Lecture 22

Uncertainty Reasoning Presentation(2 of 4)
Learning Bayesian Networks from Data

Wednesday, March 8, 2000

Jincheng Gao
Department of Geography, KSU

Readings:
"Learning Bayesian Network Structure from Massive Datasets:
The 'Sparse Candidate' Algorithm"
Friedman, Nachman, and Peer

Presentation Outline

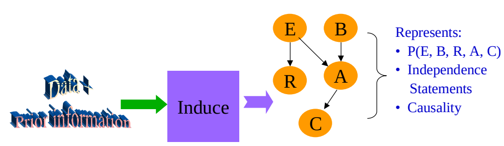
- **Paper**
  - **"Learning Bayesian Network Structure from Massive Datasets: The 'Sparse Candidate' Algorithm"**
  - **Author: Nir Friedman, Iftach Nachman and Dana Peer, Hebrew University, Israel**
- **Overview**
  - **Introduction to Bayesian Network**
  - **Outline of "Sparse Candidate" Algorithm**
  - **How to Choose Candidate Sets**
  - **Learning with Small Candidate Sets**
  - **Experimental Evaluation**
- **Goal**
  - **Introduces an algorithm that achieves a faster learning by restricting the search space**
- **References**
  - **Machine learning, T. M. Mitchell**
  - **Artificial Intelligence: A Modern Approach, S. J. Russell and P. Norvig**
  - **Bayesian Networks without Tears, E. Charniak**

Presentation Outline

- **Issues**
  - **How to guarantee all available candidate parents are selected**
  - **What is the criteria to stop its iteration to get a maximum score of network**
  - **Strengths: It presents a very useful algorithm to restrict search space in BBN**
  - **Weaknesses: It doesn't consider spurious dependent variables**
- **Outline**
  - **Why learn a Bayesian network**
  - **Introduction to Bayesian network**
    - **Terminology of Bayesian network**
    - **What is Bayesian network**
    - **How to construct a Bayesian network**
  - **"Sparse Candidate" algorithms**
    - **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

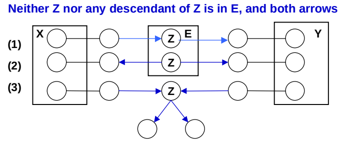
- **Why learn a Bayesian network?**
  - Solves the uncertain problems that are difficult for logic inference
  - Combines knowledge engineering and statistical induction
  - Covers the whole spectrum from knowledge-intensive model construction to data-intensive model induction
  - More than a learning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



Bayesian Networks

- **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

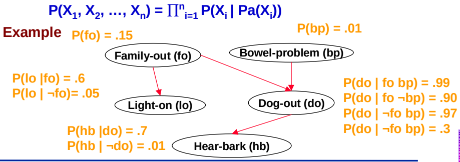


Bayesian Networks

- **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**

$$P(X_1, X_2, \ldots, X_n) = \prod^{n}_{i=1} P(X_i \mid Pa(X_i))$$

- **Example**



Bayesian Networks

- Score-Based
  - Define scoring function (*aka* 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)

$$P(D\,/\,h) \propto \prod_{i=1}^{n}\left[\prod_{Pa_i^h}\frac{\Gamma\left(\alpha\left(Pa_i^h\right)\right)}{\Gamma\left(\alpha\left(Pa_i^h\right)+N\left(Pa_i^h\right)\right)}\cdot\prod_{x_i=x_i}\frac{\Gamma\left(\alpha\left(x_i, Pa_i^h\right)+N\left(x_i, Pa_i^h\right)\right)}{\Gamma\left(\alpha\left(x_i, Pa_i^h\right)\right)}\right]$$

where $x_i \equiv x_{ij} \equiv$ particular value of X_i, $Pa_i^h \equiv Pa_{ik}^h \equiv$ particular value of $Parents_h(x_i)$.

