Bayesian network

Yuzhen Ye
School of Informatics and Computing
Indiana University, Bloomington
Spring 2013

Bayesian network

Probabilistic graphical models
- Graphical models are a marriage between probability theory and graph theory (Michael Jordan, 1998)
- Graphical models use conditional independence assumptions for efficient representation, inference and learning of joint distributions
  - A compact representation of joint probability distributions;
  - A collection of conditional independence assumptions
- Graphs
  - Nodes: random variables (probabilistic distribution over a fixed alphabet)
  - Edges (arcs), or lack of edges: conditional independence assumptions

Classification of probabilistic graphical models

<table>
<thead>
<tr>
<th>Linear</th>
<th>Branching</th>
<th>Application</th>
</tr>
</thead>
<tbody>
<tr>
<td>Directed</td>
<td>Markov Chain (HMM)</td>
<td>Bayesian network (BN)</td>
</tr>
<tr>
<td></td>
<td>Artificial Intelligence (AI) Statistics</td>
<td></td>
</tr>
<tr>
<td>Undirected</td>
<td>Linear chain conditional random field (CRF)</td>
<td>Markov network (MN)</td>
</tr>
<tr>
<td></td>
<td>Physics (Ising model) Image/Vision</td>
<td></td>
</tr>
</tbody>
</table>

Both directed and undirected arcs: chain graphs

Bayesian Network Structure
- Directed acyclic graph (DAG) G
  - Nodes $X_1, \ldots, X_n$ represent random variables; the parent nodes of $X_i$ ($pa$) represents the set of variables that $X_i$ is dependent on.
  - The variables can be discrete or continuous
- $G$ encodes local Markov assumptions
  - Conditional independence $X_i \perp X_j | X_k$
  - $X_i$ is independent of its non-descendants given its parents
  - The dependencies are modeled by Conditional Probability Distributions (CPDs; for continuous variables) or Conditional Probability Tables (CPTs; for discrete variables)
- $PA(A,B,C,D,E,F,G) = A \rightarrow B \rightarrow C \rightarrow D \rightarrow E \rightarrow F \rightarrow G$
  - 7 CPDs: each for one edge or one source node
- BN computes the joint distribution of all random variables compactly in a factorized way.
- BN is a compact representation of conditional independence assumptions about a high dimension distribution.

Contents
- Probabilistic graphical models overview
- Bayesian network overview
- Probabilistic inference in Bayesian networks
  - Exact solutions: Enumeration & Variable elimination
  - Approximate approaches: Sampling & MCMC
- Learning Bayesian networks
  - Learning parameters (given model structure)
    - No missing data: MLE
    - Missing data: EM & MCMC
  - Learning graph structure (model selection)
    - Bayesian network classifiers
      - Tree augmented NB

Figure 2. Illustration of Model Parameters for Two-Node Bayesian Network

http://www.ploscompbiol.org/article/info:doi/10.1371/journal.pcbi.0030129
Joint probabilities: 143 parameters
Conditional probabilities: 24
A network with 100 nodes, each with 3 possible values: > 10
47 vs 1,800 parameters!

BN provides compact representation of conditional independence

**Joint probabilities**
- Conditional probabilities: 24

**Conditional independence in BNs**

- Types of connections
  - RE-receptor
  - GP-G protein
  - CR-cellular response

- Serial
  - knowing GP makes RE and CR independent (intermediate cause)

- Diverging
  - knowing SI makes IN and RE independent (common cause)

- Converging
  - NOT knowing GP makes RE and SI independent (common effect)

**Why Bayesian network?**

- Combined with Bayesian method, Bayesian Network can offer solutions to a number of challenges
  - Facilitate the combination of domain knowledge and data
  - Handle incomplete data sets (marginalizing over unknown variables by considering all possible values the unknown variables may take, and averaging over them)
  - Offer an efficient and principled approach for avoiding the over fitting of data
  - Learn about causal relationships

**Inference in a Bayesian network**

- Inference in probabilistic models in general asks the following questions: given $P(X_1, X_2, \ldots, X_m)$ and a set of observations $e = \{X_i = x_i, X_j = x_j, \ldots\}$ (or data), compute
  - Marginals: $P(X_i|e)$
  - Probability of evidence: $P(e)$
  - Most probable explanation: $\arg \max_x P(x|e)$

