A Supplementary Material: Causal Inference Algorithm based on SCM

The appendix to this section is an extended description of the causal inference algorithm mentioned in this paper, showing specific algorithmic implementations of the ten causal inference problems. For simpler inference tasks, only a brief verbal description is given, while for more complex causal inference tasks we show the specific algorithmic logic of causal inference problems.

A.1 Correlation

Problem statement: Given a causal graph G and a probabilistic data set D, ask what happens to the value of Y when the value of X changes.

Algorithm description: Take $P(Y|X=1) - P(Y|X=0)$ from the probabilistic data set and determine the answer based on the result of the calculation ($P(Y|X)$ has to be calculated based on the joint probability).

A.2 Marginal Distribution

Problem statement: Given a causal graph G and a probabilistic data set D, ask for the edge probability of node X.

Algorithm description: According to the total probability formula, sum all parent nodes s_i of X in G by weighting them, i.e. $\sum_{s_i} P(X|s_i) * P(s_i)$.

A.3 Explaining Away Effect

Problem statement: Given a causal graph G and a probabilistic data set D, ask what effect the occurrence of event X would have on the value of Y, given observation V.

Algorithm description. Take $P(Y|X=1, V=1) - P(Y|V=1)$ from the probabilistic data set and determine the answer based on the result of the calculation (you need to calculate $P(Y|V=1) = P(Y|X=0, V=1) * P(X=0) + P(Y|X=1, V=1) * P(X=1)$ based on the total probability formula).

A.4 Average Treatment Effect

Problem statement. Given a causal graph G and a probabilistic data set D, ask about the true causal effect of X on Y.

Algorithm description: See Algorithm **??** for details.

A.5 Backdoor Adjustment set

Problem statement. Given a causal graph G and a probabilistic data set D, ask whether case-by-case analysis of the causal effect of X on Y based on V is better than studying the causal effect of X on Y directly.

Algorithm description. Find the set of backdoor adjustment set from X to Y to compare with V.

A.6 Collider Bias

Problem statement. Given a causal graph G and the difference between the probabilities of X and Y when V is observed, ask about the true causal effect of X on Y, given observation V.

Algorithm description. Determine whether X, Y, and V form a "collider" structure to determine the answer to the question.

A.7 Effect of treatment on the treated

Problem statement. Given a causal graph G and a probabilistic data set D, ask how changing the value of X affects Y in the group where $X = 1$ in the observed data.

Algorithm description. See Algorithm 1 for details.

Algorithm 1: Effect of treatment on the treated

```
Data: G, D, {X, Y, V}
Result: causal effect
begin
    backdoor_set = {P(V), P(Y|V, X)}
    (V is a backdoor set from X to Y)
    if backdoor_set == ∅ then
        if P(Y|X) ∈ D then
            return P(Y|X = 1) − P(Y|X = 0)
        else
            return error
        end if
    else
        if backdoor_set ∈ D then
            return
            Σ_z (P(Y = 1|Z = z, X = 1) − P(Y = 1|Z = z, X = 0)) ∗ P(Z = z)
        else
            frontdoor_set = {P(X), P(M|X), P(Y|M, X)}
            (M is a frontdoor set from X to Y)
            if frontdoor_set ∈ D then
                return
                Σ_v P(Y = 1|X = 1, V = v) ∗ (P(V = v|X = 1) − P(V = v|X = 0))
            else
                return error
            end if
        end if
    end if
    return error
end
```

A.8 Natural Direct Effect

Problem statement. Given a causal graph G and a probabilistic data set D, ask what is the true causal effect of X on Y after ignoring the influence of a particular mediating element V.

Algorithm description. See Algorithm 2 for details.

Algorithm 2: Natural Direct Effect

Data: $G, D, \{X, Y, V\}$
Result: causal effect
begin
  $backdoor\_set = \{P(V), P(Y|V, X)\}$
  (V is a backdoor set from X to Y)
  if $backdoor\_set == \emptyset$ **then**
    if $P(Y|X, V), P(V|X) \in D$ **then**
      return
      $\sum_v (P(Y|X = 1, V = v) - P(Y|X = 0, V = v)) * P(V = v|X = 0)$
    else
      return ***error***
    end if
  else
    if $backdoor\_set \in D$ **then**
      return $\sum_v \sum_k (P(Y = 1|X = 1, V = v) - P(Y = 1|X = 0, V = v))$
        $*P(V = v|X = 0, K = k) * P(K = k)$
    else
      return ***error***
    end if
  end if
  return ***error***
end

A.9 Natural Indirect Effect

Problem statement. Given a causal graph G and a probabilistic data set D, ask whether X will have a causal effect on Y through the mediating element V.

Algorithm description. See Algorithm 3 for details.

A.10 Counterfactual (deterministic)

Problem statement. Given a causal graph G and a set of node dependencies, the nodes first take values, asking what the value of Y is if X takes the value of x.

Algorithm description. According to the counterfactual inference algorithm, the incoming edge of X in G is cut and propagated forward on the graph to obtain the value of Y .

Algorithm 3: Natural Indirect Effect

```
Data: G, D, {X, Y, V}
Result: causal effect
begin
    backdoor_set = {P(V), P(Y|V, X)}
    (V is a backdoor set from X to Y)
    path_set = GETPATH(X,Y,G)
    if backdoor_set == ∅ and {V} == path_set then
        if P(Y|X) ∈ D then
            return P(Y|X = 1) − P(Y|X = 0)
        else
            return error
        end if
    else
        if backdoor_set == ∅ then
            backdoor_set_2 = BACKDOOR(V,Y,G)
            if backdoor_set_2 == ∅ then
                if P(V|X), P(Y|X, V) ∈ D then
                    return ∑_v (P(V = v|X = 1) − P(V = v|X = 0)) ∗ P(Y =
                      1|X = 0, V = v)
                else
                    return error
                end if
            else
                return ∑_v ∑_k (P(V = v|X = 1, K = k) − P(V = v|X = 0, K =
                  k)) ∗ P(Y = 1|X = 0, V = v) ∗ P(K = k)
            end if
        else
            if backdoor_set_2 ∈ D then
                return ∑_v ∑_k (P(V = v|X = 1, K = k) − P(V = v|X = 0, K =
                  k)) ∗ P(Y = 1|X = 0, V = v) ∗ P(K = k)
            else
                frontdoor_set = {P(X), P(M|X), P(Y|M, X)}
                (M is a frontdoor set from X to Y)
                if frontdoor_set ∈ D then
                    return ∑_v (P(V = v|X = 1) − P(V = v|X = 0)) ∑_k (P(Y =
                      1|V = v, K = k) ∗ P(K = k))
                else
                    return error
                end if
            end if
        end if
    end if
    return error
end
```