Outline

• Bayesian networks and queries
• Building Bayesian Networks
  • Medical diagnosis
  • Circuit diagnosis
  • Probabilistic decoding
  • Commonsense reasoning
  • Linkage analysis
• Special representations of CPTs
  • Causal independence (noisy-or, noisy-and)
  • Decision trees
Outline

- Bayesian networks and queries
  - Building Bayesian Networks
    - Medical diagnosis
    - Circuit diagnosis
    - Probabilistic decoding
    - Commonsense reasoning
    - Linkage analysis
  - Special representations of CPTs
    - Causal independence (noisy-or, noisy-and)
    - Decision trees
Real-world applications, drawn from the domains of diagnosis, reliability, genetics, channel coding, and commonsense reasoning.

Specific reasoning problems which can be addressed by posing a formal query with respect to a Bayesian network.

Constructing the required network.

Identifying the specific queries that need to be applied.
The construction of a Bayesian network involves three major steps:

- Identify relevant variables and their possible values.
- Build the network structure by connecting variables into DAG.
- Define the CPT for each network variable.

Two issues:

- The potentially large size of CPTs.
- The significance of the specific numbers used to populate them.

We present techniques for dealing with these issues.

Queries: Different queries may be relevant for different scenarios
Reasoning with Bayesian Networks

The network Asia will be used as a running example. Screenshot from Samlam.

http://reasoning.cs.ucla.edu/samiam


For other tools (e.g., GeNie/Smile) see class page
Probability of some variable instantiation $e$, $P_r(e)$.

Probability that the patient has a positive X-ray, but no dyspnoea, $P_r(X = \text{yes}, D = \text{no})$, about 3.96%. Computed by Samlam.

The variables $E = \{X, D\}$ are called evidence variables. The query $P_r(e)$ is known as a probability-of-evidence.

Other type of evidence: We may want to know the probability that the patient has either a positive X-ray or dyspnoea, $X = \text{yes}$ or $D = \text{yes}$. 
Auxiliary-node method

Bayesian network tools do not usually provide direct support for computing the probability of arbitrary pieces of evidence, but such probabilities can be computed indirectly.

We can add an auxiliary node $E$, declare nodes $X$ and $D$ as the parents of $E$, and use the following CPT for $E$:

| $X$ | $D$ | $E$ | $\Pr(e|x, d)$ |
|-----|-----|-----|-------------|
| yes | yes | yes | 1           |
| yes | no  | yes | 1           |
| no  | yes | yes | 1           |
| no  | no  | yes | 0           |

Event $E = \text{yes}$ is then equivalent to $X = \text{yes} \lor D = \text{yes}$. 
Query: Prior and Posterior Marginals

**Prior Marginals**

Given a joint probability distribution \( \Pr(x_1, \ldots, x_n) \), the marginal distribution \( \Pr(x_1, \ldots, x_m) \), \( m \leq n \), is defined as follows:

\[
\Pr(x_1, \ldots, x_m) = \sum_{x_{m+1}, \ldots, x_n} \Pr(x_1, \ldots, x_n).
\]

The marginal distribution can be viewed as a projection of the joint distribution on the smaller set of variables \( X_1, \ldots, X_m \).

**Posterior marginal given evidence \( e \)**

\[
\Pr(x_1, \ldots, x_m|e) = \sum_{x_{m+1}, \ldots, x_n} \Pr(x_1, \ldots, x_n|e).
\]
Prior Marginals in the Asia Network

C = lung cancer

<table>
<thead>
<tr>
<th>C</th>
<th>Pr(C)</th>
</tr>
</thead>
<tbody>
<tr>
<td>yes</td>
<td>5.50%</td>
</tr>
<tr>
<td>no</td>
<td>94.50%</td>
</tr>
</tbody>
</table>
Query: Posterior Marginals in the Asia Network

| C   | \( \Pr(C|e) \) |
|-----|-----------------|
| yes | 25.23%          |
| no  | 74.77%          |

e : \( X = \text{yes}, D = \text{no} \)
Query: Most Probable Explanation (MPE)

Let $X_1, \ldots, X_n$ be all network variables, and $e$ be evidence. Identify an instantiation $x_1, \ldots, x_n$ that maximizes the probability $\Pr(x_1, \ldots, x_n|e)$. Instantiation $x_1, \ldots, x_n$ is called a most probable explanation given evidence $e$.

MPE cannot be obtained directly from posterior marginals.

If $x_1, \ldots, x_n$ is an instantiation obtained by choosing each value $x_i$ so as to maximize the probability $\Pr(x_i|e)$, then $x_1, \ldots, x_n$ is not necessarily an MPE.
Query: Most Probable Explanation (MPE)

MPE is also called MAP

MPE given a positive X-ray and dyspnoea

A patient that made no visit to Asia; is a smoker; has lung cancer and bronchitis; but no tuberculosis.

slides4 COMPSCI 2021
Query: Most Probable Explanation (MPE)

Choosing values with maximal probability, we get:
\[ \alpha: A = \text{no}, \ S = \text{yes}, \ T = \text{no}, \ C = \text{no}, \ B = \text{no}, \ P = \text{no}, \ X = \text{yes}, \ D = \text{no}. \]

Probability \( \approx 20.03\% \) given evidence \( e: X = \text{yes}, \ D = \text{no} \).

