Reasoning with Graphical Models

Slides Set 3:
Building Bayesian Networks

Rina Dechter

Darwiche chapters 5

Outline

- 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.

Outline

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.

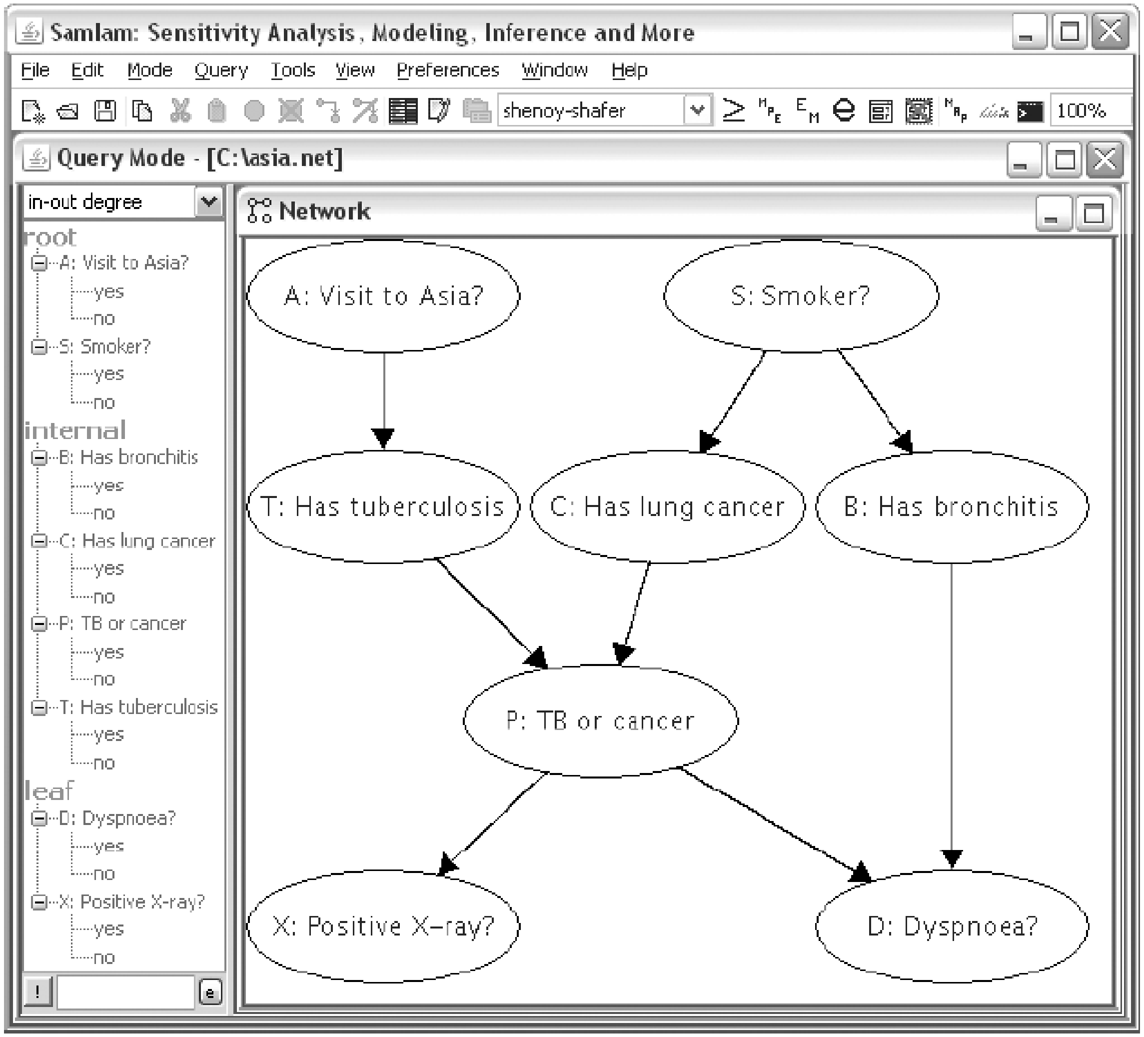
Two issues:

- 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 SamIam.

