Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence (IJCAI 2015)

Differential Semantics of Intervention in Bayesian Networks

Biao Qin
School of Information, Renmin University of China
Beijing, China
qinbiao@ruc.edu.cn

Abstract

Differentiation is an important inference method in Bayesian networks and intervention is a basic notion in causal Bayesian networks. In this paper, we reveal the connection between differentiation and intervention in Bayesian networks. We first encode an intervention as changing a conditional probabilistic table into a partial intervention table. We next introduce a jointree algorithm to compute the full atomic interventions of all nodes with respect to evidence in a Bayesian network. We further discover that an intervention has differential semantics if the intervention variables can reach the evidence in Bayesian networks and the output of the state-of-the-art algorithm is not the differentiation but the intervention of a Bayesian network if the differential nodes cannot reach any one of the evidence nodes. Finally, we present experimental results to demonstrate the efficiency of our algorithm to infer the causal effect in Bayesian networks.

1 Introduction

Bayesian networks capture uncertain knowledge naturally and efficiently, and belong to the family of probabilistic graphical models. They are widely used in various fields such as signal processing, information retrieval, sensornets and pervasive computing, natural language processing, and computer vision.

Causal Bayesian networks are another kind of directed acyclic graph, which conveys causal information as well as the traditional conditional independencies, and permits one to infer the causal effects. In causal Bayesian networks, *interventions* are usually interpreted as an external agent setting a variable to a certain value, which contrasts with an agent just passively observing variables' values. Goldszmidt and Pearl [Goldszmidt and Pearl, 1992] denoted the intervention of a variable in a causal Bayesian network as $do(V_i = v_i^k)$, or $do(v_i^k)$ for short.

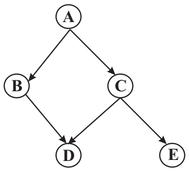
In this paper, we will use capital letters (e.g. A) for network parameters, lowercase letters (a) for any values taken by the corresponding network parameters, and lowercase letters with superscripts (a^k) as generic symbols for specific values taken by the corresponding network parameters. For ease of illustration, we also denote $|V_i|$ as the number of possible values for a network parameter V_i. Thus V_i has $|V_i|$ atomic interventions $do(v_i^k)$ $(k = 1, \ldots, |V_i|)$. Each such intervention $do(v_i^k)$ is called an *action*.

Pearl [Pearl, 2009b] introduced two methods to represent the intervention in causal Bayesian networks. One is a mutilated graph, which is obtained by deleting all links from its parent nodes to the intervention node V_i (Figure 2(a)). The other is an augmented network, which is obtained by adding a hypothetical intervention link $F_i \rightarrow V_i$ in the network(Figure 2(b)). Using the mutilated graph, for the Bayesian network shown in Figure 1, $P(d|do(B = 1))$ can be computed as follows.

$$\begin{aligned} P(d|do(B=1)) &= \sum_{ace} P(a)P(c|a)P(d|B=1,c)P(e|c) \\ &= \sum_{ac} P(a)P(c|a)P(d|B=1,c)\sum_{e} P(e|c) \\ &= \sum_{a} P(a)\sum_{c} P(c|a)P(d|B=1,c) \qquad (1) \end{aligned}$$

The *prediction* $P(d|B = 1)$ can be computed by using classical methods like variable elimination. A Bayesian network consists of two parts, the network structure and the conditional probability tables (CPTs). Pearl changed the network structure to represent the external intervention. In this paper, we encode the external intervention from another perspective by replacing CPTs with *partial intervention tables* without changing the network structure. For an action $do(v_i^k)$, we set $P(v_i^k|\mathbf{u}) = 1$ and $P(v_i^l|\mathbf{u}) = 0$ for $l \neq k$, and the CPT of V_i becomes a partial intervention table. We will give formal definition for this concept in Section 3.

Pearl's method computes V_i's atomic interventions $do(v_i^k)$ $(k = 1, \cdots, |V_i|)$ by inferring the Bayesian network $|V_i|$ times. Using the partial intervention table, we can compute all these atomic interventions by inferring the Bayesian network only once. Assume that all variables in the Bayesian network shown in Figure 1 are binary. If one intervenes and assigns a value to B, he has to, according to Pearl's method, use two probabilistic inferences for $do(B = 1)$ and $do(B = 0)$. However, we can employ only one Bayesian network inference to simultaneously compute both $do(B = 1)$ and $do(B = 0)$. The combination of $do(B = 1)$ and $do(B = 0)$ together is called a *full atomic intervention*, whose definition is given in Section 3. Our first main contribution is to propose a new representation model of the intervention by using



Θ_B

A	B	$P(b\|a)$
1	1	0.14
1	0	0.86
0	1	0.89
0	0	0.11

Θ_C

A	C	$P(c\|a)$
1	1	0.18
1	0	0.82
0	1	0.72
0	0	0.28

Θ_A

A	$P(a)$
1	0.34
0	0.66

Θ_D

B	C	D	$P(d\|bc)$
1	1	1	0.88
1	1	0	0.12
1	0	1	0.22
1	0	0	0.78
0	1	1	0.05
0	1	0	0.95
0	0	1	0.15
0	0	0	0.85

Θ_E

C	E	$P(e\|c)$
1	1	0.34
1	0	0.66
0	1	0.59
0	0	0.41

Figure 1: A Bayesian network with five nodes

the partial intervention tables, and design a novel and efficient approach to calculating interventions based on this new representation model.