**Approaches to inference**

- **Exact methods**
  - Enumeration
  - Variable elimination
  - Belief propagation in polytrees
  - Clustering / join tree algorithms

- **Approximate methods**
  - Stochastic simulation / sampling methods
  - Markov chain Monte Carlo
  - Genetic algorithms
  - Neural networks
  - Simulated annealing
  - Mean field theory

**A simple example: credit card fraud**

- Conditional Probability Tables (CPTs)
- Joint probabilities: 143 parameters
- Conditional probabilities: 24
- A network with 100 nodes, each with 3 possible values: > 10
47 vs 1,800 parameters!
Inference in the BN

- BN defines the joint probability for all involved random variables
  \[ p(f,a,s,g,j) = p(f) \cdot p(g|f) \cdot p(s|f,a) \cdot p(j|f,a,s) \]
- One can use BN to compute any probability of interest
  - Computing posterior marginal probability, e.g. The probability of fraud, given the evidence (a,s,g,j).
  \[ p(f|a,s,g,j) = \frac{\sum_{f} p(f) \cdot p(g|f) \cdot p(s|f,a) \cdot p(j|f,a,s) \cdot p(a) \cdot p(s) \cdot p(j)}{\sum_{f} p(f) \cdot p(g|f) \cdot p(s|f,a) \cdot p(j|f,a,s) \cdot p(a) \cdot p(s) \cdot p(j)} \]

When variables are in a set of discrete values, this can be computed!

Inference with missing data

- The probability of fraud, but the gender of the cardholder is unknown
  \[ p(f | a,s,g) = \frac{\sum_{a} p(f \mid a) \cdot p(a) \cdot p(s \mid a) \cdot p(g \mid a, s)}{\sum_{a} p(f \mid a) \cdot p(a) \cdot p(s \mid a) \cdot p(g \mid a, s)} \]

Don’t see p(s'): \( p(s = \text{male}) \Rightarrow p(s = \text{female}) \)

\[ p(f, \text{yes} \mid a = 30, g = \text{year}) = \frac{0.00001 \times 0.2 \times (0.05 + 0.05)}{2 \times 10^{-3} + 0.99999 \times 0.01} = 8.94 \times 10^{-6} \]

(marginalizing over variable s)

Inference by variable elimination (VE algorithm)

Consider a query that needs to compute the joint probability of \( X(x_1, x_2, \ldots, x_n) \), where \( x_i \) represents a random variable (i.e., node)

\[ P(X | \epsilon) = \sum_{x_1, \ldots, x_n} \prod_{\epsilon \subseteq x_i, \epsilon \subseteq p(x_i)} P(t_i | pa_{x_i}) \]

where \( \epsilon \) represents a subset of variables outside \( X \), \( pa_{x_i} \) represents the set of parent variables of \( x_i \)

The computation can be accelerated by a Dynamic Programming algorithm, which iteratively

- move all irrelevant terms outside of innermost sum
- perform innermost sum, get a new term
- insert the new term into the product

Inference by enumeration (examples)

- The probability of fraud (age=30, sex=female, gas=yes, jewelry=yes)
  \[ p(f | a,s,g,j) = \frac{\sum_{f} p(f) \cdot p(g|f) \cdot p(s|f,a) \cdot p(j|f,a,s) \cdot p(a) \cdot p(s) \cdot p(j)}{\sum_{f} p(f) \cdot p(g|f) \cdot p(s|f,a) \cdot p(j|f,a,s) \cdot p(a) \cdot p(s) \cdot p(j)} \]

\[ p(f = \text{yes} \mid a = 30, g = \text{yes}, j = \text{yes}) = \frac{0.00001 \times 0.2 \times 0.05}{10^{-3} + 0.99999 \times 0.01 + 0.002} = 0.0005 \]

- The probability that the card holder is female, if the card is not fraud
  \[ p(s = \text{female} \mid a = 30, f = \text{no}, g = \text{yes}, j = \text{yes}) = \frac{0.5 \times 0.0005}{2.5 \times 10^{-6} + 0.5 \times 0.0001} = \frac{1}{5} \]

Computational complexity of inference by enumeration

- Computing the joint probability
  - Multiplication of CPDs, \( O(n^3) \)

\[ p(f) = \sum_{a,s,g,j} \left( p(f) \cdot p(a) \cdot p(s \mid a) \cdot p(g \mid a,s) \cdot p(j \mid f,a,s) \right) \]

To compute the denominator, all combinations of \( f, a, s, g, j \) need to be enumerated, indicating the complexity of \( 2^n \).

