Agenda

- Bayesian Network & Probabilistic Graphical Model
- Introduction to GeNIe & SMILE
- Bayesian Network Practice with GeNIe
  - Problems
    - Designing
    - Inference with the given model
    - Learning Bayesian Network models from data
  - Example data
    - Alarm Network
    - Pima Indians Diabetes
  - Homework: inference
- Appendix
Bayesian Networks

- A compact representation of a joint probability of variables on the basis of the concept of conditional independence.

- Qualitative part: graph theory
  - Directed acyclic graph
  - Nodes: variables
  - Edges: dependency or influence of a variable on another.

- Quantitative part: probability theory
  - Set of conditional probabilities for all variables

- Naturally handles the problem of complexity and uncertainty.
BN encodes probabilistic relationships among a set of objects or variables

It is useful in that
- dependency encoding among all variables: Modular representation of knowledge.
- can be used for the learning of causal relationships → helpful in understanding a problem domain.
- has both a causal and probabilistic semantics → can naturally combine prior knowledge and data.
- provide an efficient and principled approach for avoiding overfitting data in conjunction with Bayesian statistical methods.
Bayesian Network as a Probabilistic Graphical Model

Probabilistic Graphical model

- undirected graph
  - Markov Random Field

- directed graph
  - Bayesian Networks

Graph + Probability

(C) 2010-2014, SNU Biointelligence Lab, http://bi.snu.ac.kr/
The joint probability is effectively represented as the product of independent factors.

**Markov Random Field**

\[
P(A, B, C, D, E) = \frac{1}{\alpha} \prod \varphi_i(Z_i)
\]

\[
= \frac{1}{\alpha} \varphi_1(A, B) \varphi_2(B, C, D) \varphi_3(C, D, E)
\]

**Bayesian Networks**

\[
P(A, B, C, D, E) = P(A \mid \text{Pa}(A)) P(B \mid \text{Pa}(B)) P(C \mid \text{Pa}(C)) P(D \mid \text{Pa}(D)) P(E \mid \text{Pa}(E))
\]
Real World Applications of BN

- **Intelligent agents**
  - Microsoft Office assistant: Bayesian user modeling

- **Medical diagnosis**
  - PATHFINDER (Heckerman, 1992): diagnosis of lymph node disease ➔ commercialized as INTELLIPATH (http://www.intellipath.com/)

- **Control decision support system**

- **Speech recognition (HMMs)**

- **Genome data analysis**
  - gene expression, DNA sequence, a combined analysis of heterogeneous data.

- **Turbocodes (channel coding)**
Bayesian Network Application Case: Microsoft Office Assistant

Figure 2. A portion of the Lumière Bayesian network user model.

Lumière’s Inference Behind the Scenes

- Probabilities of needs
- Probability that user desires assistance

Figure 3. Lumière’s Inference Behind the Scenes.
GeNIe: http://genie.sis.pitt.edu/

- GeNIe (Graphical Network Interface) is the graphical interface to SMILE, a fully portable Bayesian inference engine
- What you can do with GeNIe are
  - Create and modify network models with a graphical editor
  - Building Bayesian networks, influence diagrams
  - Learning structures and parameters of Bayesian networks from data
  - Probabilistic inference with given Bayesian network

Note: SMILE (Structural Modeling, Inference, and Learning Engine) is a fully portable library of C++ classes implementing graphical decision-theoretic methods
(Look slide 44 for more information)
Dataset #1: Alarm Network

Description
- The network for a medical diagnostic system developed for on-line monitoring of patients in intensive care units
- You will learn how to do inference with a given Bayesian network

Configuration of the data set
- 37 variables, discrete (2~4 levels)
- Variables represent various states of heart, blood vessel and lung
- Three kinds of variables
  - Diagnostic: basis of alarm
  - Measurement: observations
  - Intermediate: states of a patient
Description

- Pima Indians have the highest prevalence of diabetes in the world
- You will learn how to learn structures and parameters of Bayesian networks from data
- We may get possible causal relationship between features that affect diabetes in Pima tribe

Configuration of the data set

- 768 instances
- 8 attributes
  - age, number of times pregnant, results of medical tests/analysis
  - discretized set will be used for BN
- Class value = 1 (Positive example)
  - Interpreted as "tested positive for diabetes"
  - 500 instances
- Class value = 0 (Negative example)
  - 268 instances
Designing Bayesian Network Model

- **TakeHeart II**: Decision support system for clinical cardiovascular risk assessment

1. **Knowledge Engineering**
   - 2 epidemiological models

2. **Data Mining**
   - Busselton Study data
   - Bayesian network software (Netica)
   - Causal discovery (CaMML)

3. **Evaluation**
   - Other learners
   - Medical Experts

Figure 3.6 TakeHeartII Architecture: Construction and Adaptation (left) provide BNs used for risk assessment in a clinical setting (right).
Given an assignment of a subset of variables (evidence) in a BN, estimate the posterior distribution over another subset of unobserved variables of interest.

