Score based learning of Bayesian networks

Jiří Vomlel

Academy of Sciences of the Czech Republic

10th July, 2007

Conditional Independence

Example (CI model)

Assume three variables:

- person's length of hair, denoted by H,

Conditional Independence

Example (CI model)

Assume three variables:

- person's length of hair, denoted by H,
- person's stature, denoted by S, and

Conditional Independence

Example (CI model)

Assume three variables:

- person's length of hair, denoted by H,
- person's stature, denoted by S, and
- person's gender, denoted by G.

Conditional Independence

Example (CI model)

Assume three variables:

- person's length of hair, denoted by H,
- person's stature, denoted by S, and
- person's gender, denoted by G.

We can describe relations between these three variables as follows:

Conditional Independence

Example (CI model)

Assume three variables:

- person's length of hair, denoted by H,
- person's stature, denoted by S, and
- person's gender, denoted by G.

We can describe relations between these three variables as follows:

- Seeing the *length of hair* of a person will tell us more about his/her *gender* and conversely. It means, the value of G is dependent on the value of H.

Conditional Independence

Example (CI model)

Assume three variables:

- person's length of hair, denoted by H,
- person's stature, denoted by S, and
- person's gender, denoted by G.

We can describe relations between these three variables as follows:

- Seeing the *length of hair* of a person will tell us more about his/her *gender* and conversely. It means, the value of G is dependent on the value of H.
- Knowing more about the *gender* will focus our belief on his/her *stature* - S is dependent on G and (through G) also on H.

Conditional Independence

Example (CI model)

Assume three variables:

- person's length of hair, denoted by H,
- person's stature, denoted by S, and
- person's gender, denoted by G.

We can describe relations between these three variables as follows:

- Seeing the *length of hair* of a person will tell us more about his/her *gender* and conversely. It means, the value of G is dependent on the value of H.
- Knowing more about the *gender* will focus our belief on his/her *stature* - S is dependent on G and (through G) also on H.
- Nevertheless, if we know the *gender* of a person then *length of hair* of that person gives us no extra clue on his/her *stature* - H is independent of S given G.

Conditional Independence Statements

Definition (CI statement)

Let A, B, C be pairwise disjoint subsets of a set of variables N. Then the statement "A is conditionally independent of B given C" is a *CI statement* (over N), written as $I(A, B, C)$.

Conditional Independence Statements

Definition (CI statement)

Let A, B, C be pairwise disjoint subsets of a set of variables N. Then the statement "A is conditionally independent of B given C" is a *CI statement* (over N), written as $I(A, B, C)$.

Example (CI statement)

In Example 1 we have indicated only one CI statement, $I(H, S, G)$. On the other hand, we have indicated two dependence statements, namely $\neg I(G, H) = \neg I(G, H, \emptyset)$ and $\neg I(S, G)$.

Conditional Independence (CI) model

Definition (CI in PDs)

Let P be a discrete probability distribution over N. Given any $A \subseteq N$, let $\boldsymbol{x}_A$ denote a configuration of values of variables $\boldsymbol{X}_A = \{X_i\}_{i \in A}$ and for $B \subseteq N \setminus A$ let $P(\boldsymbol{x}_A \mid \boldsymbol{x}_B)$ denote the conditional probability for $\boldsymbol{X}_A = \boldsymbol{x}_A$ given $\boldsymbol{X}_B = \boldsymbol{x}_B$. The CI statement $I(A, B, C)$ is induced by probability distribution P over N if for all $\boldsymbol{x}_A, \boldsymbol{x}_B, \boldsymbol{x}_C$ such that $P(\boldsymbol{x}_C) > 0$

$$P(\boldsymbol{x}_A, \boldsymbol{x}_B \mid \boldsymbol{x}_C) \quad = \quad P(\boldsymbol{x}_A \mid \boldsymbol{x}_C) \cdot P(\boldsymbol{x}_B \mid \boldsymbol{x}_C) \ .$$

Conditional Independence (CI) model

Definition (CI in PDs)

