Pa_i) = f_i(Pa_i)\mathcal{L}(V_i|Pa_{V_i})\pi_i(Pa_i)$.

Previously Proposed Method

Posterior Computation of Directed Edge Features by Exploiting Root Nodes and Sink Nodes In this section, we describe how to compute $P(f, D) = \sum_{G_j \in G} \prod_{i \in G_j}^{n} B_i(Pa_i)$ for all directed edge features using DP algorithm which was introduced in (Tian and He 2009). For the summation of $\prod_{i \in G_j}^{n} B_i(Pa_i)$ over all possible G by dynamic programming, we use the set inclusion-exclusion principle for computing unions of the overlapping sets. We first split DAGs into sets whose DAG elements have common root nodes or sink nodes. Then $P(f, D)$ can be computed by union of those sets. Now we describe the computation of $P(f, D)$ by exploiting sets with common root nodes. Let V be the set of all variables of interest and let $\zeta^{+}(S)$ be the set of DAGs over V such that all variables in $V - S$ are root nodes. Then for any $S \subseteq V$, we define $RR(S)$ as follows:

$$RR(S) = \sum_{G \in \zeta^{+}(S)} \prod_{i \in S}^{n} B_i(Pa_i). \tag{5}$$

If we apply the set inclusion-exclusion principle to $RR(V)$, then we can derive a recursive formula for the dynamic programming in terms of $RR(V)$ as follows:

First, we define the $A_i(S)$ as follows:

$$A_i(S) = \sum_{Pa_i \subseteq S} B_i(Pa_i) = \sum_{Pa_i \subseteq S} f_i(Pa_i)\mathcal{L}(V_i|Pa_{V_i})\pi_i(Pa_i). \tag{6}$$

Then Equation (5) can be rewritten as follows:

$$RR(S) = \sum_{k=1}^{|S|} (-1)^{k+1} \sum_{T \subseteq S, |T| = k} RR(S - T) \prod_{j \in T} A_j(V - S). \tag{7}$$

Now if we see Eq. (7)[1], then $RR(V)$ can be recursively computed by dynamic programming with base case $RR(\emptyset) = 1$ and $RR(j) = A_j(V - j)$. For each $A_j(S)$, with the assumption of fixed number of maximum parents k, we can compute it using the truncated Möbius transform algorithm in time $O(k2^n)$ (Koivisto and Sood 2004). Since $RR(S)$ for all $S \subseteq V$ can be computed in $\sum_{k=0}^{n} \binom{n}{k} 2^k = 3^n$, $RR(V) = P(f_{u \to v}, D)$ can be computed in $O(3^n)$ time for the fixed maximum number of parents. Therefore all $n(n-1)$ directed edge features can be computed in $O(n^2 3^n)$ time.

Now we describe how to compute $P(f, D)$ by computing union of DAG sets with common sink nodes. For all $S \subset V$, let $\zeta(S)$ denote the set of all possible DAGs over S. Then for any $S \subseteq V$, we define $H(S)$ as follows:

$$H(S) = \sum_{G \in \zeta(S)} \prod_{i \in S} B_i(Pa_i). \tag{8}$$

If we apply the set inclusion-exclusion principle to $H(S)$, then we can derive a recursive formula for the dynamic programming in terms of $H(S)$ as follows:

$$H(S) = \sum_{k=1}^{|S|} (-1)^{k+1} \sum_{T \subseteq S, |T| = k} H(S-T) \prod_{j \in T} A_j(S-T). \tag{9}$$

Equation (9) [1] can be efficiently computed by dynamic programming. Each $H(S)$ can be computed in time $\sum_{k=1}^{|S|} \binom{|S|}{k} k = |S| 2^{|S|-1}$. All $H(S)$ for $S \subseteq V$ can be computed in time $\sum_{k=1}^{n} \binom{n}{k} k 2^{k-1} = n3^{n-1}$. In the next section, we will explain how to combine $H(S)$ and $RR(S)$ to compute posteriors of all directed edge features which is a factor of n faster than DP algorithm using only $RR(S)$.