Exact inference in an arbitrary BN for discrete variables is NP-hard (Cooper, 1987). When BN contains many undirected cycles (e.g., adding an edge \( a \rightarrow b \) forming a cycle \( f \rightarrow b \rightarrow a \rightarrow f \)), inference is intractable.

Variable elimination: Example

\[ P(d) = \sum_{b,c} p(d \mid b,c) \cdot p(b \mid a) \cdot p(c \mid a) \cdot p(a) \]

\[ = \sum_{b,c} \left( \sum_{d} p(d \mid b,c) \cdot p(b \mid a) \cdot p(c \mid a) \right) \cdot p(a) \]

\[ = \sum_{b,c} p(d \mid b,c) \cdot p(b \mid a) \cdot p(c \mid a) \cdot f(b,c) \]

\( f(b,c) \) can be computed for each combination of \( b,c \), and used for computing \( P(d) \).
A more complex example

Eliminate: $v$

\[ P(v)P(s)P(t | v)P(l | s)P(b | s)P(a | t, l)P(x | a)P(d | a, b) \]

Compute:
\[ f_t(t) = \sum_s f(v)P(s)P(t | v)P(l | s)P(b | s)P(a | t, l)P(x | a)P(d | a, b) \]

Note: let \( f_t(t) = P(t) \)

In general, however, result of elimination is not necessarily a probability term.

Eliminate: $s$

\[ P(v)P(s)P(t | v)P(l | s)P(b | s)P(a | t, l)P(x | a)P(d | a, b) \]

\[ \Rightarrow f_v(t)P(s)P(t | v)P(l | s)P(b | s)P(a | t, l)P(x | a)P(d | a, b) \]

Compute:
\[ f_v(t) = \sum_s P(s)P(t | v)P(l | s)P(b | s)P(a | t, l)P(x | a)P(d | a, b) \]

Summing on $s$ results in a dimensional matrix $f_v(t)$

In general, result of elimination may be a function of several variables.

Eliminate: $x$

\[ P(v)P(s)P(t | v)P(l | s)P(b | s)P(a | t, l)P(x | a)P(d | a, b) \]

\[ \Rightarrow f_v(t)P(s)P(t | v)P(l | s)P(b | s)P(a | t, l)P(x | a)P(d | a, b) \]

\[ \Rightarrow f_v(t)P(s)P(t | v)P(l | s)P(b | s)P(a | t, l)P(x | a)P(d | a, b) \]

Compute:
\[ f_v(t) = \sum_s P(s)P(t | v)P(l | s)P(b | s)P(a | t, l)P(x | a)P(d | a, b) \]

Note: \( f_v(t) \neq 1 \) for all values of $t$

Eliminate: $t$

\[ P(v)P(s)P(t | v)P(l | s)P(b | s)P(a | t, l)P(x | a)P(d | a, b) \]

\[ \Rightarrow f_v(t)P(s)P(t | v)P(l | s)P(b | s)P(a | t, l)P(x | a)P(d | a, b) \]

\[ \Rightarrow f_v(t)P(s)P(t | v)P(l | s)P(b | s)P(a | t, l)P(x | a)P(d | a, b) \]

Compute:
\[ f_v(t) = \sum_t f_v(t)P(t | v)P(l | s)P(b | s)P(a | t, l)P(x | a)P(d | a, b) \]

Eliminate: $l$

\[ P(v)P(s)P(t | v)P(l | s)P(b | s)P(a | t, l)P(x | a)P(d | a, b) \]

\[ \Rightarrow f_v(t)P(s)P(t | v)P(l | s)P(b | s)P(a | t, l)P(x | a)P(d | a, b) \]

\[ \Rightarrow f_v(t)P(s)P(t | v)P(l | s)P(b | s)P(a | t, l)P(x | a)P(d | a, b) \]

\[ \Rightarrow f_v(t)P(s)P(t | v)P(l | s)P(b | s)P(a | t, l)P(x | a)P(d | a, b) \]

Compute:
\[ f_v(t) = \sum_t f_v(t)P(t | v)P(l | s)P(b | s)P(a | t, l)P(x | a)P(d | a, b) \]