Bayesian networks have two classical inference methods, i.e. variable elimination and jointree. Let **e** denote the evidence. We propose a jointree based algorithm to compute the full intervention for all network parameters with respect to $P(\mathbf{e})$. This is our second main contribution.

Differentiation is one of inference methods in Bayesian networks and is conducted on a multi-linear function f called network polynomial, which has the property that $f(\mathbf{e}) = P(\mathbf{e})$ for each given evidence **e** [Darwiche, 2003]. The calculation of the differentiation of f is similar to how we obtain partial derivatives in classical mathematics. Darwiche [Darwiche, 2003] proposes an efficient framework for computing the first derivatives with respect to all network parameters. Moreover, the partial derivative has applications to sensitivity analysis [Chan and Darwiche, 2002; 2004], by identifying the minimal changes of parameters in order to satisfy probabilistic query constraints. It can be also applied in the learning of network parameters from data. The details will be reviewed in Section 2.

The first derivative $\frac{\partial P(d)}{\partial P(b|a)}$ can be computed over the Bayesian network in Figure 1 as follows.

$$\begin{aligned}\frac{\partial P(d)}{\partial P(b|a)} &= \sum_{ce} P(a)P(c|a)P(d|b,c)P(e|c)\\ &= P(a)\sum_{c} P(c|a)P(d|b,c)\sum_{e} P(e|c)\\ &= P(a)\sum_{c} P(c|a)P(d|b,c) \qquad (2)\end{aligned}$$

We find Eq. (1) can be derived from Eq. (2). Thus action is computed from the derivative function while prediction is inferred from the original function, and action and prediction can be regarded as velocity and displacement, respectively. Usually, velocity is more stable than displacement. Similarly, Pearl [Pearl, 2009b] found causal relationship is more stable than probabilistic relationship. However, if the intervention variables cannot reach the evidence variables, the distribution of evidence variables is independent of the intervention variables. We find the causal influence cannot be derived from the differentiation in this case. For example, $\frac{\partial P(c)}{\partial P(b|a)}$ can be computed by using Park and Darwiche's method [Park and Darwiche, 2004] as follows.

$$\begin{aligned}\frac{\partial P(c)}{\partial P(b|a)} &= \sum_{d,e} P(a)P(c|a)P(d|b,c)P(e|c)\\ &= P(a)P(c|a)\sum_{d} P(d|b,c)\sum_{e} P(e|c)\\ &= P(a)P(c|a) \qquad (3)\end{aligned}$$

However, we observe that $\frac{\partial P(c)}{\partial P(b|a)} = 0$ while $P(c|do(B = 1)) = \sum_a P(a)P(c|a)$. So we show that the output of Park and Darwiche's jointree algorithm is not the differentiation but the intervention of Bayesian networks in this case. Our third contribution is to reveal the connection between the intervention and the differentiation.

The rest of this paper is organized as follows. We outline Bayesian networks, their differential semantics and interventions in Section 2. We introduce a new approach to represent intervention in Section 3 and discover its differential semantics in Section 4. Experimental results are reported in Section 5. The related work is discussed in Section 6, and we conclude our work in Section 7.

2 Preliminaries

In this section, we first outline the differentiation of Bayesian networks and next briefly survey the intervention of Bayesian networks.

2.1 The differentiation of Bayesian networks

A Bayesian network is defined as a pair $BN = (G, \Theta)$. Here, $G = (\mathbf{V}, \mathbf{E})$ is a directed acyclic graph, called the network structure, where $\mathbf{V}$ denotes all nodes in G and an edge in $\mathbf{E}$ denotes conditional dependence relationships between two nodes. Θ is a set of CPTs, and one CPT is for each node in G, called the network parameter. Moreover, a Bayesian network implicitly represents the joint probability distribution of its set of variables $\mathbf{V}$. In Bayesian networks, every variable is conditionally independent of its nondescendants given its parents. Using the conditional independence property, the joint probability distribution of a Bayesian network can be simplified. For example, Figure 1 shows a Bayesian network with five nodes $\mathbf{V} = \{A, B, C, D, E\}$. $P(v)$ can be computed as shown below.

$$\begin{aligned}P(v) &= \prod P(v_i|\mathbf{u}_i)\\ &= P(a)P(b|a)P(c|a)P(d|b,c)P(e|c) \qquad (4)\end{aligned}$$

Here, $\mathbf{u}_i$ is the parent of v_i. Darwiche [Darwiche, 2003] defined for each Bayesian network a unique multilinear function, denoted by f, over a set of evidence indicators λ_v and a set of parameters $P(v|\mathbf{u})$. Each term of f corresponds to an instantiation of the network variables. Therefore, f has an exponential number of terms. The term corresponding to instantiation **v** is the product of all evidence indicators and network parameters that are compatible with the instantiation. The network polynomial of a BN is defined as follows:

$$f = \sum_{\mathbf{v}} \prod_{v\mathbf{u} \sim \mathbf{v}} P(v|\mathbf{u})\lambda_v$$