Combining $H(S)$ and $RR(S)$ to Compute Posteriors of All Directed Edge Features To compute the posteriors of all features efficiently, we extract the terms which are feature specific from the posterior computation. After we identify the terms which are common for all posterior computations, we can precompute those terms, then reuse them for all posterior computation. Let V be all the nodes in the DAG. For a fixed node v, summation over all DAGs can be decomposed into the set of nodes which are ancestors(U) of v, non-ancestors $V - U - \{v\}$ and feature specific components. Now the computation of posterior amounts to the summation over DAGs over U, which corresponds to $H(U)$ and the summation over DAGs over $V - U - \{v\}$ with $U \cup \{v\}$ as root nodes, which corresponds to $RR(V - \{v\} - U)$ and feature specific components which corresponds to $A_v(U)$.

We define $K_v(U)$ which corresponds to $RR(V - \{v\} - U)$ for any $v \in V$ and $U \subseteq V - \{v\}$ as follows:

$$K_v(U) = \sum_{T \subseteq V - \{v\} - U} (-1)^{|T|} RR(V - \{v\} - U - T) \prod_{j \in T} A_j(U). \tag{10}$$

Then the posteriors of $u \to v$ can be computed by summation over all the subset of $V - \{v\}$ of three terms as we mentioned above. We can express this as follows:

$$P(f_{u \to v}, D) = \sum_{U \subseteq V - \{v\}} A_v(U) H(U) K_v(U). \tag{11}$$

In the Eq. (11), $A_v(U)$ is a feature specific term and $H(U)$ and $K_v(U)$ are feature independent terms. To compute the posteriors for all directed edge features, first we have to compute the feature independent terms which include B_i, A_i, RR, H, and K_i under the condition of $f = 1$. Since for each directed edge feature, $f_i(Pa_i)$ will have different values, we need to recompute $A_v(S)$, where $S \subseteq V - \{v\}$. With the assumption of a fixed maximum indegree k, the computation of feature independent terms takes $O(n3^n)$ in time. And since each feature specific term $A_v(U)$ can be computed in $O(k2^n)$ time for all $U \in V - \{v\}$, it take $O(kn^2 2^n)$ for all $n(n-1)$ directed edge features. Therefore total computational complexity is $O(n3^n + kn^2 2^n) = O(n3^n)$ for all $n(n-1)$ directed edge features.

Novel Proposed Method

In this section, we describe a novel dynamic programming algorithm which computes the posterior probabilities of all $\frac{n(n-1)}{2}$ adjacency and $n\binom{n-1}{2}$ v-structure features in $O(n^3 3^n)$ time.

Novel DP Algorithm for Computing Posteriors of All Adjacency Features

The adjacency feature includes both direction of directed edge features. Therefore we can compute $P(f_{u \leftrightarrow v}, D)$ by simply adding joint probabilities for both directions. ($P(f_{u \leftrightarrow v}, D) = P(f_{u \leftarrow v}, D) + P(f_{u \to v}, D)$) Since the computational complexity for all adjacency features is same as that of all directed edge features, we can compute all $\frac{n(n-1)}{2}$ adjacency features in $O(n3^n)$ time.

Novel DP Algorithm for Computing Posteriors of All V-Structure Features

Now we describe a novel DP algorithm to compute posteriors for all $n\binom{n-1}{2}$ v-structure features. A directed edge feature can be represented by one edge feature, while a v-structure feature requires two non-adjacent parent nodes and their common child. Therefore the computation of $P(f, D)$ needs to be modified accordingly.

Among three terms in the equation (11), since $K_v(U)$ term for v-structure feature computation is same as a directed edge feature computation, we need to modify only $A_v(U)$ and $H(U)$ terms from the equation (11).

For the computation of posteriors of v-structure, in the equation (6) of definition $A_v(U)$, only the indicator function $f_v(Pa_v)$ needs to be modified from the equation (6). For example, if the feature f represents the v-structure $a \to v \leftarrow b$, since v-structure requires two parents a and b pointing to node v, the indicator function $f_v(Pa_v)$ will have 1, if $a \in Pa_v$, and $b \in Pa_v$, or, 0 otherwise.