$\Gamma(i) = (i-1)!$ for $i \in Z^+$

  - Search for structure that maximizes score
  - Decomposability Score(G:D) = $\sum_i$score(X_i | Par(X_i) : $N_{X_i, \text{par}(X_i)}$)
- Common Properties
  - Soundness: with sufficient data and computation, both learn correct structure
  - Both learn structure from observations and can incorporate knowledge
  - Constrain-based is sensitive to errors in test

Learning Structure

- Learning Weights (Conditional Probability Tables)
  - Given training data and network structure to learn target variable
    - Naïve Bayesian network
  - Given network structure and some training data to estimate unobserved variable values.
    - Gradient ascent algorithm
      - Weight update rule $w_{ijk} \leftarrow w_{ijk} + r\sum_{x \in D}\frac{P_h(y_{ij}, u_{ik}\,/\,x)}{w_{ijk}}$
  - Given training data to build a network structure
- Build structure of Bayesian networks
  - Constraint-Based
    - Perform tests of conditional independence
    - Search for network consistent with observed dependencies
    - Intuitive; closely follows definition of BBNs
    - Separates construction from form of CI tests

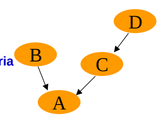
Learning Structure

- Algorithm *Max-Spanning-Tree-Structure*
  - Estimate $P(x)$ and $P(x, y)$ for all single random variables and pairs;
    $I(X; Y) = D_{KL}(P(X, Y) \,||\, P(X) \cdot P(Y))$
  - Build *complete* undirected graph:
    variables as vertices, $I(X; Y)$ as edge weights
  - $T \leftarrow$ *Build-MWST* ($V \times V$, *Weights*) // Chow-Liu algorithm: weight function $\equiv$ I
  - Set directional flow on T and place the CPTs on its edges (gradient learning)
  - RETURN: tree-structured BBN with CPT values
  - Advantage: Restricts hypothesis space and limits overfitting capability
  - Disadvantage: It only searches a single parent and some available data may be lost
- The "Sparse Candidate" Algorithm
  - It builds a network structure with maximal score by limiting H to at most K parents for each variables in BBN (K < N)
  - Searching Candidate sets K: Based on D and B_{n-1}, select for each variable X_i a set of C^n_i of candidate parents.
  - Maximize : Find a network B_n maximizing Score (B_n |D) among networks
  - Advantages: Overcoming the drawbacks of MSTS algorithm

Choosing Candidate Sets

- Discrepancy
  - Based on definition of the mutual information, it uses discrepancy between estimate P_B (X, Y) and the empirical estimate P'(X, Y).
    $M_{disc}(X_i, X_j / B) = D_{KL}(P'(X_i, X_j) // P_B(X_i, X_j))$
  - Algorithm
    - For the first loop: $M_{disc}(X_i, X_j / B_0) = I(X: Y)$.
    - Loop for each X_i $l = 1, \ldots, n$

      Calculate $M(X_i, X_j)$ for all X_j != X_i such that $X_i \notin Pa(X_j)$;
      Choose $x_1, \ldots, x_{k-l}$ with highest ranking, with $l = |Pa(X_j)|$;
      Set $C_i = Pa(X_j) \cup \{x_1, \ldots, x_{k-1}\}$;
      return $\{C_i\}$;
    - Stopping criteria
      Score-based and Candidate-based criteria
- Example
  If I (A; C) > I (A; D) > I (A; B)



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
    $I(X; Y \mid Z) = \sum_Z P'(Z)\, D_{KL}(P'(X, Y \mid Z) \,||\, P'(X \mid Z)\, P'(Y \mid Z))$
  - Shield score
    $M_{shield}(X_i, X_j \mid B) = I(X_i, X_j \mid Pa(X_i))$
  - Deficiency: It doesn't take into account the cardinality of various variables
- Score Measure
  - Handles random variables with multiple values
  - Chain rule of mutual information
    $I(X_i; X_j \mid Pa(X_i)) = I(X_i; X_j \mid Pa(X_i)) - I(X_i; Pa(X_i))$
  - Shield measure
    $M_{shield}(X_i, X_j \mid B) = I(X_i, X_j \mid Pa(X_i))$
  - Score measure
    $M_{Score}(X_i, X_j \mid B) = Score(X_i, X_j \mid Pa(X_i), D)$