Eliminate: $b$

\[ P(v)P(s)P(t | v)P(l | s)P(b | s)P(a | t, l)P(x | a)P(d | a, b) \]

\[ \Rightarrow f_v(t)P(s)P(t | v)P(l | s)P(b | s)P(a | t, l)P(x | a)P(d | a, b) \]

\[ \Rightarrow f_v(t)P(s)P(t | v)P(l | s)P(b | s)P(a | t, l)P(x | a)P(d | a, b) \]

Compute:
\[ f_v(t) = \sum_b f_v(b)P(b | s)P(t | v)P(l | s)P(a | t, l)P(x | a)P(d | a, b) \]

Eliminate: $a$

\[ P(v)P(s)P(t | v)P(l | s)P(b | s)P(a | t, l)P(x | a)P(d | a, b) \]

\[ \Rightarrow f_v(t)P(s)P(t | v)P(l | s)P(b | s)P(a | t, l)P(x | a)P(d | a, b) \]

\[ \Rightarrow f_v(t)P(s)P(t | v)P(l | s)P(b | s)P(a | t, l)P(x | a)P(d | a, b) \]

Compute:
\[ f_v(t) = \sum_a f_v(a)P(a | t, l)P(x | a)P(d | a, b) \]
Eliminate: $a, b$

\[
P(v)P(s)P(r | v)P(t | s)P(b | s)P(a | r, t)P(x | a)P(d | a, b)
\]

\[
\Rightarrow f_1(t)P(s)P(u | s)P(b | s)P(a | r, t)P(x | a)P(d | a, b)
\]

\[
\Rightarrow f_2(r)f_1(b)f_2(a)f_2(t)f_1(x)f_2(d | a, b)
\]

\[
\Rightarrow f_3(a, b)f_2(a)f_2(d | a, b) \Rightarrow f_4(b, d) \Rightarrow f_5(d)
\]

Compute:

\[
f_5(b, d) = \sum f_3(a, b)f_2(a)f_2(d | a, b)
\]

\[
f_5(d) = \sum f_4(b, d)
\]
**Predicting diseases from BN**

![Bayesian Network Diagram]

**Learning Bayesian networks: four cases**

- **Known graph—learn parameters**
  - Complete data (ML, MAP)
  - Incomplete data (EM)
- **Unknown graph—learn graph and parameters**
  - Complete data: optimization problem (search in space of graphs)
  - Incomplete data: structural EM

**Learning parameters: complete data**

The maximum likelihood estimate (MLE) of $\theta_{ij}$ can be computed by a frequency model,

$$\hat{\theta}_{ij} = \frac{N_{ij}}{N_{ii}}$$

where $N_{ij}$ is the number of cases in $D$ in which $X_i = x_i$ and $Pa_i = p_i$. If we assume the prior distribution of $\theta_{ij}$ follow a Dirichlet distribution with parameters $\alpha_{ij} = (\alpha_{i1}, \ldots, \alpha_{ri})$, i.e., the pseudo-counts, we have the MAP estimates,

$$\hat{\theta}_{ijk} = \frac{N_{ijk} + \alpha_{ijk}}{\alpha_{ijk} + \sum_k N_{ijk}}$$

Assumption: the parameters are independent (i.e., $\theta_{ij}$ are mutually independent)

**Learning parameters with missing data**

- Important property of the missing data
  - the absence of the data is dependent on the actual state of the variable
  - e.g., a missing datum in a drug study may indicate that a patient became too sick, perhaps due to the side effects of the drug, to continue in the study.
  - the absence of the data and the state of the variable are independent
- BN can handle both situations; the 2nd one is simpler and will be discussed here.

**An example**

<table>
<thead>
<tr>
<th>Fraud</th>
<th>Age</th>
<th>Sex</th>
<th>Gas</th>
<th>Jewelry</th>
</tr>
</thead>
<tbody>
<tr>
<td>no</td>
<td>35</td>
<td>M</td>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td>no</td>
<td>22</td>
<td>F</td>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td>no</td>
<td>55</td>
<td>M</td>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td>no</td>
<td>42</td>
<td>M</td>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td>no</td>
<td>51</td>
<td>F</td>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td>no</td>
<td>32</td>
<td>F</td>
<td>no</td>
<td>yes</td>
</tr>
<tr>
<td>no</td>
<td>28</td>
<td>F</td>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td>yes</td>
<td>25</td>
<td>M</td>
<td>yes</td>
<td>no</td>
</tr>
<tr>
<td>yes</td>
<td>53</td>
<td>M</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>yes</td>
<td>24</td>
<td>F</td>
<td>yes</td>
<td>yes</td>
</tr>
</tbody>
</table>