MPE given a positive X-ray and no dyspnoea (\( \approx 38.57\% \))

A patient that made no visit to Asia; is not a smoker; has no lung cancer, no bronchitis and no tuberculosis.
MAP is also called Marginal Map (MMAP)
A common method for approximating MAP is to compute an MPE and then return the values it assigns to MAP variables. We say in this case that we are projecting the MPE on MAP variables.
A common method for approximating MAP is to compute an MPE and then return the values it assigns to MAP variables. We say in this case that we are projecting the MPE on MAP variables.

**Example**

MPE given evidence $X = \text{yes}$, $D = \text{no}$:

$$A = \text{no}, \ S = \text{no}, \ T = \text{no}, \ C = \text{no}, \ B = \text{no}, \ P = \text{no}, \ X = \text{yes}, \ D = \text{no}$$

Projecting this MPE on MAP variables $\mathbf{M} = \{A, S\}$, we get:

$$A = \text{no}, \ S = \text{no},$$

with probability $\approx 48.09\%$ given the evidence.

MAP is $A = \text{no}, \ S = \text{yes}$ with a probability of about $50.74\%$. 

slides4 COMPSCI 2021
## Probabilistic Reasoning Problems

### Tasks:

<table>
<thead>
<tr>
<th>Inference Type</th>
<th>Mathematical Formulation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Max-Inference</td>
<td>$f(x^*) = \max_x \prod_{\alpha} f_\alpha(x_{\alpha})$</td>
</tr>
<tr>
<td>(most likely config, MPE.)</td>
<td></td>
</tr>
<tr>
<td>Sum-Inference</td>
<td>$Z = \sum_x \prod_{\alpha} f_\alpha(x_{\alpha})$</td>
</tr>
<tr>
<td>(data likelihood, $P(\text{evidence})$)</td>
<td></td>
</tr>
<tr>
<td>Mixed-Inference</td>
<td>$f(x^*<em>M) = \max</em>{x^M} \sum_{x_S} \prod_{\alpha} f_\alpha(x_{\alpha})$</td>
</tr>
<tr>
<td>(optimal prediction, MAP, Marginal Map)</td>
<td></td>
</tr>
</tbody>
</table>

- **Harder**

Combinatorial search / counting queries

Exact reasoning NP-complete (or worse)
Probabilistic Reasoning Problems

Exact Algorithm: Bucket Elimination, Complexity $\epsilon^{\text{tree-width}}$

- **Max-Inference**  
  (most likely config.)  
  \[ f(x^*) = \max_x \prod_{\alpha} f_{\alpha}(x_{\alpha}) \]

- **Sum-Inference**  
  (data likelihood)  
  \[ Z = \sum_x \prod_{\alpha} f_{\alpha}(x_{\alpha}) \]

- **Mixed-Inference**  
  (optimal prediction)  
  \[ f(x_M^*) = \max_{x_M} \sum_{x_S} \prod_{\alpha} f_{\alpha}(x_{\alpha}) \]

- **Mixed-Inference**  
  (maximum expected utility)  
  \[ \text{MEU} = \max_{\Delta} \mathbb{E}_{P(X,D)} [\sum_{U_i \in U} U_i] \]
Bayesian networks will be constructed in three consecutive steps.

**Step 1**

Define the network variables and their values.

- **A query variable** is one which we need to ask questions about, such as compute its posterior marginal.
- **An evidence variable** is one which we may need to assert evidence about.
- **An intermediary variable** is neither query nor evidence and is meant to aid the modeling process by detailing the relationship between evidence and query variables.

The distinction between query, evidence and intermediary variables is not a property of the Bayesian network, but of the task at hand.
Bayesian networks will be constructed in three consecutive steps.

**Step 2**

Define the network structure (edges).

We will be guided by a causal interpretation of network structure.

The determination of network structure will be reduced to answering the following question about each network variable $X$: what set of variables we regard as the direct causes of $X$?

What about the boundary strata?
Constructing a Bayesian Network for any Distribution \( P \)

COROLLARY 3: Given a probability distribution \( P(x_1, x_2, ..., x_n) \) and any ordering \( d \) of the variables, the DAG created by designating as parents of \( X_i \) any minimal set \( \Pi_{X_i} \) of predecessors satisfying

\[
P(x_i \mid n_{X_i}) = P(x_i \mid x_1, ..., x_{i-1}), \quad \Pi_{X_i} \subseteq \{X_1, X_2, ..., X_{i-1}\}
\]

is a Bayesian network of \( P \).

- If \( P \) is strictly positive, then all of the parent sets are unique (see Theorem 4) and the Bayesian network is unique (given \( d \)).

COROLLARY 4: Given a DAG \( D \) and a probability distribution \( P \), a necessary and sufficient condition for \( D \) to be a Bayesian network of \( P \) is that each variable \( X \) be conditionally independent of all its non-descendants, given its parents \( \Pi_X \), and that no proper subset of \( \Pi_X \) satisfy this condition.

Intuition: The causes of \( X \) can serve as the parents
Ask: who does a variable listen to
Modeling with Bayesian Networks

Step 3
Define the network CPTs.

- CPTs can sometimes be determined completely from the problem statement by objective considerations.
- CPTs can be a reflection of subjective beliefs.
- CPTs can be estimated from data.
Outline

- Bayesian networks and queries
- Building Bayesian Networks
  - Medical diagnosis
  - Circuit diagnosis
  - Probabilistic decoding
  - Commonsense reasoning
  - Linkage analysis
- Special representations of CPTs
  - Causal independence (noisy-or, noisy-and)
  - Decision trees
Diagnosis I: Model from Expert

Example

The flu is an acute disease characterized by fever, body aches and pains, and can be associated with chilling and a sore throat. The cold is a bodily disorder popularly associated with chilling and can cause a sore throat. Tonsillitis is inflammation of the tonsils which leads to a sore throat and can be associated with fever.