where $\sim$ denotes the compatibility relation among instantiations (that is, $v\mathbf{u} \sim \mathbf{v}$ says that instantiations $v\mathbf{u}$ and **v** agree on values of their common variables). The value of network polynomial f at evidence **e**, denoted by $f(\mathbf{e})$, is the result of replacing each evidence indicator λ_v in f with 1 if v is consistent with **e**, and with 0 otherwise. In order to differentiate Bayesian networks, Darwiche [Darwiche, 2003] compiled the network polynomial into an arithmetic circuit. The circuit is a rooted, directed acyclic graph with two types of leaf nodes, λ_v and $P(v|\mathbf{u})$, and its other nodes are multiplication and addition operations. Parameter variables are set according to the network CPTs while indicator variables are set according to the given evidence. Once the circuit inputs are set, it can be evaluated using a bottom-up pass, which proceeds from the circuit inputs to its output. Moreover, it can be differentiated using a top-down pass, which computes the first derivative of the circuit output with respect to every circuit input. Since $f(\mathbf{e}) = P(\mathbf{e})$, Darwiche proved that the first derivatives with respect to network parameters can be calculated below.

$$\frac{\partial P(\mathbf{e})}{\partial P(v|\mathbf{u})} = \frac{P(v, \mathbf{u}, \mathbf{e})}{P(v|\mathbf{u})}, when\ P(v|\mathbf{u}) \neq 0.$$

A jointree [Shenoy and Shafer, 1986; Jensen and Andersen, 1990] for BN is a pair $(\mathcal{T}, C)$, where $\mathcal{T}$ is a tree and C is a function that maps each node i in the tree $\mathcal{T}$ into a label C_i, which is called a *cluster*. A jointree must satisfy three properties: (1) the cluster C_i is a set of nodes from the BN; (2) each family in the BN must appear in some cluster C_j; (3) if a node appears in two clusters C_i and C_j, it must also appear in every cluster C_k on the path between C_i and C_j. The edges of a jointree are called *separators*, which is denoted by S_{ij} and defined as $C_i \cap C_j$. The *width* of a jointree is defined as the size of its largest cluster minus one. Given some evidence **e**, a jointree algorithm [Pearl, 1988; Shenoy and Shafer, 1986; Jensen and Andersen, 1990] propagates messages between clusters. After passing an inward and an outward messages for each cluster C_i, one can compute its marginal $P(C_i, \mathbf{e})$.

Park and Darwiche [Park and Darwiche, 2004] further proposed a differential semantics for jointree algorithms and computed the first derivative with respect to $P(v|\mathbf{u})$ in the jointree. A jointree can implicitly encode an arithmetic circuit. Inward message propagation evaluates the circuit while outward message propagation differentiates it. Suppose that CPT $\Theta_{X_j|\mathbf{U}_j}$ is assigned to cluster C_i which has variables C_i. Then:

$$\frac{\partial P(\mathbf{e})}{\partial P(v_j|\mathbf{u}_j)} = \left[\sum_{C_i \setminus v_j \mathbf{u}_j} \prod_{j} M_{ji} \prod_{k \neq j} P(v_k|\mathbf{u}_k)\right](v_j\mathbf{u}_j) \quad (5)$$

where $P(v_k|\mathbf{u}_k)$ ranges over all evidences and CPTs assigned to cluster C_i and M_{ji} denotes the message from C_j to C_i.

2.2 The intervention of Bayesian networks

The simplest type of external intervention is one in which a single variable, say V_i, is forced to take on some fixed value v_i^k. Such an intervention is called atomic intervention. Pearl [Pearl, 2009b] introduced two techniques to denote the intervention in a Bayesian network. One is to describe an intervention by a mutilated graph derived from the original Bayesian network by removing all the incoming arcs to V_i and setting the CPT of V_i as $P(V_i = v_i^k) = 1$. In this case, intervention essentially separates V_i from its direct causes and V_i becomes a root node. For example, if we want to compute the intervention $do(B = 1)$ in the Bayesian networks shown in Figure 1, we should first delete the link $A \rightarrow B$ and then assign $P(B = 1) = 1$. The graph resulting from this operation is shown in Figure 2(a), and the resulting joint distribution on the remaining variables will be

$$P(a, c, d, e|do(1)) = P(a)P(c|a)P(d|c, B = 1)P(e|c)$$

Using the mutilated graph, Pearl [Pearl, 2009a] revealed that the distribution generated by an intervention $do(V_i = v_i^k)$ on a set **V** is given by the truncated factorization

$$P(v_1, \ldots, v_{i-1}, v_{i+1} \ldots, v_n|do(v_i^k)) = \prod_{j \neq i} P(v_j|\mathbf{u}_j)$$

where $P(v_j|\mathbf{u}_j)$ are the pre-intervention conditional probabilities. Thus the distribution of evidence **e** given the intervention can be computed by

$$P(\mathbf{e}|do(v_i^k)) = \sum_{\mathbf{v} \setminus \mathbf{e}} \prod_{j \neq i} P(v_j|\mathbf{u}_j) \quad (6)$$

An alternative way is to encode an intervention by an augmented network [Pearl, 1993] which is generated by adding to BN a link $F_i \rightarrow V_i$. For example, if we want to compute the intervention $do(B = 1)$ in the Bayesian networks shown in Figure 1, we should add a link $F_i \rightarrow B$ in the network. The augmented network is shown in Figure 2(b), where F_i is a new variable taking values in $\{do(v_i^a),\text{idle}\}$, v_i^a can be any value of V_i and "idle" denotes no intervention. Thus, the new parent set of V_i in the augmented network is $\mathbf{U}_i^a = \mathbf{U}_i \cup \{F_i\}$, and it is related to V_i by the following conditional probability

$$P(v_i|\mathbf{u}_i^a) = \begin{cases} P(v_i|\mathbf{u}_i) & \text{if } F_i = \text{idle}, \\ 0 & \text{if } F_i = do(v_i^a) \text{ and } V_i \neq v_i^a, \\ 1 & \text{if } F_i = do(v_i^a) \text{ and } V_i = v_i^a. \end{cases}$$