Adding pseudo-count 1 for each case

- Sample size is too small

**Learning parameters: missing data**

- Gibbs sampling (MCMC) algorithm
  - Randomly choose an initial state for each of the variables without observations, forming the initial configuration
  - Pick a random variable $x_i$, compute its probability distribution given the states of the other $n-1$ variables
  - Sample a state of variable $x_i$, forming a new configuration
  - Iterate the two previous steps, and record all visited configurations
  - Compute the MLE parameters involving the variables with missing data
Learning parameters: missing data

- EM algorithm: finding a local ML
  - Randomly assign parameters to the distribution involving the variables without observations
  - E-step: using BN inference algorithm to obtain the probability distribution of these variables, given the entire network
  - M-step: update model parameters by using MLE based on the frequencies derived from E-step
  - Iterate between E and M steps until the model converges

Learning graph structure

- Constrained-based structure learning algorithms (dependence analysis and search)
  - Independence test: \( P(X,Y) = P(X)P(Y) \)
- Structure scoring methods (optimization of a scoring function) (scoring and search)
  \[ \hat{G} = \arg \max_{G} \text{Score}(G) \]
- Hybrid methods
  - Constraint-based methods can be more efficient for large samples; the detection of conditional independencies may be sensitive; and may not assign a direction to every edge
  - Score-based approach is generally preferred, esp when dealing with small sample size and noisy data.

Constraint-based methods

- Constraint-based methods focus on identifying conditional independence relationships (i.e., Markov conditions) between variable using observed data; conditional independencies are used to constrain the underlying network structure.
- Typically, hypothesis testing procedures, such as the chi-square test and mutual information test, are first used to remove edges from a fully connected undirected graph based on findings of unconditional independence.
- Then directions are added to edges between nodes according to the d-separation (directed separation) criteria.

Grow-shrink method

- Based on the concept of Markov blanket
  - The Markov blanket of a node in a BN consists of its parents, children, and its children’s other parents.
- The GS algorithm
  - Starts with a variable X and an empty set S. The growing phase adds variables to S if they are dependent on X, conditional on the variables currently in S. In the shrinking phase, variables that are rendered independent of X, based on the current members of S, are then removed from S.
  - Represent S (together with X) as a fully connected, undirected network.
  - Examining triples of variables using the d-separation criteria (e.g., remove spousal links between two nodes Y and Z by looking for a d-separating set around Y and Z, and give directions to edges if conditioning on a middle node creates a dependency.

D-separation criteria

- If X, Y and Z are three disjoint sets of nodes in a BN, then Y is said to d-separate X from Z if and only if Y blocks every path from a node in X to a node in Z.

Score-based methods

- Scoring function
- Search space
- Search strategy

Score(G) – measures how well a model fits the data
Finding the best model is NP-hard optimization; Use heuristic search algorithms instead
Scoring functions

- Likelihood scores
  \[ \max_{\theta}(L(S,\theta)) = \max_{\theta} \{L(S,\theta)|D)\} \]

- Penalized log-likelihood scores
  - BIC Bayesian information criteria
  - MDL Minimum description length

- Bayesian scores
  \[ \max_{\theta}(P(S|D)P(\theta|S)) = \max_{\theta}(L(S,\theta)|D) \]

Search methods

- Most search methods make successive changes of edge linkages to the network, and employ the local criterion to assess the merit of each change.

- One simple heuristic search algorithm is greedy search
  - may be stuck at local minima; can start from multiple initial points
  - global optimization approaches can apply: simulated annealing, best-first search, etc.

Bayesian networks as classifiers

- Two types of nodes: a class node (C) and attribute nodes

- A BN can be used as a classifier that gives the posterior probability distribution of the class node, given attributes X.

  \[ P(C|X, G) = \frac{P(C, X|G)}{P(X|G)} = \frac{P(C, X|G)}{\int P(C, X|\theta, S)\,d\theta} \]

- A NB (Naive Bayes) classifier can be viewed as a BN classifier with a simple structure

Model selection trade-offs

- Naive Bayes = too simple (less parameters, but bad model)
- Unrestricted BN = too complex (possible overfitting + complexity)

Tree-approximation

Extensions to NB classifier

- TAN model: A network with an edge between the class node and each of the attributes (to ensure that all attributes are part of the class variable Markov blanket)
  - Dashed lines, edges required by NB classifier
  - Solid lines, correlation edges between attributes (relax the independence assumption between the attributes)

Ref: Friedman et al, 1997
CL algorithm for constructing a tree BN from data

1. Compute $I_p(X; Y)$ between each pair of variables $i \neq j$, where
$$I_p(X; Y) = \sum p(x) \log \frac{p(x, y)}{p(x)p(y)}$$

is the mutual information function. Roughly speaking, this function measures how much information $Y$ provides about $X$. See Appendix A for a more detailed description of this function.