Let P be a discrete probability distribution over N. Given any $A \subseteq N$, let $\boldsymbol{x}_A$ denote a configuration of values of variables $\boldsymbol{X}_A = \{X_i\}_{i \in A}$ and for $B \subseteq N \setminus A$ let $P(\boldsymbol{x}_A \mid \boldsymbol{x}_B)$ denote the conditional probability for $\boldsymbol{X}_A = \boldsymbol{x}_A$ given $\boldsymbol{X}_B = \boldsymbol{x}_B$. The CI statement $I(A, B, C)$ is induced by probability distribution P over N if for all $\boldsymbol{x}_A, \boldsymbol{x}_B, \boldsymbol{x}_C$ such that $P(\boldsymbol{x}_C) > 0$

$$P(\boldsymbol{x}_A, \boldsymbol{x}_B \mid \boldsymbol{x}_C) \quad = \quad P(\boldsymbol{x}_A \mid \boldsymbol{x}_C) \cdot P(\boldsymbol{x}_B \mid \boldsymbol{x}_C) \ .$$

Example

In Example 1 we have indicated CI statement, $I(H, S, G)$. For all values h, s, g of variables H, S, G it holds that

$$P(h, s \mid g) \quad = \quad P(h \mid g) \cdot P(s \mid g)$$

Conditional Independence (CI) model

Definition (CI in PDs)

Let P be a discrete probability distribution over N. Given any $A \subseteq N$, let $\boldsymbol{x}_A$ denote a configuration of values of variables $\boldsymbol{X}_A = \{X_i\}_{i \in A}$ and for $B \subseteq N \setminus A$ let $P(\boldsymbol{x}_A \mid \boldsymbol{x}_B)$ denote the conditional probability for $\boldsymbol{X}_A = \boldsymbol{x}_A$ given $\boldsymbol{X}_B = \boldsymbol{x}_B$. The CI statement $I(A, B, C)$ is induced by probability distribution P over N if for all $\boldsymbol{x}_A, \boldsymbol{x}_B, \boldsymbol{x}_C$ such that $P(\boldsymbol{x}_C) > 0$

$$P(\boldsymbol{x}_A, \boldsymbol{x}_B \mid \boldsymbol{x}_C) = P(\boldsymbol{x}_A \mid \boldsymbol{x}_C) \cdot P(\boldsymbol{x}_B \mid \boldsymbol{x}_C) .$$

Example

In Example 1 we have indicated CI statement, $I(H, S, G)$. For all values h, s, g of variables H, S, G it holds that

$$\begin{aligned} P(h, s \mid g) &= P(h \mid g) \cdot P(s \mid g) \text{ or, equivalently} \\ P(h \mid g, s) &= P(h \mid g) \end{aligned}$$

What CI-statements are represented by a DAG?

Definition (d-separation criteria)

Two nodes a and b in a DAG G are d-seprated by a set C if for all paths between a and b there is a node c ($c \neq a$ and $c \neq b$) such that either:

What CI-statements are represented by a DAG?

Definition (d-separation criteria)

Two nodes a and b in a DAG G are d-seprated by a set C if for all paths between a and b there is a node c ($c \neq a$ and $c \neq b$) such that either:

- the path contains a node $c \in C$, in which edges **do not meet "head-to-head"** or

What CI-statements are represented by a DAG?

Definition (d-separation criteria)

Two nodes a and b in a DAG G are d-seprated by a set C if for all paths between a and b there is a node c ($c \neq a$ and $c \neq b$) such that either:

- the path contains a node $c \in C$, in which edges **do not meet "head-to-head"** or
- the path contains a node c in which edges **meet "head-to-head"** and neither c nor any of its descendants belong to C.

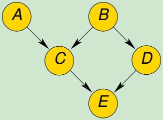
What CI-statements are represented by a DAG?

Definition (d-separation criteria)

Two nodes a and b in a DAG G are d-seprated by a set C if for all paths between a and b there is a node c ($c \neq a$ and $c \neq b$) such that either:

- the path contains a node $c \in C$, in which edges **do not meet "head-to-head"** or
- the path contains a node c in which edges **meet "head-to-head"** and neither c nor any of its descendants belong to C.