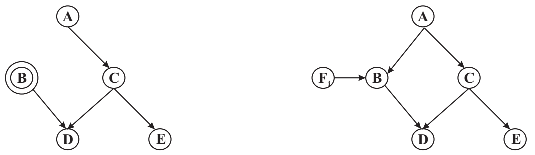
Then the causal effect of the intervention $do(v_i^a)$ can be given by

$$P(v_1, \ldots, v_n|v_i^a) = P_a(v_1, \ldots, v_n|F_i = do(v_i^a))$$

where P_a is the distribution specified by the augmented network $BN^a = BN \cup \{F_i \rightarrow V_i\}$ and $P(v_i|\mathbf{u}_i^a)$, with an arbitrary prior distribution on F_i.

3 A novel approach to computing the intervention

In this section, we introduce a new model to represent the intervention without changing the network structure, and propose a jointree based approach to calculating the intervention by using this novel representation model. We first give the following definition.



(a) a mutilated graph (b) an augmented network

Figure 2: Network representation of the action of "setting B=1"

Definition 3.1 *Given an intervention $do(v_i^k)$, the* **partial intervention table** *(PIT in short) of V_i demonstrates the* **intervention probability***, $P'(v_i^k|\boldsymbol{u}_i)$, w.r.t $do(v_i^k)$, which is calculated as below.*

$$P'(v_i^k|\boldsymbol{u}_i) = \begin{cases} 1 & \textit{if } V_i = v_i^k, \\ 0 & \textit{otherwise.} \end{cases} \tag{7}$$

For example, if we want to compute the action $do(B = 1)$ in the Bayesian networks shown in Figure 1, then $P(B = 1) = 1$ and $P(B = 0) = 0$. Thus we can change Θ_B to $\Theta'_B(B = 1)$ as shown below.

$\Theta'_B(B = 1)$

A	B	$P'(b\|a)$
1	1	1
1	0	0
0	1	1
0	0	0

A *PIT representation* of an intervention is obtained by simply replacing the CPT of each intervention variable by its corresponding PIT without any changes to the network structure.

Theorem 3.1 *The PIT representation is correct in terms of calculating the intervention.*

Proof. We prove this by showing that the causal effect, $P(\mathbf{e}|do(v_i^k))$, will be the same when calculated by using the CPTs of the mutilated-graph representation and the PITs of the PIT representation respectively. Using Pearl's mutilated graph, we get the following causal effect.

$$P_1(\mathbf{e}|do(v_i^k)) = \sum_{\mathbf{v}\backslash\mathbf{e}} \prod_{j \neq i} P(v_j|\mathbf{u}_j)$$

Using our partial intervention table, we compute the following causal effect.

$$P_2(\mathbf{e}|do(v_i^k)) = \sum_{\mathbf{v}\backslash\mathbf{e}} P'(v_i|\mathbf{u}) \prod_{j \neq i} P(v_j|\mathbf{u}_j)$$

Since $P'(v_i|\mathbf{u}) = 1$, $P_1(\mathbf{e}|do(v_i^k)) = P_2(\mathbf{e}|do(v_i^k))$. Thus the theorem is proved. □

For example, if one intervenes and sets $B = 1$, then we can derive by using $P'(B = 1|a) = 1$ as follows.

$$\begin{aligned} \Phi &= P(a, c, d, e|do(B = 1)) \\ &= P(a)P(c|a)P'(B = 1|a)P(d|c, B = 1)P(e|c) \\ &= P(a)P(c|a)P(d|c, B = 1)P(e|c) \end{aligned}$$

So we can get the same result as Pearl's method. Next, we give the following definition for the full atomic intervention.

Definition 3.2 *A* **full atomic intervention** *of V_i consists of all atomic interventions of V_i each of which is w.r.t a possible value of V_i. It is denoted as $do(V_i)$ and contains $do(v_i^1), \ldots, do(v_i^{|V_i|})$.*

For example, if one wants to compute the interventions $do(B = 1)$ and $do(B = 0)$ in Figure 1, he has to compute $P(a, c, d, e|do(B = 1))$ and $P(a, c, d, e|do(B = 0))$ one by one according to Pearl's method. However, we can simultaneously compute $do(B = 1)$ and $do(B = 0)$ using the following technique. First, since $P(B = 1) = 1$ and $P(B = 0) = 1$, we change Θ_B to a full intervention table $\Theta'_B(B)$ as shown below.