Our goal here is to develop a Bayesian network to capture this knowledge and then use it to diagnose the condition of a patient suffering from some of the symptoms mentioned above.

Variables? Arcs? Try it.

slides4 COMPSCI 2021
A naive Bayes structure has the following edges $C \rightarrow A_1, \ldots, C \rightarrow A_m$, where $C$ is called the class variable and $A_1; \ldots; A_m$ are called the attributes.

Variables are binary: values are either true or false. More refined information may suggest different degrees of body ache.
The naive Bayes structure commits to the single-fault assumption.

Suppose the patient is known to have a cold.

**Naive Bayes structure**
Fever and sore throat become independent as they are d-separated by “Condition”.

**Original structure**
Fever may increase our belief in tonsillitis, which could then increase our belief in a sore throat.
CPTs can be obtained from medical experts, who supply this information based on known medical statistics or subjective beliefs gained through practical experience.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>true</td>
<td>false</td>
<td>?</td>
<td>true</td>
<td>false</td>
<td>false</td>
<td>false</td>
</tr>
<tr>
<td>2</td>
<td>false</td>
<td>true</td>
<td>false</td>
<td>true</td>
<td>true</td>
<td>false</td>
<td>true</td>
</tr>
<tr>
<td>3</td>
<td>?</td>
<td>?</td>
<td>true</td>
<td>false</td>
<td>?</td>
<td>true</td>
<td>false</td>
</tr>
</tbody>
</table>

? indicates the unavailability of corresponding data for that patient.
Tools for Bayesian network inference can generate a network parameterization $\Theta$, which tries to maximize the probability of seeing the given cases.

If each case is represented by event $d_i$, such tools will generate a parametrization $\Theta$ which leads to a probability distribution $Pr$ that attempts to maximize:

$$\prod_{i=1}^{N} Pr(d_i).$$

Term $Pr(d_i)$ represents the probability of seeing the case $i$.

The product represents the probability of seeing all $N$ cases (assuming the cases are independent).
Diagnosis II: Model from Expert

Example

A few weeks after inseminating a cow, we have three possible tests to confirm pregnancy. The first is a scanning test which has a false positive of 1% and a false negative of 10%. The second is a blood test, which detects progesterone with a false positive of 10% and a false negative of 30%. The third test is a urine test, which also detects progesterone with a false positive of 10% and a false negative of 20%. The probability of a detectable progesterone level is 90% given pregnancy, and 1% given no pregnancy. The probability that insemination will impregnate a cow is 87%.

Our task here is to build a Bayesian network and use it to compute the probability of pregnancy given the results of some of these pregnancy tests.

Try it: Variables and values? Structure? CPTs?
Diagnosis II: Model from Expert

Try with GeNie/Smile

| $P$ | $\theta_P$ | $P$ | $S$ | $\theta_{S|P}$ | $P$ | $L$ | $\theta_{L|P}$ |
|-----|------------|-----|-----|----------------|-----|-----|----------------|
| yes | .87        | yes | −ve | .10            | yes | undetectable | .10 |
|     |            | no  | +ve | .01            | no  | detectable   | .01 |

| $L$  | $B$       | $\theta_{B|L}$ | $L$  | $U$       | $\theta_{U|L}$ |
|------|-----------|----------------|------|-----------|----------------|
| detectable | −ve | .30           | detectable | −ve | .20          |
| undetectable| +ve | .10           | undetectable| +ve | .10          |
Example

We inseminate a cow, wait for a few weeks, and then perform the three tests which all come out negative:

\[ e: \ S = -ve, \ B = -ve, \ U = -ve. \]

Posterior marginal for pregnancy given this evidence:

|   | \( \Pr(P|e) \) |
|---|----------------|
| yes | 10.21%        |
| no  | 89.79%        |

Probability of pregnancy is reduced from 87% to 10.21%, but still relatively high given that all three tests came out negative.

slides4 COMPSCI 2021
Sensitivity Analysis

Example

A farmer is not too happy with this and would like three negative tests to drop the probability of pregnancy to no more than 5%. The farmer is willing to replace the test kits for this purpose, but needs to know the false positive and negative rates of the new tests, which would ensure the above constraint.

This is a problem of sensitivity analysis in which we try to understand the relationship between the parameters of a Bayesian network and the conclusions drawn based on the network.

Read in the book.
We will not cover this. Also about level of granularity
Diagnosis III: Model from Design

Problem statement

Given some values for the circuit primary inputs and output (test vector), decide if the circuit is behaving normally. If not, find the most likely health states of its components.

Try it: Variables? Values? Structure?
Diagnosis III: Model from Design

Problem statement
Given some values for the circuit primary inputs and output (test vector), decide if the circuit is behaving normally. If not, find the most likely health states of its components.

Evidence variables
Primary inputs and output of the circuit, A, B and E.
Diagnosis III: Model from Design

Problem statement
Given some values for the circuit primary inputs and output (test vector), decide if the circuit is behaving normally. If not, find the most likely health states of its components.

Evidence variables
Primary inputs and output of the circuit, A, B and E.

Query variables
Health of components X, Y and Z.
Diagnosis III: Model from Design

Problem statement
Given some values for the circuit primary inputs and output (test vector), decide if the circuit is behaving normally. If not, find the most likely health states of its components.

Evidence variables
Primary inputs and output of the circuit, $A$, $B$ and $E$.