Example



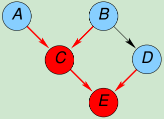
What CI-statements are represented by a DAG?

Definition (d-separation criteria)

Two nodes a and b in a DAG G are d-seprated by a set C if for all paths between a and b there is a node c ($c \neq a$ and $c \neq b$) such that either:

- the path contains a node $c \in C$, in which edges **do not meet "head-to-head"** or
- the path contains a node c in which edges **meet "head-to-head"** and neither c nor any of its descendants belong to C.

Example



$I(A, D)$

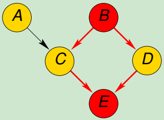
What CI-statements are represented by a DAG?

Definition (d-separation criteria)

Two nodes a and b in a DAG G are d-seprated by a set C if for all paths between a and b there is a node c ($c \neq a$ and $c \neq b$) such that either:

- the path contains a node $c \in C$, in which edges **do not meet "head-to-head"** or
- the path contains a node c in which edges **meet "head-to-head"** and neither c nor any of its descendants belong to C.

Example



$I(A, D, B)$

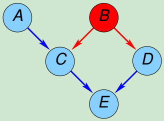
What CI-statements are represented by a DAG?

Definition (d-separation criteria)

Two nodes a and b in a DAG G are d-seprated by a set C if for all paths between a and b there is a node c ($c \neq a$ and $c \neq b$) such that either:

- the path contains a node $c \in C$, in which edges **do not meet "head-to-head"** or
- the path contains a node c in which edges **meet "head-to-head"** and neither c nor any of its descendants belong to C.

Example



$\neg I(A, D, \{B, E\})$

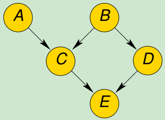
What CI-statements are represented by a DAG?

Definition (d-separation criteria)

Two nodes a and b in a DAG G are d-seprated by a set C if for all paths between a and b there is a node c ($c \neq a$ and $c \neq b$) such that either:

- the path contains a node $c \in C$, in which edges **do not meet "head-to-head"** or
- the path contains a node c in which edges **meet "head-to-head"** and neither c nor any of its descendants belong to C.

Example



$?I(C, D, E)$

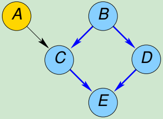
What CI-statements are represented by a DAG?

Definition (d-separation criteria)

Two nodes a and b in a DAG G are d-seprated by a set C if for all paths between a and b there is a node c ($c \neq a$ and $c \neq b$) such that either:

- the path contains a node $c \in C$, in which edges **do not meet "head-to-head"** or
- the path contains a node c in which edges **meet "head-to-head"** and neither c nor any of its descendants belong to C.

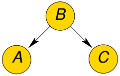
Example

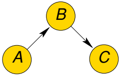


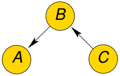
$\neg I(C, D, E)$

Equivalence classes of Bayesian networks

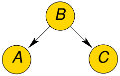
What independence statements are represented by these three models?

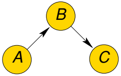


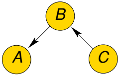




What independence statements are represented by these three models?



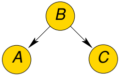




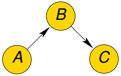
$I(A, C, B)$

Equivalence classes of Bayesian networks

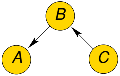
What independence statements are represented by these three models?



$I(A, C, B)$

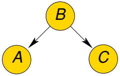


$I(A, C, B)$

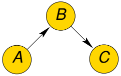


Equivalence classes of Bayesian networks

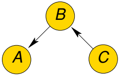
What independence statements are represented by these three models?



$I(A, C, B)$



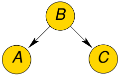
$I(A, C, B)$



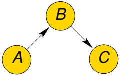
$I(A, C, B)$

Equivalence classes of Bayesian networks

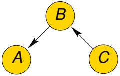
What independence statements are represented by these three models?



$I(A, C, B)$



$I(A, C, B)$

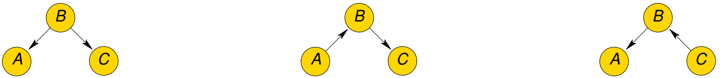


$I(A, C, B)$

Different graphs may represent the same set of CI-statements!

