**Bayesian Networks**

- Terminology of Bayesian Network
  - Conditional Independence: If every undirected path from a node in X to a node in Y is d-separated by E, then X and Y are conditionally independent given E.
  - D-separate: A set of node E d-separates two sets of nodes X and Y if every undirected path from a node in X to a node in Y is blocked given E.
  - Block Conditions (1) Z is in E and Z has one arrow on the path leading in and one arrow out
     (2) Z is in E and Z has both path arrows leading out
     (3) Neither Z nor any descendant of Z is in E, and both arrows lead in to Z

- Bayesian Network
  A directed acyclic graph that represents a joint probability distribution for a set of random variables.
  - Vertices (nodes): denote events (each a random variable)
  - Edges (arcs, links): denote conditional dependencies
  - Conditional probability tables (CPT): Assumptions - Each node is asserted to be conditionally dependent of its nondescendants, given its immediate parents.
  - Chain Rule for (Exact) inference in Bayesian networks
    \[ P(X_1, X_2, \ldots, X_n) = \prod_{i=1}^{n} P(X_i | Pa(X_i)) \]
  - Example
    \[ P(\text{do}) = 0.15 \]
    \[ P(\text{do} | \text{lo} \& \text{bp}) = 0.99 \]
    \[ P(\text{do} | \text{lo} \& \neg \text{bp}) = 0.01 \]

**Terminology of Bayesian network**

- Conditional Independence
- D-separate
- Maximize spanning tree structure
- "Sparse candidate" algorithm
- How to select candidate parents
- How to find the maximize score of a Bayesian network
- Experimental Evaluation

**Introduction to Bayesian Network**

- Why learn a Bayesian network?
  - Solves the uncertain problems that are difficult for logic inference
  - Combines knowledge engineering and statistical induction
  - Covers the whole spectrum from knowledge-intensive model construction to data-intensive model induction
  - More than a learning black-box
  - Causal representation, reasoning, and discovery
  - Increasing interests in AI

**Bayesian Networks**

- Bayesian Network
  A directed acyclic graph that represents a joint probability distribution for a set of random variables.
  - Vertices (nodes): denote events (each a random variable)
  - Edges (arcs, links): denote conditional dependencies
  - Conditional probability tables (CPT)
  - Assumptions - Each node is asserted to be conditionally dependent of its nondescendants, given its immediate parents.
  - Chain Rule for (Exact) inference in Bayesian networks
    \[ P(X_1, X_2, \ldots, X_n) = \prod_{i=1}^{n} P(X_i | Pa(X_i)) \]
  - Example
    \[ P(\text{do}) = 0.15 \]
    \[ P(\text{do} | \text{lo} \& \text{bp}) = 0.99 \]
    \[ P(\text{do} | \text{lo} \& \neg \text{bp}) = 0.01 \]
Bayesian Networks

- Score-Based
  - Define \( \text{scoring function} \) (aka \( \text{score} \)) that evaluates how well (in)dependencies in a structure match observations, such as Bayesian score and MDL.
  - Bayesian Score for Marginal Likelihood \( P(D|H) \)
    \[
    P(D|H) = \frac{1}{|H|} \prod \left[ \frac{\left( P'(X_1, X_2) \right)}{P'(X_1) P'(X_2)} \right] \frac{\left( P'(X_1, X_2, X_3) \right)}{P'(X_1, X_2)}
    \]
    where \( x_i \) = \( x_i \) particular value of \( X \), \( Pa(X) \) = particular value of Parents of \( X \).
  - \( \mathcal{F}(i) = (0 - 1) \) for \( i \in Z^+ \).
  - \( \text{Search for structure that maximizes score} \)
  - \( \text{Dependability} \)
    \[ \text{Score}(G,D) = \sum \left[ \text{score}(X_i | \text{Par}(X_i) \cup \text{Pa}(X_i)) \right] \]

Common Properties
- \( \text{Soundness} \): with sufficient data and computation, both learn correct structure
- \( \text{Sufficiency} \): can incorporate knowledge
- \( \text{Constrain-based} \): sensitive to errors in test

Learning Structure

- Learning Weights (Conditional Probability Tables)
  - Given training data and network structure to learn target variable
  - Naive Bayesian networks
  - Given network structure and some training data to estimate unobserved variable values.
    - \( \text{Gradient ascent algorithm} \)
    - \( \text{Weight update rule} \)
    \[ w_{ik} \leftarrow w_{ik} + \frac{P(y_i | x_k, w_{ik})}{P(y_i)} \]
  - Given training data to build a network structure

Choosing Candidate Sets

- Shield Measure
  - Conditional mutual information - to measure the error of our assume that \( X \) and \( Y \) are independent given different values of \( Z \)
    \[ I(X; Y | Z) = \frac{D(P'(X, Y | Z) || P(X, Y) P(Z))}{P(Z)} \]
    - \( \text{Shield score} \)
    \[ M_{\text{shield}}(X_i, X_j | B) = I(X_i; X_j | \text{Par}(X_i)) - I(X_i; \text{Par}(X_j)) \]
  - \( \text{Score measure} \)
    \[ M_{\text{score}}(X_i, X_j | B) = \text{Score}(X_i; X_j | \text{Par}(X_i), B) \]
  - \( \text{Handles random variables with multiple values} \)
  - \( \text{Chain rule of mutual information} \)
    \[ I(X_i, X_j; \text{Par}(X_i)) = I(X_i; \text{Par}(X_i)) + I(X_j; \text{Par}(X_i)) \]
  - \( \text{Shield measure} \)
    \[ M_{\text{shield}}(X_i, X_j | B) = I(X_i, X_j; \text{Par}(X_i), \text{Par}(X_j)) \]
  - \( \text{Score measure} \)
    \[ M_{\text{score}}(X_i, X_j | B) = \text{Score}(X_i, X_j; \text{Par}(X_i), \text{Par}(X_j), B) \]