Query variables
Health of components $X$, $Y$ and $Z$.

Intermediary variables
Internal wires, $C$ and $D$. 
Diagnosis III: Model from Design

Values of circuit wires: low or high

Health states: ok or faulty
faulty is too vague as a component may fail in a number of modes.

- stuck-at-zero fault: low output regardless of gate inputs.
- stuck-at-one fault: high output regardless of gate inputs.
- input-output-short fault: inverter shorts input to its output.

Fault modes demand more when specifying the CPTs.
Three classes of CPTs

- primary inputs \((A, B)\)
- gate outputs \((C, D, E)\)
- component health \((X, Y, Z)\)

CPTs for health variables depend on their values


<table>
<thead>
<tr>
<th>(X)</th>
<th>(\theta_X)</th>
<th>(X)</th>
<th>(\theta_X)</th>
</tr>
</thead>
<tbody>
<tr>
<td>ok</td>
<td>.99</td>
<td>ok</td>
<td>.99</td>
</tr>
<tr>
<td>faulty</td>
<td>.01</td>
<td>stuckat0</td>
<td>.005</td>
</tr>
<tr>
<td></td>
<td></td>
<td>stuckat1</td>
<td>.005</td>
</tr>
</tbody>
</table>

Need to know the probabilities of various fault modes.
CPTs for component outputs determined from functionality.

| $A$  | $X$   | $C$  | $\theta_{C|a,x}$ |
|------|-------|------|------------------|
| high | ok    | high | 0                |
| low  | ok    | high | 1                |
| high | stuckat0 | high | 0               |
| low  | stuckat0 | high | 0               |
| high | stuckat1 | high | 1               |
| low  | stuckat1 | high | 1               |
CPTs for component outputs determined from functionality.

### Example

CPT for inverter $X$.  

| $A$     | $X$  | $C$  | $\theta_{c|a,x}$ |
|---------|------|------|------------------|
| high    | ok   | high | 0                |
| low     | ok   | high | 1                |
| high    | stuckat0 | high | 0                |
| low     | stuckat0 | high | 0                |
| high    | stuckat1 | high | 1                |
| low     | stuckat1 | high | 1                |

If we do not represent health states:

| $A$     | $X$  | $C$  | $\theta_{c|a,x}$ |
|---------|------|------|------------------|
| high    | ok   | high | 0                |
| low     | ok   | high | 1                |
| high    | faulty | high | ?                |
| low     | faulty | high | ?                |

Common to use a probability of .50 in this case.
Example

Given test vector $\mathbf{e}$: $A = \text{high}$, $B = \text{high}$, $E = \text{low}$, compute MAP over health variables $X$, $Y$ and $Z$. 

A Diagnosis Example

Example

Given test vector $e$: $A=$ high, $B=$ high, $E=$ low, compute MAP over health variables $X$, $Y$ and $Z$.

Network with fault modes gives two MAP instantiations:

<table>
<thead>
<tr>
<th>MAP given $e$</th>
<th>$X$</th>
<th>$Y$</th>
<th>$Z$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>ok</td>
<td>stuckat0</td>
<td>ok</td>
</tr>
<tr>
<td></td>
<td>ok</td>
<td>ok</td>
<td>stuckat0</td>
</tr>
</tbody>
</table>

Each probability $\approx 49.4\%$
A Diagnosis Example

Example
Given test vector $\mathbf{e}$: $A=\text{high}$, $B=\text{high}$, $E=\text{low}$, compute MAP over health variables $X$, $Y$ and $Z$.

Network with fault modes gives two MAP instantiations:

<table>
<thead>
<tr>
<th>MAP given $\mathbf{e}$</th>
<th>$X$</th>
<th>$Y$</th>
<th>$Z$</th>
<th>each probability $\approx 49.4%$</th>
</tr>
</thead>
<tbody>
<tr>
<td>ok</td>
<td>stuckat0</td>
<td>ok</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ok</td>
<td>ok</td>
<td>stuckat0</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Network with no fault modes gives two MAP instantiations:

<table>
<thead>
<tr>
<th>MAP given $\mathbf{e}$</th>
<th>$X$</th>
<th>$Y$</th>
<th>$Z$</th>
<th>each probability $\approx 49.4%$</th>
</tr>
</thead>
<tbody>
<tr>
<td>ok</td>
<td>faulty</td>
<td>ok</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ok</td>
<td>ok</td>
<td>faulty</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Integrating Time

Suppose we have two test vectors instead of only one.
Integrating Time

Suppose we have two test vectors instead of only one.

Additional evidence variables

$A'$, $B'$ and $E'$
Suppose we have two test vectors instead of only one.

**Additional evidence variables**

$A'$, $B'$ and $E'$

**Additional intermediary variables**

$C'$ and $D'$
Integrating Time

Suppose we have two test vectors instead of only one.