Equivalence classes of Bayesian networks

What independence statements are represented by these three models?



$I(A, C, B)$ $I(A, C, B)$ $I(A, C, B)$

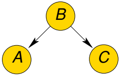
Different graphs may represent the same set of CI-statements!

Definition (Equivalence class)

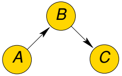
We say that Bayesian networks with DAGs representing the same set of CI-statements belong to an equivalence class.

Equivalence classes of Bayesian networks

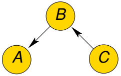
What independence statements are represented by these three models?



$I(A, C, B)$



$I(A, C, B)$



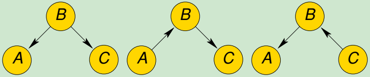
$I(A, C, B)$

Different graphs may represent the same set of CI-statements!

Definition (Equivalence class)

We say that Bayesian networks with DAGs representing the same set of CI-statements belong to an equivalence class.

Example



belong to the same equivalence class.

Equivalence classes of Bayesian networks

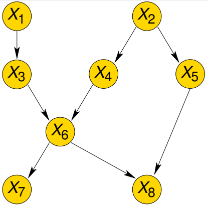
Definition (Immorality)

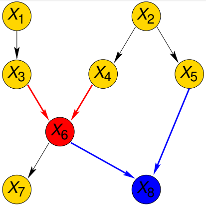
An immorality in a DAG G is a induced subgraph of G for a set $\{A, B, C\}$, where A, B, C are distinct nodes of G such that there are edges $A \rightarrow C$ and $B \rightarrow C$ and there is no edge between A and B in G.

Equivalence classes of Bayesian networks

Definition (Immorality)

An immorality in a DAG G is a induced subgraph of G for a set $\{A, B, C\}$, where A, B, C are distinct nodes of G such that there are edges $A \rightarrow C$ and $B \rightarrow C$ and there is no edge between A and B in G.





Equivalence classes of Bayesian networks

Definition (Immorality)

An immorality in a DAG G is a induced subgraph of G for a set $\{A, B, C\}$, where A, B, C are distinct nodes of G such that there are edges $A \rightarrow C$ and $B \rightarrow C$ and there is no edge between A and B in G.

Definition (Underlaying graph)

An underlaying graph of a DAG is the undirected graph that has the same set of nodes and all directed edges $A \rightarrow B$ are replaced by undirected edges $A - B$.

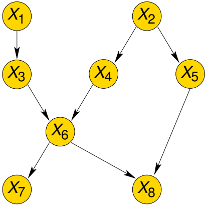
Equivalence classes of Bayesian networks

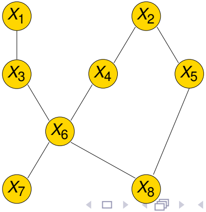
Definition (Immorality)

An immorality in a DAG G is a induced subgraph of G for a set $\{A, B, C\}$, where A, B, C are distinct nodes of G such that there are edges $A \rightarrow C$ and $B \rightarrow C$ and there is no edge between A and B in G.

Definition (Underlaying graph)

An underlaying graph of a DAG is the undirected graph that has the same set of nodes and all directed edges $A \rightarrow B$ are replaced by undirected edges $A - B$.





Equivalence classes of Bayesian networks

Definition (Immorality)

An immorality in a DAG G is a induced subgraph of G for a set $\{A, B, C\}$, where A, B, C are distinct nodes of G such that there are edges $A \to C$ and $B \to C$ and there is no edge between A and B in G.

Definition (Underlaying graph)

An underlaying graph of a DAG is the undirected graph that has the same set of nodes and all directed edges $A \to B$ are replaced by undirected edges $A - B$.

Theorem

Bayesian networks belong to the same equivalence class iff they have the same underlaying graph and the same set of immoralities.