$\Theta'_B(B)$

A	B	$P'(b\|a)$
1	1	1
1	0	1
0	1	1
0	0	1

We next compute $P(a, c, d, e|do(B))$ as follows.

$$P(a, c, d, e|do(B)) = P(a)P(c|a)P'(b|a)P(d|c, b)P(e|c)$$

Since we compute the post-intervention distribution of a full atomic intervention, we should make a distinction between $do(B = 1)$ and $do(B = 0)$. Thus we should keep the intervention variable and its parent variables in the result. Let $Vars(\Theta_{V_i})$ denote all network variables in Θ_{V_i}. Since $\{A, B\} \subset Vars(\Theta_A \cup \Theta_C \cup \Theta_D \cup \Theta_E)$ and $P'(b|a) = 1$, it can yield

$$P(a, c, d, e|do(B)) = P(a)P(c|a)P(d|c, b)P(e|c) \tag{8}$$

Using the PIT presentation, we can compute each full atomic intervention by one Bayesian network inference, e.g, Eq. (8). While using Pearl's methods, a full atomic intervention for node V_i requires $|V_i|$ Bayesian network inferences.

The jointree algorithm is widely utilized for performing Bayesian network inferences. The following theorem illustrates a method to compute the full atomic intervention by using the jointree algorithm.

Theorem 3.2 *In a jointree constructed over G, the full atomic intervention of every node V_i with respect to $P(\boldsymbol{e})$ in cluster C_l can be computed below.*

$$P(\boldsymbol{e}|do(V_i)) = \begin{cases} \Phi_1 & \textit{if } V_i \textit{ can reach } \boldsymbol{e}, \\ \Phi_2 & \textit{else.} \end{cases}$$

Here, $\Phi_1 = \sum_{\{C_l, \boldsymbol{e}\}\backslash\{v_i, \boldsymbol{e}\}} P'(v_i|\boldsymbol{u}_i) \prod_j M_{jl} \prod_{k \neq i} P(v_k|\boldsymbol{u}_k)$, and $\Phi_2 = \sum_{\{C_l, \boldsymbol{e}\}\backslash\boldsymbol{e}} P'(v_i|\boldsymbol{u}_i) \prod_j M_{jl} \prod_{k \neq i} P(v_k|\boldsymbol{u}_k)$.

Proof. 1. If node V_i can reach $\mathbf{e}$, we can use the jointree algorithm to compute the full intervention as follows.

$$\begin{aligned} P(\mathbf{e}|do(V_i)) &= \sum_{\mathbf{v}\backslash\{v_i, \mathbf{e}\}} \prod_{j \neq i} P(v_j|\mathbf{u}_j) \\ &= \sum_{\mathbf{v}\backslash\{v_i, \mathbf{e}\}} P'(x_i|\mathbf{u}_i) \prod_{j \neq i} P(v_j|\mathbf{u}_j) \\ &= \sum_{C'_l\backslash\{v_i, \mathbf{e}\}} P'(x_i|\mathbf{u}_i) \prod_j M_{jl} \prod_{k \neq i} P(v_k|\mathbf{u}_k) \end{aligned}$$

Here, C'_l denotes $\{C_l, \mathbf{e}\}$.

2. If node V_i cannot reach $\mathbf{e}$, we can use the jointree algorithm to compute the full intervention as shown below.

$$\begin{aligned} P(\mathbf{e}|do(V_i)) &= P(\mathbf{e}) \\ &= \sum_{\mathbf{v}\backslash\mathbf{e}} P'(x_i|\mathbf{u}_i)\prod_{j\neq i} P(v_j|\mathbf{u}_j) \\ &= \sum_{\{C_l,\mathbf{e}\}\backslash\mathbf{e}} P'(x_i|\mathbf{u}_i)\prod_{j} M_{jl}\prod_{k\neq i} P(v_k|\mathbf{u}_k) \end{aligned}$$

The proof is then done. □

Using Theorem 3.2, we propose Algorithm 1 to compute the full atomic intervention of all nodes w.r.t evidence in Bayesian networks. We now explain the main aspects of the algorithm. We first compile a jointree by the network structure and $P(\mathbf{e})$. Then the jointree passes messages toward its root. According to Theorem 3.2, we can use a cached factor Φ_i to compute the full atomic interventions of all nodes in cluster C_i with respect to $P(\mathbf{e})$ in some order π_i if a cluster C_i has received messages from all its neighbors in the outward phase. After computing the full atomic intervention of a node V_j, our algorithm updates Φ_i. In this way, we need not compute the full atomic intervention of its next node from scratch but from the cached factor each time. All clusters can compute the full atomic interventions in parallel to accelerate the process.

Algorithm 1 FullDo

```
Input: e: a set of evidences
Output: the full atomic intervention of all network variables
compile the jointree;
messages are passed toward the root;
messages are passed away from the root;
for (each cluster $C_i$) do
    receive messages from its all neighbors;
    select an intervention order $\pi_i$ in the cluster;
    for (each node $V_j \in \pi_i$) do
        if ($V_j$ is the first node) $\Phi_i = \emptyset$;
        use Theorem 3.2 and $\Phi_i$ to compute $P(\mathbf{e}|do(V_j))$;
        update $\Phi_i$;
    end for;
end for;
return (the full atomic interventions of all variables);
```

The time complexity of our algorithm for computing the full intervention in Bayesian networks is $O(n \cdot exp(w) + n_1^2 \cdot l \cdot exp(w_1))$, where n is the number of network variables, w is the network treewidth, n_1 is the number of network variables in the biggest cluster, l is the number of clusters for the jointree and w_1 is the biggest network treewidth of all clusters. Using Pearl's Eq. (6), we implement a jointree algorithm called Do to compute the causal effect of the full atomic interventions of all nodes in a Bayesian network. The time complexity of Do is $O(n \cdot exp(w) + m \cdot n_1^2 \cdot l \cdot exp(w_1))$, where m is the maximum value of $|V_i|$ $(i = 1, \cdots, |\mathbf{V}|)$.