Additional evidence variables

A', B' and E'

Additional intermediary variables

C' and D'

Additional health variables on whether we allow intermittent faults

If health of a component can change from one test to another, we need additional health variables X', Y', and Z'. Otherwise, the original health variables are sufficient.
Integrating Time: No Intermittent Faults

Two test vectors

$e: A = \text{high}, B = \text{high}, E = \text{low}$

$e': A = \text{low}, B = \text{low}, E = \text{low}.$
Integrating Time: No Intermittent Faults

Two test vectors
\[ \mathbf{e} : A = \text{high, } B = \text{high, } E = \text{low} \]
\[ \mathbf{e'} : A = \text{low, } B = \text{low, } E = \text{low}. \]

MAP using second structure

<table>
<thead>
<tr>
<th>MAP given ( \mathbf{e}, \mathbf{e'} )</th>
<th>( X )</th>
<th>( Y )</th>
<th>( Z )</th>
</tr>
</thead>
<tbody>
<tr>
<td>ok</td>
<td>ok</td>
<td>faulty</td>
<td></td>
</tr>
</tbody>
</table>

with probability \( \approx 97.53\% \)
Integrating Time: Intermittent Faults

**Dynamic Bayesian network (DBN)**

Two test vectors

- $e$: $A = \text{high}$, $B = \text{high}$, $E = \text{low}$
- $e'$: $A = \text{low}$, $B = \text{low}$, $E = \text{low}$.

**Persistence model for the health of component $X$**

| $X$   | $X'$     | $\theta_{X'|X}$ | Notes                                      |
|-------|----------|------------------|--------------------------------------------|
| ok    | ok       | .99              |                                            |
| ok    | faulty   | .01              | healthy component becomes faulty           |
| faulty| ok       | .001             | faulty component becomes healthy           |
| faulty| faulty   | .999             |                                            |

slides4 COMPSCI 2021
Channel Coding

Four bits $U_1, U_2, U_3$ and $U_4$ are sent from a source $S$ to a destination $D$

over a noisy channel, where there is a 1% chance that a bit will be inverted before it gets to the destination.
Channel Coding

Four bits $U_1$, $U_2$, $U_3$ and $U_4$ are sent from a source $S$ to a destination $D$ over a noisy channel, where there is a 1% chance that a bit will be inverted before it gets to the destination.

To improve the reliability of this process, we will add three redundant bits $X_1$, $X_2$ and $X_3$ to the message, where $X_1$ is the XOR of $U_1$ and $U_3$, $X_2$ is the XOR of $U_2$ and $U_4$, and $X_3$ is the XOR of $U_1$ and $U_4$. 
Channel Coding

Four bits $U_1, U_2, U_3$ and $U_4$ are sent from a source $S$ to a destination $D$ over a noisy channel, where there is a 1% chance that a bit will be inverted before it gets to the destination.

To improve the reliability of this process we will add three redundant bits $X_1, X_2$ and $X_3$ to the message, where $X_1$ is the XOR of $U_1$ and $U_3$, $X_2$ is the XOR of $U_2$ and $U_4$, and $X_3$ is the XOR of $U_1$ and $U_4$.

Given that we received a message containing seven bits at destination $D$ our goal is to restore the message generated at the source $S$.

Try it: Variables, values, structure?
In channel coding terminology

$U_1, \ldots, U_4$ are known as information bits;
$X_1, \ldots, X_3$ are known as redundant bits;
$U_1, \ldots, U_4, X_1, \ldots, X_3$ is known as the code word or channel input;
$Y_1, \ldots, Y_7$ is known as the channel output.
Channel Coding

In channel coding terminology

- \( U_1, \ldots, U_4 \) are known as information bits;
- \( X_1, \ldots, X_3 \) are known as redundant bits;
- \( U_1, \ldots, U_4, X_1, \ldots, X_3 \) is known as the code word or channel input;
- \( Y_1, \ldots, Y_7 \) is known as the channel output.

Goal to restore the channel input given some channel output.
# Channel Coding

**In channel coding terminology**

- \( U_1, \ldots, U_4 \) are known as **information bits**;
- \( X_1, \ldots, X_3 \) are known as **redundant bits**;
- \( U_1, \ldots, U_4, X_1, \ldots, X_3 \) is known as the **code word or channel input**;
- \( Y_1, \ldots, Y_7 \) is known as the **channel output**.

**Goal to restore the channel input given some channel output.**

**Evidence variables are**

- \( Y_1, \ldots, Y_7 \): bits received at destination \( D \)
Channel Coding

In channel coding terminology

- $U_1, \ldots, U_4$ are known as information bits;
- $X_1, \ldots, X_3$ are known as redundant bits;
- $U_1, \ldots, U_4, X_1, \ldots, X_3$ is known as the code word or channel input;
- $Y_1, \ldots, Y_7$ is known as the channel output.

Goal to restore the channel input given some channel output.

Evidence variables are

- $Y_1, \ldots, Y_7$: bits received at destination $D$

Query variables are

- $U_1, \ldots, U_4$: bits originating at source $S$
In channel coding terminology

- $U_1, \ldots, U_4$ are known as information bits;
- $X_1, \ldots, X_3$ are known as redundant bits;
- $U_1, \ldots, U_4, X_1, \ldots, X_3$ is known as the code word or channel input;
- $Y_1, \ldots, Y_7$ is known as the channel output.

Goal to restore the channel input given some channel output.

Evidence variables are

- $Y_1, \ldots, Y_7$: bits received at destination $D$

Query variables are

- $U_1, \ldots, U_4$: bits originating at source $S$

Bits $X_1, \ldots, X_3$ either query variables or intermediary variables.
There are three CPT types in the problem.
Channel Coding

There are three CPT types in the problem.

CPT for each redundant bit, say $X_1$:

$$P_r(x_1|u_1, u_3) = 1 \text{ iff } x_1 = u_1 \oplus u_3 \quad (\oplus \text{ is the XOR function})$$
What queries should we use here?

P(Y not equal U) = 0.01
MAP or Posterior-Marginal (PM) Decoders?