2. Build a complete undirected graph in which the vertices are the variables in $X$. Annotate the weight of an edge connecting $X_i$ to $X_j$ by $I_p(X_i; X_j)$.

3. Build a minimum weighted spanning tree.

4. Transform the resulting undirected tree to a directed one by choosing a root variable and setting the direction of all edges to be outward from it.

CL proves this procedure finds the tree that maximizes the likelihood given the data $D$.

Ref: Chou and Liu, 1968

Learning of causal relationships

- **Causal Markov condition:**
  - Variable $a$ is a direct cause of variable $b$ if and only if there is a direct edge from $a$ to $b$; then the BN is called a causal graph.
  - Variable $a$ is a cause of variable $b$ (or $b$ is dependent on $a$) if there exists a $d$-connecting path from $a$ to $b$ given evidence $E$ (a set of variables)
    - A path from $a$ to $b$ is $d$-connecting if each interior node $n$ in the path is either
      - Linear or diverging and not a member of $E$; or
      - Converging, and either $n$ or one of its descendants is in $E$.

$d$-separation path

- $a$ is the cause of $b$; but $a$ is not the cause of $b$, given $c$ as evidence
- $a$ is not the cause of $b$; but $a$ is the cause of $b$, given $c$ as evidence
- $b$ is the cause of $a$; $b$ may not be the cause of $a$, if $c$ is given as evidence

Dynamic Bayesian network (DBN)

(Vehicle localization task) A moving car tried to track its current location using the data obtained from a, possibly faulty sensor. The system state can be encoded (very simply) using the:

- Location – the car’s current location
- Velocity – the car’s current velocity
- Weather – the current weather
- Failure – the failure status of the sensor
- Obs – the current observation

We have one such set of variables for every time point $t$. A joint probability distribution over all of these sets defines a probability distribution over trajectories of the car. Using this distribution, we want to ask a variety of queries, such as:

- Given a sequence of observations about the car, where is it now?
  - $P(\text{obs}|\text{obs}^0, ... , \text{obs}^{t-2}, \text{obs}^{t-1})$

- Where is it likely to be in 10 minutes?
  - $P(\text{obs}^{t+10}|\text{obs}^0, ... , \text{obs}^{t-2}, \text{obs}^{t-1})$

- Did it stop at the red light?
  - $P(V^t|\text{obs}^0, ... , \text{obs}^{t-2}, \text{obs}^{t-1})$

Reasoning if the model is given

- Given a sequence of observations about the car, where is it now?
  - $P(\text{obs}|\text{obs}^0, ... , \text{obs}^{t-2}, \text{obs}^{t-1})$

- Where is it likely to be in 10 minutes?
  - $P(\text{obs}^{t+10}|\text{obs}^0, ... , \text{obs}^{t-2}, \text{obs}^{t-1})$

- Did it stop at the red light?
  - $P(V^t|\text{obs}^0, ... , \text{obs}^{t-2}, \text{obs}^{t-1})$
HMM as a 2-DBN

DBN is more general than HMM: 1) the CPD of hidden states can be modeled by a BN, rather than a simple Markov chain; 2) more than one observation variable can be modeled simultaneously (like multivariate HMM).

R packages
- mlbench – machine learning benchmark problems
  - E.g., pima data: PimaIndiansDiabetes2
- bnlearn - an R package for Bayesian network learning and inference (http://www.bnlearn.com/)

Applications of BN

Sparse candidate algorithm:
- Only consider a small number of parent candidates for each gene to constrain the search space of the network

Local probability models:
- Multinomial model
- Linear Gaussian model

General architecture of the magic Bayesian Network.

The system (MAGIC) formally incorporates expert knowledge about relative accuracies of data sources to combine them within a normative framework.

Conditional probability tables for each connection were assessed formally from yeast genetics expert