Now $H(U)$ needs to be modified for the posterior computation of v-structures. $H(U)$ represents the summation over the parents (U) of v. Let the v-structure feature we are testing be $a \to v \leftarrow b$. Since this v-structure requires that two

[1] The proof of Eq. (7) and (9) can be found in (Tian and He 2009).

parents a and b should not be adjacent, DAGs with edges between a and b should be excluded from the computation of $H(U)$. For $(n-2)$ v-structures with fixed two parents a and b, there are significant overlaps in the posterior computation. If we compute the posterior probabilities of v-structures with same two parents nodes at once, we can exploit these overlaps to reduce the time and space complexity.

To this end, for all pairs of (a,b), and any S, where $a \in V$, $b \in V$ and $S \subset V$, we define $HV(S,a,b)$ as follows:

$$HV(S,a,b) = \sum_{k=1}^{|S|} (-1)^{k+1} \sum_{T \subseteq S, |T|=k} HV(S-T,a,b) \prod_{j \in T} AT_j(S-T,a,b), \tag{12}$$

where $AT_j(S-T,a,b)$ is (1)$A_j(S-T-\{a,b\})$ when $j=a$ or $j=b$, (2) otherwise $A_j(S-T)$. Equation (12) computes the summation over all DAGs over S except the DAGs which contain $a \rightarrow b$ or $a \leftarrow b$.

Then to apply our two modifications to equation (11), $H(U)$ needs to be replaced by $HV(U,a,b)$ for the v-structure $a \rightarrow v \leftarrow b$. Now we can express $P(f,D)$ for v-structure feature $a \rightarrow v \leftarrow b$ as follows:

$$P(f_{a \rightarrow v \leftarrow b}, D) = \sum_{U \subseteq V-\{v\}} A_v(U) HV(U,a,b) K_v(U). \tag{13}$$

where a and b are the two non-adjacent parents for v-structure feature f that we are testing. Now we give the complexity analysis of $P(f,D)$ computation for all $n\binom{n-1}{2}$ v-structure features. To fully exploit the computation overlaps in the all v-structure posterior computation, for fixed two parents a and b, and all v_i, where $v_i \in V$, $v_i \neq a$ and $v_i \neq b$, $a,b \in V$, all $(n-2)$ posteriors $(P(f_{a \rightarrow v_i \leftarrow b}, D))$ for all v_i need to be computed at once. Once we compute $HV(S,a,b)$ for all $S \subset V$ with fixed two parents a and b, we can reuse those $HV(S,a,b)$ for the computation of posteriors of all v-structure with parents a and b. This precomputation takes $\sum_{k=1}^{|V|} \binom{|V|}{k} = 2^{|V|}$ in space. For a single pair of (a,b), where $a,b \in U$, and for all $U \subset V$, $HV(U,a,b)$ can be computed in $O(n3^{n-1})$ time. Therefore, all $\frac{n(n-1)}{2}$ pairs of (a,b) can be computed in $O(n^3 3^{n-1})$ time. The computational complexity of $A_v(U)$ of v-structure is same as that of the directed edge case. So for $n\binom{n-1}{2}$ v-structure features, $A_v(U)$ computation can be done in $O(kn^3 3^n)$ time. Lastly computation complexity for $K_v(U)$ computation is same as that of directed edge case, which is $O(n3^n)$ time. Therefore we can compute $P(f,D)$ for all $n\binom{n-1}{2}$ v-structures in $O(n^3 3^n)$ time.

Experiments

In this section we empirically compare the effectiveness of the three approaches (Inference with adjacencies + v-structures, inference with only directed edges, and the maximum likelihood approach) for discovering causal graphical features. In particular, we focus on determining which of the three approaches allows us to choose an appropriate threshold for the existence of features, which is the key step for the discovery procedure. For this comparison, we have applied these three approaches to a synthetic data set generated in the following way.