To restore the channel input given channel output

1. Compute a **MAP** for the channel input $U_1, \ldots, U_4, X_1, \ldots, X_3$ given channel output $Y_1, \ldots, Y_7$.

2. Compute the **PM** for each bit $U_i/X_i$ in the channel input, given channel output $Y_1, \ldots, Y_7$, and then select the value of $U_i/X_i$ which is most probable.
MAP or Posterior-Marginal (PM) Decoders?

To restore the channel input given channel output

1. Compute a **MAP** for the channel input $U_1, \ldots, U_4, X_1, \ldots, X_3$ given channel output $Y_1, \ldots, Y_7$.
2. Compute the **PM** for each bit $U_i/X_i$ in the channel input, given channel output $Y_1, \ldots, Y_7$, and then select the value of $U_i/X_i$ which is most probable.

The choice between MAP and PM decoders is a matter of the performance measure one is interested in optimizing.

WER (word error rate), BER (bit error rate)

MAP (MPE) minimizes WER, PM minimize BER…
What do you think?
A more realistic and common noise model

Transmitting our code bits $x_i$ through a channel that adds Gaussian noise, with mean $x_i$ and standard deviation $\sigma$.

Channel output $Y_i$ is a continuous variable governed by

conditional density function $f(y_i|x_i) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-(y_i-x_i)^2/2\sigma^2}$
Convolutional and turbo codes correspond to different methods for generating redundant bits.
Convolutional Codes

Convolutional and turbo codes correspond to different methods for generating redundant bits.

Convolutional and turbo codes provide examples of modeling systems with feedback loops using dynamic Bayesian networks.
Convolutional Codes

Dynamic Bayesian network for a convolutional code.

A sequence of replicated slices where slice $k$ is responsible for generating the codeword bits $x_{2k}$ and $x_{2k+1}$ for the information bit $u_k$. 
Convolutional Codes

A sequence of replicated slices where slice $k$ is responsible for generating the codeword bits $x_{2k}$ and $x_{2k+1}$ for the information bit $u_k$.

Each slice has a variable $S_k$ representing the state of the encoder. This state is determined by the previous state variable $S_{k-1}$ and the information bit $U_k$. 
Given four information bits $u_0, \ldots, u_3$. 
Given four information bits $u_0, \ldots, u_3$.

In a convolutional code, we generate 4 redundant bits leading to an 8-bit codeword.
Given four information bits $u_0, \ldots, u_3$. 

**In a convolutional code** we generate 4 redundant bits leading to an 8-bit codeword.

**In a turbo code we apply a convolutional code twice** once on the original bit sequence $u_0, u_1, u_2, u_3$, and another on some permutation, say, $u_1, u_3, u_2, u_0$. This leads to 8 redundant bits and a 12-bit codeword.
Lower network represents a convolutional code for the bit sequence \( u_0, \ldots, u_3 \).

Upper network represents a convolutional code for the bit sequence \( u_4, \ldots, u_7 \).
The excitement about probabilistic decoding in the 90’s
And the rise of belief propagation
Task (PM for each bit)

Edges that cross between the networks are meant to establish the bit sequence $u_4, \ldots, u_7$ (upper network) as a permutation of the bit sequence $u_0, \ldots, u_3$ (lower network).
Commonsense reasoning

When SamBot goes home at night, he wants to know if his family is home before he tries the doors.

Often when SamBot's wife leaves the house she turns on an outdoor light. However, she sometimes turns on this light if she is expecting a guest.

Also, SamBot's family has a dog. When nobody is home, the dog is in the back yard. The same is true if the dog has bowel trouble.

If the dog is in the back yard, SamBot will probably hear her barking, but sometimes he can be confused by other dogs barking.

SamBot is equipped with two sensors: a light-sensor for detecting outdoor lights and a sound-sensor for detecting the barking of dogs. Both of these sensors are not completely reliable and can break. Moreover, they both require SamBot's battery to be in good condition.
Genetic Linkage Analysis

A pedigree

is useful in reasoning about heritable characteristics which are determined by genes, where different genes are responsible for the expression of different characteristics.
Genetic Linkage Analysis

A pedigree is useful in reasoning about heritable characteristics which are determined by genes, where different genes are responsible for the expression of different characteristics.

A gene may occur in different states called alleles. Each individual carries two alleles of each gene, one received from their mother and the other from their father. The alleles of an individual are called the genotype, while the heritable characteristic expressed by these alleles (such as hair color, blood type, etc) are called the phenotype of the individual.
Recombination Events

**Haplotype**

The alleles received by an individual from one parent. Each individual has two haplotypes, one paternal and another maternal.

Gene $G_1$ has alleles $A$ and $a$.
Gene $G_2$ has alleles $B$ and $b$. 
If $G_1$ and $G_2$ are close then they are likely to pass down from the same haplotype (grandmother or grandfather)

- Mary can pass only one haplotype to her child Jack: $AB$.
- John can pass only one haplotype to Jack: $ab$.
- Jack can pass one of four haplotypes to his children: $AB, Ab, aB, ab$. 

Recombination Events
Genetic Linkage and Gene Maps

If two genes are inherited independently
the probability of a recombination is expected to be 1/2.

Genetic linkage
Two alleles which were passed in the haplotype from a grandparent
to a parent tend to be passed again in the same haplotype from
the parent to a child.

Goal of genetic linkage analysis
is to estimate the extent to which two genes are linked.
Genetic Linkage and Gene Maps

The extent to which genes $G_1$ and $G_2$ are linked is measured by a recombination fraction or frequency, $\theta$, which is the probability that a recombination between $G_1$ and $G_2$ will occur.