Essential graphs

Definition (Essential graph)

The essential graph G^* of an equivalence class $\mathcal{G}$ of DAGs over N is a hybrid graph over N defined as follows:

- $a \rightarrow b$ in G^* if $a \rightarrow b$ in G for every $G \in \mathcal{G}$,

Essential graphs

Definition (Essential graph)

The essential graph G^* of an equivalence class $\mathcal{G}$ of DAGs over N is a hybrid graph over N defined as follows:

- $a \rightarrow b$ in G^* if $a \rightarrow b$ in G for every $G \in \mathcal{G}$,
- $a - b$ in G^* if $\exists\, G_1, G_2 \in \mathcal{G}$ such that $a \rightarrow b$ in G_1 and $a \leftarrow b$ in G_2.

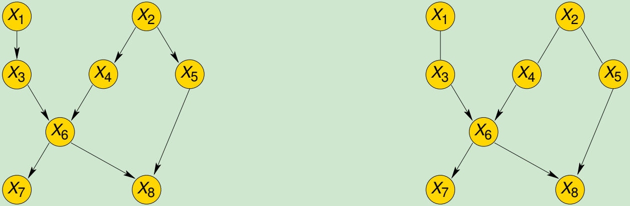
Essential graphs

Definition (Essential graph)

The essential graph G^* of an equivalence class $\mathcal{G}$ of DAGs over N is a hybrid graph over N defined as follows:

- $a \to b$ in G^* if $a \to b$ in G for every $G \in \mathcal{G}$,
- $a - b$ in G^* if $\exists\, G_1, G_2 \in \mathcal{G}$ such that $a \to b$ in G_1 and $a \leftarrow b$ in G_2.

Example



Inclusion neigbourhood

$\mathcal{M}_G$ will denote the set of CI-statements generated by a DAG G.

Inclusion neigbourhood

$\mathcal{M}_G$ will denote the set of CI-statements generated by a DAG G.

Definition

Given two DAGs K, L over N, we say that they are inclusion neighbors and write $\mathcal{M}_K \sqsubset \mathcal{M}_L$ if $\mathcal{M}_K \subset \mathcal{M}_L$ and there is no DAG G such that $\mathcal{M}_K \sqsubset \mathcal{M}_G \sqsubset \mathcal{M}_L$.

Inclusion neigbourhood

$\mathcal{M}_G$ will denote the set of CI-statements generated by a DAG G.

Definition

Given two DAGs K, L over N, we say that they are inclusion neighbors and write $\mathcal{M}_K \sqsubset \mathcal{M}_L$ if $\mathcal{M}_K \subset \mathcal{M}_L$ and there is no DAG G such that $\mathcal{M}_K \sqsubset \mathcal{M}_G \sqsubset \mathcal{M}_L$. We say then that $\mathcal{M}_L$ is an upper neighbor of $\mathcal{M}_K$ or, dually, that $\mathcal{M}_K$ is a lower neighbor of $\mathcal{M}_L$.

Inclusion neigbourhood

$\mathcal{M}_G$ will denote the set of CI-statements generated by a DAG G.