The continuous-valued synthetic data sets were generated under the assumption that the generative causal model contains linear functions while unobserved variables are Gaussian. The samples of each variable v_i which has k parents were thus generated using the following equation.

$$v_i = \frac{\sum_{j=1}^{k} \alpha_j Pa_j(v_i) + N(0,1)}{\sqrt{\sum_{j=1}^{k} \alpha_j^2 + 2\sum_{a<b} \alpha_a \alpha_b cov(Pa_a(v_i), Pa_b(v_i)) + 1}}, \tag{14}$$

where $Pa_j(v_i)$ is the j^{th} parent of v_i, α_j is the causal coefficient of j^{th} parent to the v_i, and $N(0,1)$ is the Gaussian noise with 0 mean and 1 standard deviation. We assume that these unobserved Gaussian noise variables are independent of each other. After adding all the effects from parents, we divide this variable by $\sqrt{\sum_{j=1}^{k} \alpha_j^2 + 2\sum_{a<b} \alpha_a \alpha_b cov(Pa_a(v_i), Pa_b(v_i)) + 1}$ to normalize the variance of each variable. This synthetic data set does not contain any missing values. For this continuous-valued synthetic data set, we model it as linear model with Gaussian noise and each parameter (α_j) is estimated by maximum likelihood fashion. We give the following as the structural prior for our model.

$$\pi(G) = \frac{1}{N^{df}}, \tag{15}$$

where N is the number of samples and df is the degree of freedom of given model G. The above structural prior has Markov equivalent and modularity property which meets our assumption, since our algorithm computes posterior probabilities by summing over all possible DAG structures, and Eq. (15) only depends on the number of samples and the degree of freedom, which amounts to the number of edges in G.

Now we want to compare three different prediction approaches for the task of identifying graphical features. As we mentioned above, our hypothesis is that just predicting directed edge features under limited sample makes it difficult to determine a single appropriate threshold for existence of graphical features. To verify this hypothesis, we evaluated these approaches using data sets with various sample sizes. We randomly generated 100 causal diagrams with 5 variables. For each causal graphical model, we simulated samples under linear model with Gaussian noise as we explained above. Three different data sets were generated, containing 50, 100, and 200 samples.

We computed the posterior probabilities for all possible graphical features by three different approaches. For the directed edge approach, posterior probabilities for all $n(n-1)$ directed edge features were computed. For adjacency + v-structure approach, posterior probabilities for all $\frac{n(n-1)}{2}$ adjacency features and all $n\binom{n-1}{2}$ v-structure features were computed. For the ML-approach, we chose one model which had highest likelihood score among all the possible models. If the feature exists in the maximum likelihood model, its posterior is 1, otherwise 0.

Figure (2) shows the posterior probability distribution of positive features (features which exist in the data generating model) and negative features (features which don't exist

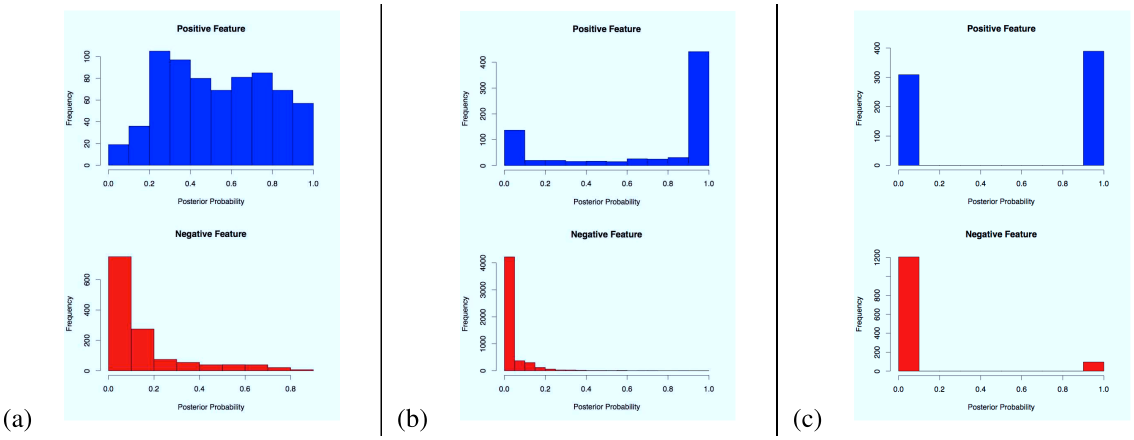

Figure 2: (a) Directed edge approach (b) Adjacency + v-structure approach (c) Maximum likelihood approach