Learning with Small Candidate Sets

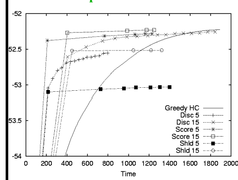
- Maximal Restrict Bayesian Network (MRBN)
  - Input: A set D = $\{X^1, \ldots, X^n\}$ of instances; a digraph H of bounded in-degree K; and a decomposable score S
  - Output: A network B = <G, Θ> so that G ⊆ H, that maximizes S with respect to D
- Standard Heuristics
  - No knowledge of expected structure, local change (e.g. arc deletion, arc addtition, and arc reversal), and local maximum score
  - Algorithms: Greedy hill-climbing; Best-first search; and Simulated annealing
  - Time complexity In Greedy hill climbing is $O(n^2)$ for initial change, then becomes linear O(n) for each iteration
  - Time complexity in MRBN is O(kn) for initial calculation, then becomes O(k)
- Divide and Conquer Heuristics
  - Input: A digraph H = $\{X_j \rightarrow X_i : X_j \in C_i\}$, and a set of weights $w(X_i, Y)$ for each X_i, $Y \in C_i$
  - Output: An acyclic subgraph G ⊆ H that maximizes $W_H[G] = \sum_i w(X_i, Pa(X_i))$
  - Decompose H by using standard graph decomposition methods
  - Find a local maximum weight
  - Combine them into a global solution.

Decomposition

- **Strongly Connected Components: (SCC)**
  - A subset of vertices A is strongly connected if for each X, Y ∈ 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 H1 and H2 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 the decomposition of H for large size clusters

Experimental Evaluation

- Using TABU search to find global max score
- "Alarm" network
  - Samples: 10000
  - variables: 37
  - including 13 have 2 values, 22 have 3 values, and 2 have 4 values
- Text Test
  - Samples: 20 * 1000 sets



Method	Iter	Time	Score	KL	Stats
Greedy		40	-15.35	0.0499	2656
Disc 5	1	14	-18.41	3.0608	908
	2	19	-16.71	1.3634	1063
	3	23	-16.21	0.8704	1183
Disc 10	1	20	-15.53	0.2398	1235
	2	26	-15.43	0.1481	1512
	3	32	-15.43	0.1481	1733
Shld 5	1	14	-17.50	2.1675	915
	2	29	-17.25	1.8905	1728
	3	36	-16.92	1.5632	1907
Shid 10	1	20	-15.86	0.5357	1244
	2	35	-15.50	0.1989	1968
	3	41	-15.50	0.1974	2109
Score 5	1	12	-15.94	0.6756	893
	2	27	-15.34	0.0550	1838
	3	34	-15.33	0.0479	2206
Score 10	1	17	-15.54	0.2559	1169
	2	30	-15.31	0.0352	1917
	3	34	-15.31	0.0352	2058

CIS 830: Advanced Topics in Artificial Intelligence
Kansas State University
Department of Computing and Information Sciences

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.
  - The limited candidate sets provide a small hypothesis space
  - The time complexity of searching the maximum score in BBN is linear
  - It is especially efficient for massive datasets
- **Weaknesses**
  - It doesn't consider the existing of spurious dependency of random variables
  - The search of candidate sets is complex.
  - It is no better for small datasets than standard heuristic algorithms

Presentation Critique

- **Audiences:** Medical diagnosis; Mapping learning; language understanding; Image processing
- **Positive points:** Presents a useful approach in building BBN structure
- **Negative points:** No comparison with other algorithms

CIS 830: Advanced Topics in Artificial Intelligence
Kansas State University
Department of Computing and Information Sciences