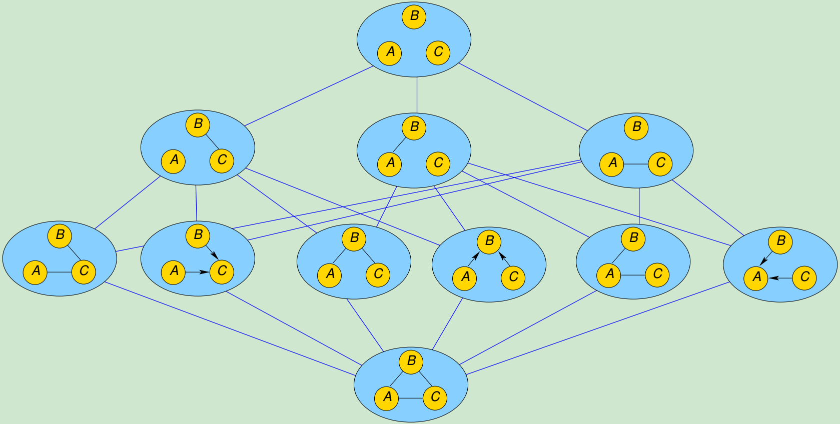
Definition

Given two DAGs K, L over N, we say that they are inclusion neighbors and write $\mathcal{M}_K \sqsubset \mathcal{M}_L$ if $\mathcal{M}_K \subset \mathcal{M}_L$ and there is no DAG G such that $\mathcal{M}_K \sqsubset \mathcal{M}_G \sqsubset \mathcal{M}_L$. We say then that $\mathcal{M}_L$ is an upper neighbor of $\mathcal{M}_K$ or, dually, that $\mathcal{M}_K$ is a lower neighbor of $\mathcal{M}_L$.

The inclusion neigborhood allows us to define a greedy search procedure that finds a globally optimal Bayesian network.

Search space for models of three variables

Example



Likelihood of data

- Let $D = \{\boldsymbol{x}^m, m = 1, \ldots, M\}$ be the learning dataset, where $\boldsymbol{x}$ is the vector of values of variable $\boldsymbol{X} = \{X_i\}_{i=1}^{N}$,

Likelihood of data

- Let $D = \{\boldsymbol{x}^m, m = 1, \ldots, M\}$ be the learning dataset, where $\boldsymbol{x}$ is the vector of values of variable $\boldsymbol{X} = \{X_i\}_{i=1}^N$,
- $X_i, i = 1, \ldots, N$ be the variables and nodes of the graph G of Bayesian network,

Likelihood of data

- Let $D = \{\boldsymbol{x}^m, m = 1, \ldots, M\}$ be the learning dataset, where $\boldsymbol{x}$ is the vector of values of variable $\boldsymbol{X} = \{X_i\}_{i=1}^N$,
- $X_i, i = 1, \ldots, N$ be the variables and nodes of the graph G of Bayesian network,
- $r(i)$ denote number of states of variable X_i,

Likelihood of data

- Let $D = \{\boldsymbol{x}^m, m = 1, \ldots, M\}$ be the learning dataset, where $\boldsymbol{x}$ is the vector of values of variable $\boldsymbol{X} = \{X_i\}_{i=1}^N$,
- $X_i, i = 1, \ldots, N$ be the variables and nodes of the graph G of Bayesian network,
- $r(i)$ denote number of states of variable X_i,
- $q(i, G)$ denote number of parent configurations for parents $\boldsymbol{X}_{pa(i)}$ of variable X_i, and

Likelihood of data

- Let $D = \{\boldsymbol{x}^m, m = 1, \ldots, M\}$ be the learning dataset, where $\boldsymbol{x}$ is the vector of values of variable $\boldsymbol{X} = \{X_i\}_{i=1}^N$,
- $X_i, i = 1, \ldots, N$ be the variables and nodes of the graph G of Bayesian network,
- $r(i)$ denote number of states of variable X_i,
- $q(i, G)$ denote number of parent configurations for parents $\boldsymbol{X}_{pa(i)}$ of variable X_i, and
- $N(i, j, k)$ denote ocurrance of the corresponding configuration in the learning dataset D.

Likelihood of data

- Let $D = \{\boldsymbol{x}^m, m = 1, \ldots, M\}$ be the learning dataset, where $\boldsymbol{x}$ is the vector of values of variable $\boldsymbol{X} = \{X_i\}_{i=1}^N$,
- $X_i, i = 1, \ldots, N$ be the variables and nodes of the graph G of Bayesian network,
- $r(i)$ denote number of states of variable X_i,
- $q(i, G)$ denote number of parent configurations for parents $\boldsymbol{X}_{pa(i)}$ of variable X_i, and
- $N(i, j, k)$ denote ocurrance of the corresponding configuration in the learning dataset D.