4 The differential semantics of intervention

We recompute $\frac{\partial P(c)}{\partial P(b|a)}$ shown in Eq. (3). We call node B as *differential node* and first compute $P(c)$ as shown below.

$$\begin{aligned} P(c) &= \sum_{a,b,d,e} P(a)P(b|a)P(c|a)P(d|b,c)P(e|c) \\ &= \sum_{a} P(a)P(c|a)\sum_{b} P(b|a)\sum_{d} P(d|b,c) \\ &\quad \sum_{e} P(e|c) \\ &= \sum_{a} P(a)P(c|a) \end{aligned}$$

This is because first $\sum_e P(e|c) = 1$, next $\sum_d P(d|b,c) = 1$ and finally $\sum_b P(b|a) = 1$. Since $P(c) = \sum_a P(a)P(c|a)$, $\frac{\partial P(c)}{\partial P(b|a)} = \frac{\partial \sum_a P(a)P(c|a)}{\partial P(b|a)} = 0$. Thus $P(e|c), P(b|a)$ and $P(d|b,c)$ have no contribution to $\frac{\partial P(c)}{\partial P(b|a)}$. We observe that nodes E, B and D cannot reach node C. From this finding, we give the following theorem.

Theorem 4.1 *If a differential node V_i cannot reach any one of the evidence nodes $\boldsymbol{e}$, then $\frac{\partial P(\boldsymbol{e})}{\partial P(v_i|\boldsymbol{u}_i)} = 0$.*

Proof. Shachter [Shachter, 1986] had proved that any node which cannot reach any one of the evidence nodes $\mathbf{e}$ has no contribution to the marginal probability in a Bayesian network. Thus $\frac{\partial P(\mathbf{e})}{\partial P(v_i|\mathbf{u}_i)} = 0$. □

So the outputs of the jointree algorithm [Park and Darwiche, 2004] is not differentiation of Bayesian networks in this case. We then reveal the differential semantics of intervention in Bayesian networks and begin with the differentiation of $P(\mathbf{e})$ with respect to a differential node V_i.

Lemma 4.1 *If V_i can reach $\boldsymbol{e}$ in a Bayesian network, we can compute the first derivative of $P(\boldsymbol{e})$ with respect to $P(v_i|\boldsymbol{u}_i)$ using the following formula:*

$$\frac{\partial P(\boldsymbol{e})}{\partial P(v_i|\boldsymbol{u}_i)} = \sum_{\boldsymbol{v}\backslash\{v_i,\boldsymbol{u}_i,\boldsymbol{e}\}} P'(v_i|\boldsymbol{u}_i)\prod_{j\neq i} P(v_j|\boldsymbol{u}_j)$$

Proof. We prove the theorem as follows.

$$\begin{aligned} \frac{\partial P(\mathbf{e})}{\partial P(v_i|\mathbf{u}_i)} &= \frac{\partial \sum_{\mathbf{v}\backslash\mathbf{e}} \prod P(v_j|\mathbf{u}_j)}{\partial P(v_i|\mathbf{u}_i)} \\ &= \frac{\partial P(v_i|\mathbf{u}_i) \sum_{\mathbf{v}\backslash\mathbf{e}} P'(v_i|\mathbf{u}_i)\prod_{j\neq i} P(v_j|\mathbf{u}_j)}{\partial P(v_i|\mathbf{u}_i)} \\ &= \sum_{\mathbf{v}\backslash\{v_i,\mathbf{u}_i,\mathbf{e}\}} P'(v_i|\mathbf{u}_i)\prod_{j\neq i} P(v_j|\mathbf{u}_j) \end{aligned}$$

Hence the lemma is proved. □

We reveal the following connection between the first derivative and atomic intervention of Bayesian networks.

Theorem 4.2 *In Bayesian networks, a full intervention has the following differential semantics.*

$$P(\boldsymbol{e}|do(V_i)) = \begin{cases} \sum_{\boldsymbol{u}_i} \frac{\partial P(\boldsymbol{e})}{\partial P(v_i|\boldsymbol{u}_i)} & \textit{if } V_i \textit{ can reach } \boldsymbol{e}, \\ P(\boldsymbol{e}) & \textit{else.} \end{cases}$$

Proof. 1. From Lemma 4.1, we know if V_i can reach $\mathbf{e}$, $\frac{\partial P(\mathbf{e})}{\partial P(v_i|\mathbf{u}_i)} = \sum_{\mathbf{v}\setminus\{v_i,\mathbf{u}_i,\mathbf{e}\}} P'(v_i|\mathbf{u}_i) \prod_{j\neq i} P(v_j|\mathbf{u}_j)$

Next we can derive $P(\mathbf{e}|do(V_i))$ as follows.

$$\begin{aligned} P(\mathbf{e}|do(V_i)) &= \sum_{\mathbf{v}\setminus\{v_i,\mathbf{e}\}} \prod_{j\neq i} P(v_j|\mathbf{u}_j) \\ &= \sum_{\mathbf{v}\setminus\{v_i,\mathbf{e}\}} P'(v_i|\mathbf{u}_i) \prod_{j\neq i} P(v_j|\mathbf{u}_j) \\ &= \sum_{\mathbf{u}_i} \frac{\partial P(\mathbf{e})}{\partial P(v_i|\mathbf{u}_i)} \end{aligned}$$

2. According to Shachter [Shachter, 1986] theorem, if node V_i which cannot reach any one of the evidence nodes, then V_i has no contribution to the marginal probability of $\mathbf{e}$ in a Bayesian network. Thus $P(\mathbf{e}|v_i) = P(\mathbf{e})$ whenever $P(v_i) > 0$. If we intervene on node V_i, we change its CPT into a PIT. Since we do not change the network structure, $P(\mathbf{e}|v_i) = P(\mathbf{e})$ whenever $P(v_i) > 0$. □

Using Theorem 4.2, we can derive the following Corollary.