Genes that are inherited independently are characterized by a recombination frequency $\theta = 1/2$ and are said to be unlinked. Linked genes on the other hand are characterized by a recombination frequency $\theta < 1/2$. 
Two Loci Inheritance

Recombinant
Bayesian Network for Recombination

\[ P(s_{23t} \mid s_{13t}, \theta) = \begin{bmatrix} 1 - \theta & \theta \\ \theta & 1 - \theta \end{bmatrix} \]  
where \( t \in \{m,f\} \)
Linkage analysis:
6 people, 3 markers

slides4 COMP861 2021
Outline

• Bayesian networks and queries
• Building Bayesian Networks
• Special representations of CPTs
  • Causal Independence (e.g., Noisy OR)
  • Context Specific Independence
  • Determinism
  • Mixed Networks
Dealing with Large CPTs

The size of a CPT
for binary variable $E$ with binary parents $C_1, \ldots, C_n$

<table>
<thead>
<tr>
<th>Number of Parents: $n$</th>
<th>Parameter Count: $2^n$</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td>3</td>
<td>8</td>
</tr>
<tr>
<td>6</td>
<td>64</td>
</tr>
<tr>
<td>10</td>
<td>1024</td>
</tr>
<tr>
<td>20</td>
<td>1,048,576</td>
</tr>
<tr>
<td>30</td>
<td>1,073,741,824</td>
</tr>
</tbody>
</table>

slides4 COMPSCI 2021
Think about headache and 10 different conditions that may cause it.

A noisy-or circuit

A micro model details the relationship between a variable $E$ and its parents $C_1, \ldots, C_n$.

We wish to specify cpt with less parameters
Binary OR

|   |   | P(X=0|A,B) | P(X=1|A,B) |
|---|---|---------|---------|
| 0 | 0 | 1       | 0       |
| 0 | 1 | 0       | 1       |
| 1 | 0 | 0       | 1       |
| 1 | 1 | 0       | 1       |
Noisy-or Model

- Cause $C_i$ is capable of establishing effect $E$, except under some unusual circumstances summarized by suppressor $Q_i$.
- When suppressor $Q_i$ is active, $C_i$ is no longer able to establish $E$.
- The leak variable $L$ represents all other causes of $E$ which were not modeled explicitly.
- When none of the causes $C_i$ are active, the effect $E$ may still be established by the leak variable $L$. 
Noisy-or Model

The noisy-or model requires $n + 1$ parameters.
Noisy–or Model

The noisy-or model requires $n + 1$ parameters.

To model the relationship between headache and ten different conditions:

- $\theta_{q_i} = \Pr(Q_i = \text{active})$: probability that suppressor of $C_i$ is active.
- $\theta_l = \Pr(L = \text{active})$: probability that leak is active.
Noisy-or Model

- Let $l_\alpha$ be the indices of causes that are active in $\alpha$. 
Noisy-or Model

- Let $l_\alpha$ be the indices of causes that are active in $\alpha$.
- If

$$\alpha: C_1 = \text{active}, C_2 = \text{active}, C_3 = \text{passive}, C_4 = \text{passive}, C_5 = \text{active},$$

then $l_\alpha = \{1, 2, 5\}$. 
Let \( l_\alpha \) be the indices of causes that are active in \( \alpha \).

If

\[
\alpha: C_1 = \text{active}, C_2 = \text{active}, C_3 = \text{passive}, C_4 = \text{passive}, C_5 = \text{active},
\]

then \( l_\alpha = \{1, 2, 5\} \).

We then have

\[
\Pr(E = \text{passive} | \alpha) = (1 - \theta_1) \prod_{i \in l_\alpha} \theta_{q_i}
\]

\[
\Pr(E = \text{active} | \alpha) = 1 - \Pr(E = \text{passive} | \alpha).
\]
Noisy-or Model

- Let $l_\alpha$ be the indices of causes that are active in $\alpha$.
- If

  \[ \alpha: C_1 = \text{active}, \ C_2 = \text{active}, \ C_3 = \text{passive}, \ C_4 = \text{passive}, \ C_5 = \text{active}, \]

  then $l_\alpha = \{1, 2, 5\}$.
- We then have

  \[
  \Pr(E = \text{passive} | \alpha) = \left(1 - \theta_1\right) \prod_{i \in l_\alpha} \theta_{q_i}
  \]

  \[
  \Pr(E = \text{active} | \alpha) = 1 - \Pr(E = \text{passive} | \alpha).
  \]

The full CPT for variable $E$, with its $2^n$ parameters, can be induced from the $n + 1$ parameters of the noisy-or model.
Noisy-or Model

Example

Sore throat \((S)\) has three causes: cold \((C)\), flu \((F)\), tonsillitis \((T)\).
Noisy-or Model

Example
Sore throat \((S)\) has three causes: cold \((C)\), flu \((F)\), tonsillitis \((T)\).

If we assume that \(S\) is related to its causes by a noisy-or model, we can then specify the CPT for \(S\) by the following four probabilities:

- The suppressor probability for cold, say \(0.15\)
- The suppressor probability for flu, say \(0.01\)
- The suppressor probability for tonsillitis, say \(0.05\)
- The leak probability, say \(0.02\)
Noisy-or Model

Example
Sore throat \( (S) \) has three causes: cold \( (C) \), flu \( (F) \), tonsillitis \( (T) \).
Noisy-or Model

Example

Sore throat ($S$) has three causes: cold ($C$), flu ($F$), tonsillitis ($T$).