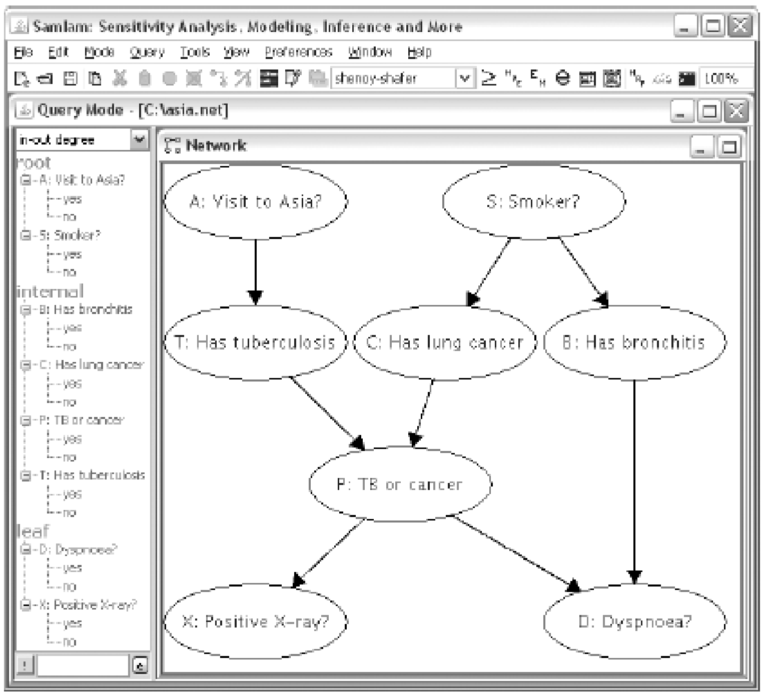
http://reasoning.cs.ucla.edu/samiam

SamIam available at http://reasoning.cs.ucla.edu/samiam/.

For other tools (e.g., GeNie/Smile) see class page

Query: Probability of Evidence

Probability of some variable instantiation $\mathbf{e}$, $\Pr(\mathbf{e})$.



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

The variables $\mathbf{E} = \{X, D\}$ are called evidence variables. The query $\Pr(\mathbf{e})$ 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 =yes **or** D=yes.

Query: Probability of Evidence

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	$\Pr(e \mid x, d)$
yes	yes	yes	1
yes	no	yes	1
no	yes	yes	1
no	no	yes	0

Event $E = \text{yes}$ is then equivalent to $X = \text{yes} \vee D = \text{yes}$.

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:

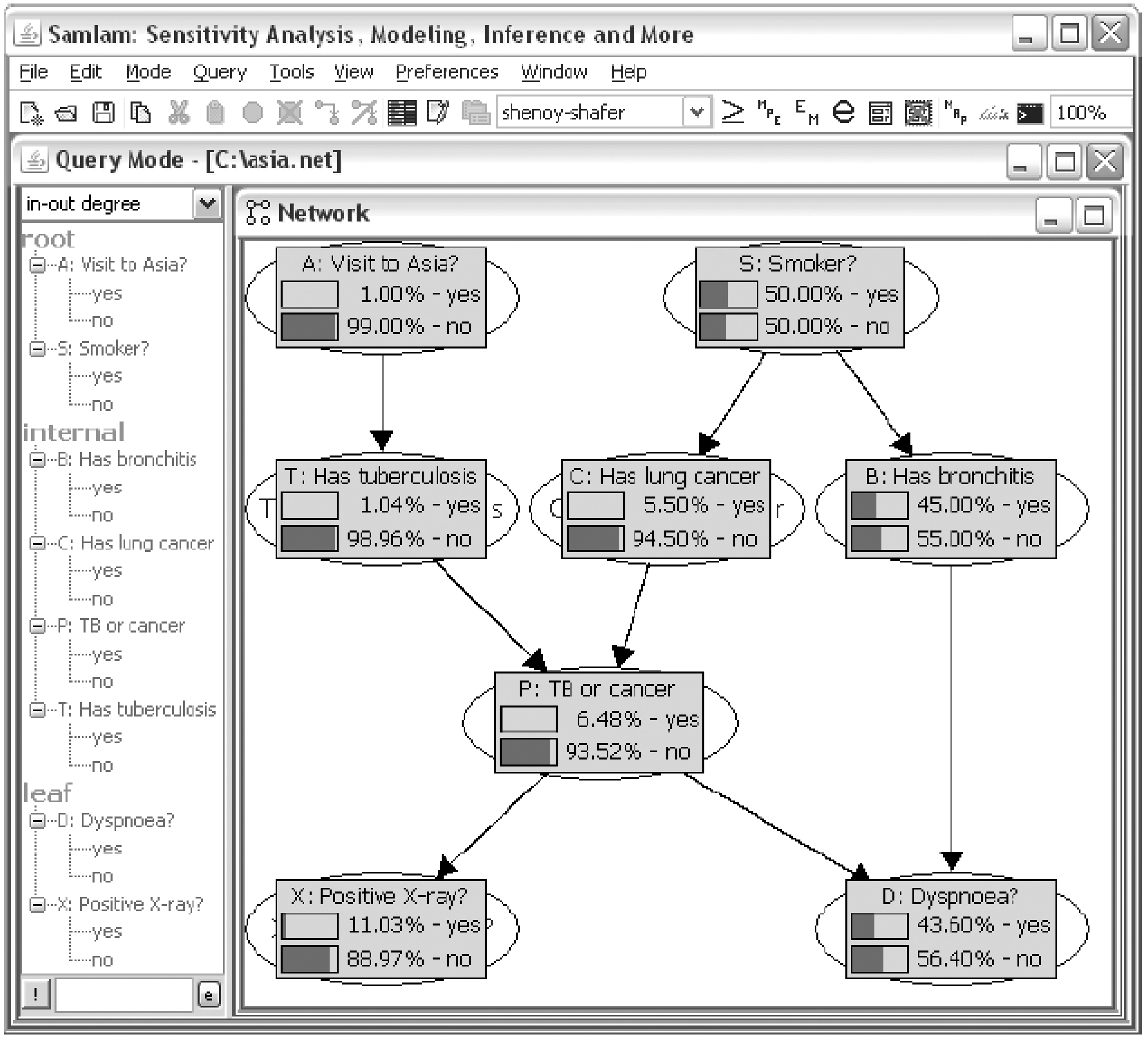
$$\Pr(x_1, \ldots, x_m) = \sum_{x_{m+1}, \ldots, x_n} \Pr(x_1, \ldots, x_n).$$