Corollary 4.1 *In Bayesian networks, an atomic intervention has the following differential semantics.*

$$P(\boldsymbol{e}|do(v_i^k)) = \begin{cases} \sum_{\boldsymbol{u}_i} \frac{\partial P(\boldsymbol{e})}{\partial P(v_i|\boldsymbol{u}_i)} & \textit{if } V_i \textit{ can reach } \boldsymbol{e}, \\ P(\boldsymbol{e}) & \textit{else.} \end{cases}$$

Let $V_i = v_i^k$. Corollary 4.1 directly follows from Theorem 4.2.

So far we highlight the relationship between atomic interventions and first derivatives. Intuitively, there should be similar relationship between multiple interventions and higher-order derivatives. For example, if both B and C are intervention variables, the post-intervention distribution would be

$$P(d, e|do(B), do(C)) = \sum_a P(a)P(d|b, c)P(e|c)$$

On the other hand, the corresponding second derivative $\frac{\partial^2 P(d,e)}{\partial P(b|a)\partial P(c|a)} = P(a)P(d|b, c)P(e|c)$. We can see that the two also have similar connections. Due to space limitation, we do not include the discussions in this paper.

5 Experiments

To test our algorithm, we study its performance in real Bayesian networks. We implement our *FullDo* and *Do* based on BNJ [kdd, 2006]. The experiments were conducted on a PC with Intel core2, 1.8GHz, 4.0G memory and Linux. We compute the full atomic intervention of the real world Bayesian networks, which are selected from the benchmark [ace, 2003]. The statistics of these networks are summarized in Table 1, where $\sum |V_i|$ denotes the number of values for all network variables in a Bayesian network and $|\bar{m}|$ represents the average number of values for each network variable.

Table 1: Statistics for the Bayesian networks

Network	#Nodes	#Arcs	$\sum \lvert V_i \rvert$	$\lvert \bar{m} \rvert$
alarm	37	46	104	2.8
blockmap_05_01	700	1183	1400	2
blockmap_05_02	855	1461	1710	2
blockmap_05_03	1005	1729	2010	2
diabetes	413	602	4682	11.3
fs-04	262	388	524	2
hailfinder	56	66	223	4
mastermind_3_8_3	1220	2068	2440	2
munin1	189	282	995	5.3
munin2	1003	1244	5376	5.4
munin3	1044	1315	5603	5.4
munin4	1041	1397	5647	5.4
pathfinder	109	195	448	4.1
pigs	441	592	1323	3
Students_03_02	376	647	752	2
water	32	66	116	3.6

Since both *FullDo* and *Do* are based on the jointree algorithm, we first compile the jointree for them and use *Compile* to denote the compilation phase of *FullDo* and *Do*. We next compute the full atomic intervention of all nodes with respect to evidence variables and use *FullDo*-I and *Do*-I to represent the time for computing the full atomic intervention of all nodes in the outward phase by using our method and Pearl's method respectively. From Table 2, we can see that *FullDo* performs better than *Do* for all Bayesian networks. This is because our *FullDo* computes the full atomic intervention of each node by a single Bayesian network inference while *Do* computes each atomic intervention by one Bayesian network inference. In order to compute the full atomic intervention of each node V_i, *Do* has to employ $|V_i|$ Bayesian network inferences. Comparing $|\bar{m}|$ in Table 1 with the ratio of *FullDo*-I to *Do*-I in Table 2, we further observe that the speedup ratio is positively correlated with $|\bar{m}|$ but less than $|\bar{m}|$. The reasons are as follows. Since it can independently compute every atomic intervention of a node, *Do* computes each full atomic intervention in a parallel way. Moreover, the intermediate results of our algorithm are larger than those of *Do*.

6 Related Work

Differentiation and evaluation are two inference methods in Bayesian networks. Evaluation is widely studied in Bayesian networks. Russell et al. [Russell *et al.*, 1995] originally studied the derivatives with respect to network parameters. Castillo et al. [Castillo *et al.*, 1997] initially introduced the network polynomial to denote the Bayesian networks. After compiling the network polynomial into arithmetic circuits, Darwiche [Darwiche, 2003] performed inference in Bayesian networks by evaluating the circuits in a bottom-up manner and computed the first derivatives with respect to network parameters by differentiating the circuits in a top-down manner. Park and Darwiche [Park and Darwiche, 2004] further introduce a jointree algorithm to compute the first derivatives and studied the relationship between circuit propagation and jointree propagation. They found that an arithmetic circuit is implicitly encoded in a jointree. However, we find that the outputs of their jointree algorithm [Park and Darwiche, 2004] are not the differentiation but the intervention of the Bayesian networks if a differential node V_i cannot reach the set of evidence nodes $\mathbf{e}$. Accordingly, the output of arithmetic circuit