The likelihood of D given G is the probability of data D being generated from the Bayesian network model with the structure given by directed acyclic graph G and representing joint probability distribution P is

$$P(D|G) = \prod_{m=1}^{M} P(\boldsymbol{X} = \boldsymbol{x}^m)$$

Scores

Lemma (Maximum loglikelihood)

The maximum log-likelihood for a given Bayesian network with graph G is

$$MLL(G|D) = \sum_{i=1}^{N} \sum_{k=1}^{r(i)} \sum_{j=1}^{q(i,G)} N(i,j,k) \log \frac{N(i,j,k)}{N(i,j)}$$

Scores

Lemma (Maximum loglikelihood)

The maximum log-likelihood for a given Bayesian network with graph G is

$$MLL(G|D) = \sum_{i=1}^{N} \sum_{k=1}^{r(i)} \sum_{j=1}^{q(i,G)} N(i,j,k) \log \frac{N(i,j,k)}{N(i,j)}$$

Let $d(G)$ be the number of free parameters in the Bayesian network model with graph G. It is given by

$$d(G) = \sum_{i=1}^{N} (r(i)-1)q(i,G)$$

Scores

Definition (Akaike Information Criterion)

$$AIC(G|D) = MLL(G|D) - d(G)$$

Scores

Definition (Akaike Information Criterion)

$$AIC(G|D) = MLL(G|D) - d(G)$$

Definition (Bayesian Information Criterion)

$$BIC(G|D) = MLL(G|D) - \frac{\log M}{2} d(G)$$

Greedy Equivalence Search (GES) algorithm

The GES algorithm starts with an empty graph and has two stages:

1. deleting CI-statements, which corresponds to edge additions and directing some other edges

Greedy Equivalence Search (GES) algorithm

The GES algorithm starts with an empty graph and has two stages:

1. deleting CI-statements, which corresponds to edge additions and directing some other edges
2. adding CI-statements, which corresponds to edge removal.

Greedy Equivalence Search (GES) algorithm

The GES algorithm starts with an empty graph and has two stages:

1. deleting CI-statements, which corresponds to edge additions and directing some other edges
2. adding CI-statements, which corresponds to edge removal.

In each step of the GES algorithm:

Greedy Equivalence Search (GES) algorithm

The GES algorithm starts with an empty graph and has two stages:

1. deleting CI-statements, which corresponds to edge additions and directing some other edges
2. adding CI-statements, which corresponds to edge removal.

In each step of the GES algorithm:

- we search only in the inclusion neigborhood,

Greedy Equivalence Search (GES) algorithm

The GES algorithm starts with an empty graph and has two stages:

1. deleting CI-statements, which corresponds to edge additions and directing some other edges
2. adding CI-statements, which corresponds to edge removal.

In each step of the GES algorithm:

- we search only in the inclusion neigborhood,
- select the model that maximizes the criteria, and

Greedy Equivalence Search (GES) algorithm

The GES algorithm starts with an empty graph and has two stages:

1. deleting CI-statements, which corresponds to edge additions and directing some other edges
2. adding CI-statements, which corresponds to edge removal.

In each step of the GES algorithm:

- we search only in the inclusion neigborhood,
- select the model that maximizes the criteria, and
- if there is no better model than the current one we start the second stage or terminate if we are in the second stage,

Greedy Equivalence Search (GES) algorithm

The GES algorithm starts with an empty graph and has two stages:

1. deleting CI-statements, which corresponds to edge additions and directing some other edges
2. adding CI-statements, which corresponds to edge removal.

In each step of the GES algorithm:

- we search only in the inclusion neigborhood,
- select the model that maximizes the criteria, and
- if there is no better model than the current one we start the second stage or terminate if we are in the second stage,

Theorem

When the greedy equivalent search algorithm terminates, the current model is the global optimum. If the dataset is "faithfull" with respect to the generative model then the algorithm terminates in this model.

Greedy Equivalence Search (GES) algorithm

The GES algorithm starts with an empty graph and has two stages:

1. deleting CI-statements, which corresponds to edge additions and directing some other edges
2. adding CI-statements, which corresponds to edge removal.

In each step of the GES algorithm:

- we search only in the inclusion neigborhood,
- select the model that maximizes the criteria, and
- if there is no better model than the current one we start the second stage or terminate if we are in the second stage,

Theorem

When the greedy equivalent search algorithm terminates, the current model is the global optimum. If the dataset is "faithfull" with respect to the generative model then the algorithm terminates in this model.

Demo of GES in R.