Inferences viewed as message passing along the network.
Learning Bayesian Networks

Data Acquisition → Preprocessing

BN = Structure + Local probability distribution

Prior knowledge

Bayesian network learning

- Structure Search
- Score Metric
- Parameter Learning
Practice

❖ **Disease-Test Model**
  - Building a Bayesian network manually
  - Inference with the built model

❖ **Alarm Network**
  - Inference (in GeNIe, select following menu)
    - ‘Network – Update Beliefs’, or ‘F5’
    - Network - Probability of Evidence
    - Network - Annealed MAP

❖ **Pima Indians Diabetes**
  - Preprocessing: discretization with Weka
  - Learning Bayesian network from data
    - Structure learning
    - Parameter learning
  - Analyzing the resulting model
Dataset for Practice with GeNle

❖ Disease-Test Model
  ■ Given on the coming slides

❖ Alarm Network
  ■ data_Alarm_modified.xdsl

❖ Pima Indians Diabetes
  ■ discretization with Weka: pima_diabetes.arff
    (result: pima_diabetes_supervised_discretized.csv)
  ■ Learning Bayesian network from data:
    ■ pima_diabetes_supervised_discretized.csv
Practice 1: Building a Bayesian Network
Practice 1: Building a Bayesian Network
Practice 1: Building a Bayesian Network
Practice 1: Building a Bayesian Network

Node properties: Node1

<table>
<thead>
<tr>
<th>General</th>
<th>Definition</th>
<th>Format</th>
<th>User properties</th>
</tr>
</thead>
<tbody>
<tr>
<td>Add</td>
<td>Insert</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

| State0   | 0.583333338 |
| State1   | 0.416666667 |

(C) 2010-2014, SNU Biointelligence Lab, http://bi.snu.ac.kr/
Practice 1: Building a Bayesian Network

![Image of a Bayesian Network software interface]

(C) 2010-2014, SNU Biointelligence Lab, http://bi.snu.ac.kr/
Practice 1: Building a Bayesian Network
Practice 1: Disease-Test Model

\[
P(\text{Test}=\text{negative} \mid \text{Disease}=\text{absent}) = 0.95 \\
P(\text{Test}=\text{negative} \mid \text{Disease}=\text{present}) = 0.02 \\
P(\text{Test}=\text{positive} \mid \text{Disease}=\text{absent}) = 0.05 \\
P(\text{Test}=\text{positive} \mid \text{Disease}=\text{present}) = 0.98
\]
Inference

- Set evidence
- Update belief
- Check updated probability
Practice 2: Alarm Network

: diagnostic node  : measurement node  : intermediate node
Practice 2: Alarm Network

- **Inference tasks**
  - Set evidences (according to observations or sensors)
  - ‘Network – Update Beliefs’, or ‘F5’
Practice 2: Alarm Network

- Inference tasks
  - Network - Probability of Evidence

The Alarm network has been developed for online monitoring of patients.
Practice 2: Alarm Network

- **Inference tasks**
  - Based on a set of observed nodes
    - we can estimate the most probable states of target nodes
    - We can calculate the probability of this configuration
  - Network - Annealed MAP
Practice 3: Learning from Data

- **Pima Indians Diabetes data**
  - **Step 1**: discretization of real-valued features with Weka
    1. Open ‘pima_diabetes.arff’
    2. Apply ‘Filter-Supervised-Attribute-Discretize’ with default setting
    3. Save into ‘pima_diabetes_supervised_discretized.csv’
Practice 3: Learning from Data

Pima Indians Diabetes data

Step 2: Learning structure of the Bayesian network

1. File-Open Data File: pima_diabetes_supervised_discretized.csv
2. Data-Learn New Network
3. Set parameters as in Fig. 1
4. Edit the resulting graph: changing position, color

Fig. 1 Parameter setting

Fig. 2 Learned structure

(C) 2010-2014, SNU Biointelligence Lab, http://bi.snu.ac.kr/
Practice 3: Learning from Data

- **Pima Indians Diabetes data**
  - **Step 3: Learning parameters of the Bayesian network**
    1. Check the default parameters (based on counts in the data)
      1. ‘F8’ key will show distributions for all the nodes as a bar chart
      2. ‘F5’ key will show you the probability
    2. Network – Learn Parameters
    3. Just click ‘OK’ button for each dialogue box
    4. Check the change of the parameters with ‘F5’ key
Approach the problem as an optimization problem
Use a scoring metric to measure how well a particular structure fits the observed set of cases

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td>S1</td>
<td>H</td>
<td>L</td>
<td>L</td>
<td>L</td>
</tr>
<tr>
<td>S2</td>
<td>H</td>
<td>H</td>
<td>H</td>
<td>H</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>SM</td>
<td>L</td>
<td>H</td>
<td>H</td>
<td>L</td>
</tr>
</tbody>
</table>