Learning with Small Candidate Sets

- \( \text{Maximal Restrict Bayesian Network (MRBN)} \)
  - \( \text{Input:} \) a set \( O = \{ X_1, \ldots , X^n \} \) of instances; a digraph \( H \) of bounded in-degree \( K \) and a decomposable score \( S \)
  - \( \text{Output:} \) a network \( B \) of \( G \) so that \( G \subset H \), that maximizes \( S \) with respect to \( D \)

- \( \text{Standard Heuristics} \)
  - No knowledge of expected structure, local change (e.g. arc deletion, arc addition, and arc reversal), and local maximum score
  - \( \text{Algorithm:} \) Greedy hill-climbing; Best-first search; Simulated annealing
  - \( \text{Time complexity} \) \( \text{in MRBN} \) is \( O(n) \) for initial calculation, then becomes \( O(k) \)

- \( \text{Divide and Conquer Heuristics} \)
  - \( \text{Input:} \) a digraph \( H = \{ X_i \rightarrow X_j : i < j \} \) and a set of weights \( w(x_i,y) \) for each \( X_i, Y \)
  - \( \text{Output:} \) an acyclic subgraph \( G \) of \( H \) that maximizes \( \sum_{i,j} w(x_i,y) \text{Pa}(x_i) \text{Pa}(x_j) \)
  - \( \text{Decompose} \) \( H \) by using standard graph decomposition methods
  - \( \text{Find a local maximum weight} \)
  - \( \text{Combine them into a global solution} \)
**Decomposition**

- **Strongly Connected Components (SCC)**
  - A subset of vertices $A$ is strongly connected if for each $X, Y \in A$, there is a directed path from $X$ to $Y$ and a directed path from $Y$ to $X$.
  - Decomposition of SCC into maximal sets that have no strongly connected components.

- **Separator Decomposition**
  - Searching a separator of $H$ which separate $H$ into $H_1$ and $H_2$ with no edges between them.

- **Cluster-Tree Decomposition**
  - Cluster tree definition.
  - Decomposing into cluster tree.

- **Cluster-Tree Heuristic**
  - A mixture of cluster-tree decomposition algorithm and standard heuristics.
  - Using for decomposition of $H$ for large size clusters.

**Experimental Evaluation**

<table>
<thead>
<tr>
<th>Method</th>
<th>Iter</th>
<th>Time</th>
<th>Score</th>
<th>KL</th>
<th>Stats</th>
</tr>
</thead>
<tbody>
<tr>
<td>Greedy</td>
<td>40</td>
<td>-15.35</td>
<td>0.0499</td>
<td>2656</td>
<td></td>
</tr>
<tr>
<td>Disc 5</td>
<td>1</td>
<td>14</td>
<td>-18.41</td>
<td>3.0608</td>
<td>908</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>19</td>
<td>-16.71</td>
<td>1.9396</td>
<td>1163</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>23</td>
<td>-16.21</td>
<td>0.7504</td>
<td>1163</td>
</tr>
<tr>
<td>Disc 10</td>
<td>1</td>
<td>20</td>
<td>-15.53</td>
<td>2.1580</td>
<td>1163</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>26</td>
<td>-16.71</td>
<td>1.3634</td>
<td>1063</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>32</td>
<td>-16.21</td>
<td>0.8704</td>
<td>1183</td>
</tr>
<tr>
<td>Shld 5</td>
<td>1</td>
<td>14</td>
<td>-17.50</td>
<td>2.1675</td>
<td>915</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>29</td>
<td>-17.25</td>
<td>1.8905</td>
<td>1728</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>36</td>
<td>-16.92</td>
<td>1.5632</td>
<td>1907</td>
</tr>
<tr>
<td>Shld 10</td>
<td>1</td>
<td>20</td>
<td>-15.86</td>
<td>0.5357</td>
<td>1244</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>35</td>
<td>-15.50</td>
<td>0.1989</td>
<td>1968</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>41</td>
<td>-15.50</td>
<td>0.1974</td>
<td>2109</td>
</tr>
<tr>
<td>Score 5</td>
<td>1</td>
<td>12</td>
<td>-15.86</td>
<td>0.7668</td>
<td>915</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>27</td>
<td>-15.54</td>
<td>0.6993</td>
<td>1838</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>34</td>
<td>-15.33</td>
<td>0.4797</td>
<td>2204</td>
</tr>
<tr>
<td>Score 10</td>
<td>1</td>
<td>17</td>
<td>-15.54</td>
<td>0.2153</td>
<td>1169</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>30</td>
<td>-15.31</td>
<td>0.0392</td>
<td>1649</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>36</td>
<td>-15.31</td>
<td>0.0352</td>
<td>2108</td>
</tr>
</tbody>
</table>

**Summary**

**Content Critique**

- **Key Contribution**: It presents an algorithm to select candidate sets and to discover efficiently the maximum score of Bayesian networks.
- **Strengths**: It uses scoring measure instead of mutual information to measure the dependency of parent and children, then uses the maximum score to build BBN, this algorithm can allow children to have multiple parents and handle random variables with multiple values.
- **Weaknesses**: The limited candidate sets provide a small hypothesis space.
- **Presentation Critique**: Audiences: Medical diagnosis; Mapping learning; language understanding; Image processing.