in the data generating model) with three approaches when 200 samples are generated. Due to space limitation, we only show the 200 sample case, but we observed similar results in the 50 and 100 sample cases. Figure (2) (a) represents the posterior distribution when applying the directed edge only approach. As can be seen in the upper and lower plots of Fig. 1 (a), there is no clean separation for posterior probabilities of edge features, which makes it difficult to choose a threshold. Figure (2) (b) represents posterior distribution, when adjacency + v-structure approach applied. As Figure (2) (b) shows, the posterior of positive features are mostly greater than 0.9 and the posterior of negative features are mostly less than 0.5. We can clearly see that adjacency + v-structure approach has better distinguishing power than other approaches.

Conclusion

In this paper we propose a more effective way of identifying the Markov equivalence class of the data generating model from observational data by computing posteriors of graphical features. The previous approach of computing posteriors of only directed edge features has the problem of deciding a single appropriate threshold due to the scale difference between directed edges forming v-structures and directed edges not forming v-structures. We claim that computing posteriors of both adjacencies and v-structures is necessary and more effective for discovering graphical features, since it allows us to find a single appropriate decision threshold for the existence of the features that we are testing. Empirical validation supports that adjacency + v-structure approach is more effective than traditional directed edge only approach. For the efficient computation, we provide a novel dynamic programming algorithm which computes all $\frac{n(n-1)}{2}$ adjacency and all $n\binom{n-1}{2}$ v-structure features in $O(n^3 3^n)$ time based on DP suggested in (Tian and He 2009).

Acknowledgments

The authors would like to thank Buhm Han and the anonymous reviewers for helpful advice. E.Y.K and E.E. are supported by National Science Foundation grants 0513612, 0731455, 0729049 and 0916676, and NIH grants K25-HL080079, U01-DA024417 and N01-ES-45530.

References

Eaton, D., and Murphy, K. 2007. Bayesian structure learning using dynamic programming and MCMC. In *Uncertainty in Artificial Intelligence (UAI)*.

Ellis, B., and Wong, W. H. 2008. Learning causal Bayesian network structures from experimental data. *Journal of the American Statistical Association* 103:778–789.

Friedman, N., and Koller, D. 2003. Being Bayesian about network structure. In *Machine Learning*, 95–125.

Geiger, D., and Heckerman, D. 1994. Learning Gaussian networks. Technical Report MSR-TR-94-10, Redmond, WA.

Koivisto, M., and Sood, K. 2004. Exact Bayesian structure discovery in Bayesian networks. In *Journal of Machine Learning Research*, 594–573.

Koivisto, M. 2006. Advances in exact Bayesian structure discovery in Bayesian networks. In *Uncertainty in Artificial Intelligence (UAI)*.

Madigan, D., and York, J. 1995. Bayesian graphical models for discrete data. In *International Statistical Review*, 215–232.

Pearl, J., and Verma, T. S. 1991. A theory of inferred causation. In *Principles of Knowledge Representation and Reasoning: Proceedings of the Second International Conference*, 441–452.

Pearl, J. 1988. *Probabilistic Reasoning in Intelligent Systems*. Morgan and Kaufmann, San Mateo.

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

Spirtes, P.; Glymour, C.; and Scheines, R. 1993. *Causation, Prediction, and Search*. Springer Verlag, New York.

Tian, J., and He, R. 2009. Computing posterior probabilities of structural features in Bayesian networks. In *Uncertainty in Artificial Intelligence (UAI)*.

Verma, T. S., and Pearl, J. 1990. Equivalence and synthesis of causal models. Technical Report R-150, Department of Computer Science, University of California, Los Angeles.