Scoring metric
Search strategy

Score
Task: Given a network structure, estimate the parameters of the model from data.
What is the probability that it is raining, given the grass is wet?
Homework 2 (inference)

- Q1) \( p(U|R,Q,S) = ? \)
- Q2) \( p(P|Q) = ? \)
- Q3) \( p(Q|P) = ? \)

- First, you may try to calculate by hand
- Next, you can check the answer with GeNIe
Appendix
JOINT PROBABILITY

- Key concept: two or more random variables may interact. Thus, the probability of one taking on a certain value depends on which value(s) the others are taking.
- We call this a joint ensemble and write

\[ p(x, y) = \text{prob}(X = x \text{ and } Y = y) \]
Marginal Probabilities

- We can "sum out" part of a joint distribution to get the marginal distribution of a subset of variables:

\[ p(x) = \sum_y p(x, y) \]

- This is like adding slices of the table together.

- Another equivalent definition: \[ p(x) = \sum_y p(x | y)p(y) \].

slide from the lecture ‘Probabilistic Graphical Model’ by Sam Roweis in MLSS ‘05
**Conditional Probability**

- If we know that some event has occurred, it changes our belief about the probability of other events.
- This is like taking a "slice" through the joint table.

\[
p(x|y) = \frac{p(x, y)}{p(y)}
\]

slide from the lecture ‘Probabilistic Graphical Model’ by Sam Roweis in MLSS ‘05
Joint Probabilities

- Goal 1: represent a joint distribution $P(X) = P(x_1, x_2, \ldots, x_n)$ compactly even when there are many variables.
- Goal 2: efficiently calculate marginal and conditionals of such compactly represented joint distributions.
- Notice: for $n$ discrete variables of arity $k$, the naive (table) representation is HUGE: it requires $k^n$ entries.
- We need to make some assumptions about the distribution. One simple assumption: independence $\iff$ complete factorization: $P(X) = \prod_i P(x_i)$
- But the independence assumption is too restrictive. So we make conditional independence assumptions instead.
The Joint Distribution

Recipe for making a joint distribution of M variables:

1. Make a truth table listing all combinations of values of your variables (if there are M Boolean variables then the table will have $2^M$ rows).
2. For each combination of values, say how probable it is.
3. If you subscribe to the axioms of probability, those numbers must sum to 1.

Example: Boolean variables A, B, C

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>Prob</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.30</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0.05</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0.10</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0.05</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.05</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0.10</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0.25</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0.10</td>
</tr>
</tbody>
</table>
Main Issues in BN

**Inference** in Bayesian networks

- Given an assignment of a subset of variables (evidence) in a BN, estimate the posterior distribution over another subset of unobserved variables of interest.

\[ P(x_{un} \mid x_{obs}) = \frac{P(x_{un}, x_{obs})}{P(x_{obs})} \]

**Learning** Bayesian network **from data**

- **Parameter Learning**
  - Given a data set, estimate local probability distributions 
    \[ P(X_i \mid \text{Pa}(X_i)) \], for all variables (nodes) comprising the BN.

- **Structure learning**
  - For a data set, search a network structure \( G \) (dependency structure) which is best or at least plausible.
Naïve Bayes Classifier

Tree-Augmented Naïve Bayes Classifier (TAN)
**GeNIe and SMILE**

A developer’s environment for graphical decision models ([http://genie.sis.pitt.edu/](http://genie.sis.pitt.edu/)).

- **Model developer module**: GeNIe.Implemented in Visual C++ in Windows environment.

- **Wrappers**: SMILE.NET, jSMILE, Pocket SMILE

  Allow SMILE to be accessed from applications other than C++ compiler

- **Learning and discovery module**: SMiLe

- **Qualitative interface**: QGeNIe

- **Support for model building**: ImaGeNIe

- **Diagnosis**: Diagnose

- **Reasoning engine**: SMiLe (Structural Modeling, Inference, and Learning Engine).

  A platform independent library of C++ classes for graphical models.

(C) 2010-2014, SNU Biointelligence Lab, [http://bi.snu.ac.kr/](http://bi.snu.ac.kr/)
Both graphical model

- Specify a **factorization** (how to express the joint distribution)
- Define a set of **conditional independence** properties

- Chain graphs: graphs that include both directed and undirected links
Many probabilistic models can be represented as Bayesian networks

- Hidden Markov Models
- Kalman Filters / Particle Filters
- PCA, ICA, ...
- Hierarchical Bayes models
- Etc.

\[ x_1 \rightarrow x_{t-1} \rightarrow x_t \rightarrow x_{t+1} \rightarrow x_T \]

\[ o_1 \rightarrow o_{t-1} \rightarrow o_t \rightarrow o_{t+1} \rightarrow o_T \]