The CPT for sore throat is then determined completely as follows:

| $C$  | $F$  | $T$  | $S$  | $\theta_{s|c,f,t}$  | $1 - (1 - .02)(.15)(.01)(.05)$ |
|------|------|------|------|----------------------|----------------------------------|
| true | true | true | true | 0.9999265            |                                  |
| true | true | false| true | 0.99853              |                                  |
| true | false| true | true | 0.99265              |                                  |
| :    | :    | :    | :    | :                   |                                  |
| false| false| false| true | .02                 | $1 - (1 - .02)$                  |

slides4 COMPSCI 2021
Noisy/OR CPDs

Figure 11: the CPDS network for diagnosis of internal diseases. The network contains 448 nodes, 906 links.
Decision Trees

<table>
<thead>
<tr>
<th>C1</th>
<th>C2</th>
<th>C3</th>
<th>C4</th>
<th>Pr(E=1)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0.0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0.0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0.0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0.0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0.0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0.0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0.0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0.9</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0.9</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0.9</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0.9</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0.3</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0.3</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0.6</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.8</td>
</tr>
</tbody>
</table>
If-Then Rules

A CPT for variable $E$ can be represented using a set of if-then rules of the form

If $\alpha_i$ then $\Pr(e) = p_i$, for each value $e$ of variable $E$, where $\alpha_i$ is a propositional sentence constructed using the parents of variable $E$. 
### If-Then Rules

A CPT for variable $E$ can be represented using a set of if-then rules of the form

If $\alpha_i$ then $\Pr(e) = p_i$, for each value $e$ of variable $E$, where $\alpha_i$ is a propositional sentence constructed using the parents of variable $E$.

<table>
<thead>
<tr>
<th>Condition</th>
<th>Probability ( \Pr(E=1) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>$C_1 = 1$</td>
<td>$0.0$</td>
</tr>
<tr>
<td>$C_1 = 0 \land C_2 = 1$</td>
<td>$0.9$</td>
</tr>
<tr>
<td>$C_1 = 0 \land C_2 = 0 \land C_3 = 1$</td>
<td>$0.3$</td>
</tr>
<tr>
<td>$C_1 = 0 \land C_2 = 0 \land C_3 = 0 \land C_4 = 1$</td>
<td>$0.6$</td>
</tr>
<tr>
<td>$C_1 = 0 \land C_2 = 0 \land C_3 = 0 \land C_4 = 0$</td>
<td>$0.8$</td>
</tr>
</tbody>
</table>
A CPT for variable $E$ can be represented using a set of if-then rules of the form

If $\alpha_i$ then $Pr(e) = p_i$, for each value $e$ of variable $E$, where $\alpha_i$ is a propositional sentence constructed using the parents of variable $E$.

For the rule-based representation to be complete and consistent

- The premises $\alpha_i$ must be mutually exclusive. That is, $\alpha_i \land \alpha_j$ is inconsistent for $i \neq j$. This ensures that the rules will not conflict with each other.

- The premises $\alpha_i$ must be exhaustive. That is, $\bigvee_i \alpha_i$ must be valid. This ensures that every CPT parameter $\theta_{e|...}$ is implied by the rules.
A student’s example

Difficulty → Intelligence

Grade ← Intelligence

SAT ← Difficulty

Apply ← Grade

Letter ← Apply

Job ← Letter
Tree CPD

If the student does not Apply, SAT and L are irrelevant

Tree-CPD for job
Captures irrelevant variables

[Diagram with nodes and edges labeled with indices and coordinates, showing a tree structure with nodes named C, L1, L2, Choice, Letter1, and Letter2, and connections indicating the flow of data or decisions.]
A CPD $P(Y|A,Z_1,Z_2,\ldots,Z_k)$ is a multiplexer iff $\text{Val}(A)=1,2,\ldots,k$, and 
$P(Y|A,Z_1,\ldots,Z_k)=Z_a$
Deterministic CPTs

Can we use hidden variables?

| A     | X    | C    | $\theta_{c|a,x}$ |
|-------|------|------|------------------|
| high  | ok   | high | 0                |
| low   | ok   | high | 1                |
| high  | stuckat0 | high | 0                |
| low   | stuckat0 | high | 0                |
| high  | stuckat1 | high | 1                |
| low   | stuckat1 | high | 1                |

We can represent this CPT as follows:

$$(X = \text{ok} \land A = \text{high}) \lor X = \text{stuckat0} \iff C = \text{low}$$

$$(X = \text{ok} \land A = \text{low}) \lor X = \text{stuckat1} \iff C = \text{high}$$
Mixed Networks
(Dechter 2013)

Augmenting Probabilistic networks with constraints because:

- Some information in the world is deterministic and undirected \((X \neq Y)\)
- Some queries are complex or evidence are complex (cnfs)

Queries are probabilistic queries
Probabilistic Reasoning

Party example: the weather effect

Alex is **likely** to go in bad weather
Chris **rarely** goes in bad weather
Becky is indifferent but **unpredictable**

Questions:

Given bad weather, which group of individuals is most likely to show up at the party?

What is the probability that Chris goes to the party but Becky does not?

\[
P(W,A,C,B) = P(B|W) \cdot P(C|W) \cdot P(A|W) \cdot P(W)
\]

\[
P(A,C,B|W=\text{bad}) = 0.9 \cdot 0.1 \cdot 0.5
\]
Party Example Again

Bayes Network

Constraint Network

Query:
Is it likely that Chris goes to the party if Becky does not but the weather is bad?

\[ P(C, \neg B \mid w = \text{bad}, A \rightarrow B, C \rightarrow A) \]