Table 2: The performance of *FullDo* vs. *Do* (msec)

Network	*Compile*	*FullDo*-I	*Do*-I
alarm	7.7	3.2	4.5
blockmap_05_01	495	37.8	50.1
blockmap_05_02	780	48.9	63.4
blockmap_05_03	857	51.2	66.6
diabetes	40	31.8	167.9
fs-04	18.8	2.6	3.4
hailfinder	10.5	3.1	5.8
mastermind_3_8_3	703	28.2	37.7
munin1	613	74.5	219.1
munin2	70.3	38.9	108.1
munin3	92.3	53.8	142.8
munin4	107.5	65.2	182.4
pathfinder	38	13.3	28.6
pigs	34.2	11.2	17.7
students_03_02	70.3	13.8	19.9
water	21.2	7.1	12.1

is not the differentiation but the intervention of a Bayesian network in this case

In causal networks, there are two ways to represent the intervention. One is graphs as models of intervention and the other is interventions as variables [Pearl, 1993]. The $do(x)$ was first used in Goldszmidt and Pearl [Goldszmidt and Pearl, 1992] and has become popular. Pearl [Pearl, 2009b] encoded the intervention by changing the network structure. In this paper, we study the intervention in two novel ways. One way is that we encode an atomic intervention as changing a CPT into a partial intervention table. The other way is that we reveal the connection between the differentiation and intervention. Pearl [Pearl, 2009b] discovered that statisticians read structural equations as statements about $E(Y|x)$ while economists read them as $E(Y|do(x))$. We find that statisticians directly compute the result from the observation while economists compute the result by differentiating the observation. Thus, statisticians and economists use the observation from different perspectives.

7 Conclusions

Bayesian networks are a very general tool that can be used for a large number of artificial intelligence problems: reasoning, learning, planning, perception and etc. Causal Bayesian networks can represent and respond to external or spontaneous changes. Any local reconfiguration of the mechanisms in the environment can be translated, with only minor modification, into an isomorphic reconfiguration of the network topology. In this paper, we show that action is the derivative of the observation while prediction directly computes from the observation. Thus, there is one step between causal effect and prediction just like the relationship between velocity and displacement. Moreover, there are two ways to encode intervention. One is Pearl's method to change the network structure, and the other is our technique to alter the network parameters (CPTs). As a future work, we will reveal the relationship between multiple interventions and higher-order derivative of Bayesian networks.

Acknowledgment

This work is partially funded by the National Natural Science Foundation of China under Grant No. 61170012 and the National Basic Research Program of China (973 Program) under Grant No. 2012CB316205

References

[ace, 2003] *http://reasoning.cs.ucla.edu/ace*. 2003.

[Castillo *et al.*, 1997] E. Castillo, J. Gutierrez, and A. Hadi. Sensitivity analysis in discrete bayesian networks. *IEEE Transactions on Systems, Man, and Cybernetics, Part A (Systems and Humans)*, 27(4):412–423, 1997.

[Chan and Darwiche, 2002] H. Chan and A. Darwiche. When do numbers really matter? *Journal of Artificial Intelligence Research*, 17:265–287, 2002.

[Chan and Darwiche, 2004] H. Chan and A. Darwiche. Sensitivity analysis in bayesian networks: from single to multiple parameters. In *UAI*, pages 67–75, 2004.

[Darwiche, 2003] A. Darwiche. A differential approach to inference in bayesian networks. *Journal of the ACM*, 50(3):280–305, 2003.

[Goldszmidt and Pearl, 1992] M. Goldszmidt and J. Pearl. Rank-based systems: A simple approach to belief revision, belief update, and reasoning about evidence and actions. In *KR*, pages 661–672, 1992.

[Jensen and Andersen, 1990] F. Jensen and S. Andersen. Approximations in bayesian belief universes for knowledge based systems. In *UAI*, pages 162–169, 1990.

[kdd, 2006] *http://sourceforge.net/projects/bnj/*. 2006.

[Park and Darwiche, 2004] J. Park and A. Darwiche. A differential semantics for jointree algorithms. *Artificial Intelligence*, 156(2):197–216, 2004.

[Pearl, 1988] J. Pearl. *Probabilistic reasoning in intelligent systems: networks of plausible inference*. Morgan Kaufmann Publishers Inc., San Francisco, 1988.

[Pearl, 1993] J. Pearl. Comment: Graphical models, causality, and intervention. *Statistical Science*, 8(3):266–269, 1993.

[Pearl, 2009a] J. Pearl. Causal inference in statistics: An overview. *Statistics Surveys*, 3:96–146, 2009.

[Pearl, 2009b] Judea Pearl. *Causality: Models, Reasoning, and Inference*. Cambridge University Press, Cambridge, 2009.

[Russell *et al.*, 1995] S. Russell, J. Binder, D. Koller, and K. Kanazawa. Local learning in probabilistic networks with hidden variables. In *IJCAI*, pages 1146–1152, 1995.

[Shachter, 1986] R. Shachter. Evaluating influence diagrams. *Operations Research*, 34(6):871–882, 1986.

[Shenoy and Shafer, 1986] P. Shenoy and G. Shafer. Propagating belief functions with local computations. *IEEE Expert*, 1(3):43–52, 1986.