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

$\Pr(x_1, \ldots, x_m | \mathbf{e}) = \sum_{x_{m+1}, \ldots, x_n} \Pr(x_1, \ldots, x_n | \mathbf{e})$.

Prior Marginals in the Asia Network

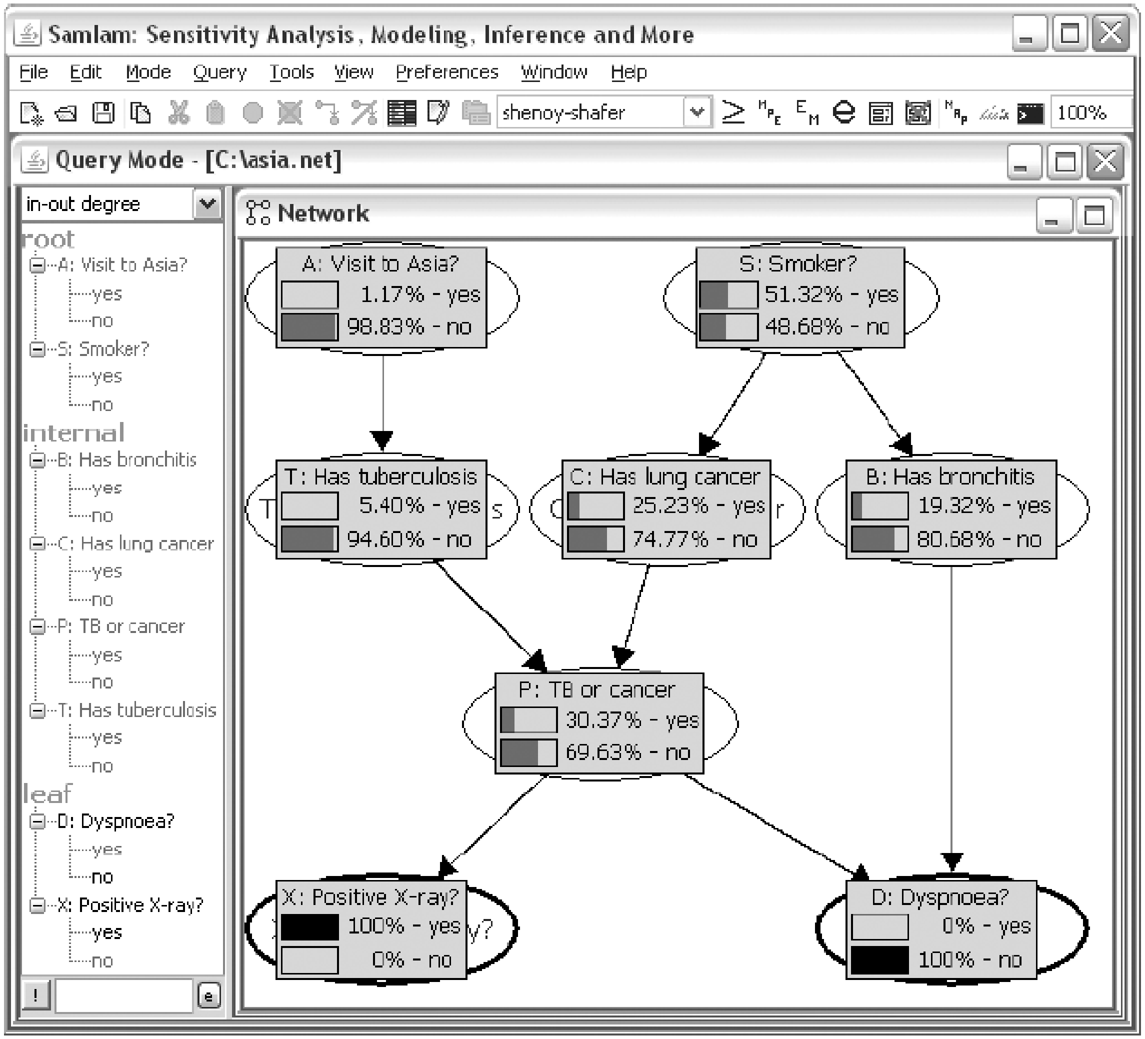


C= lung cancer

Prior marginal

C	$\Pr(C)$
yes	5.50%
no	94.50%

Query: Posterior Marginals in the Asia Network

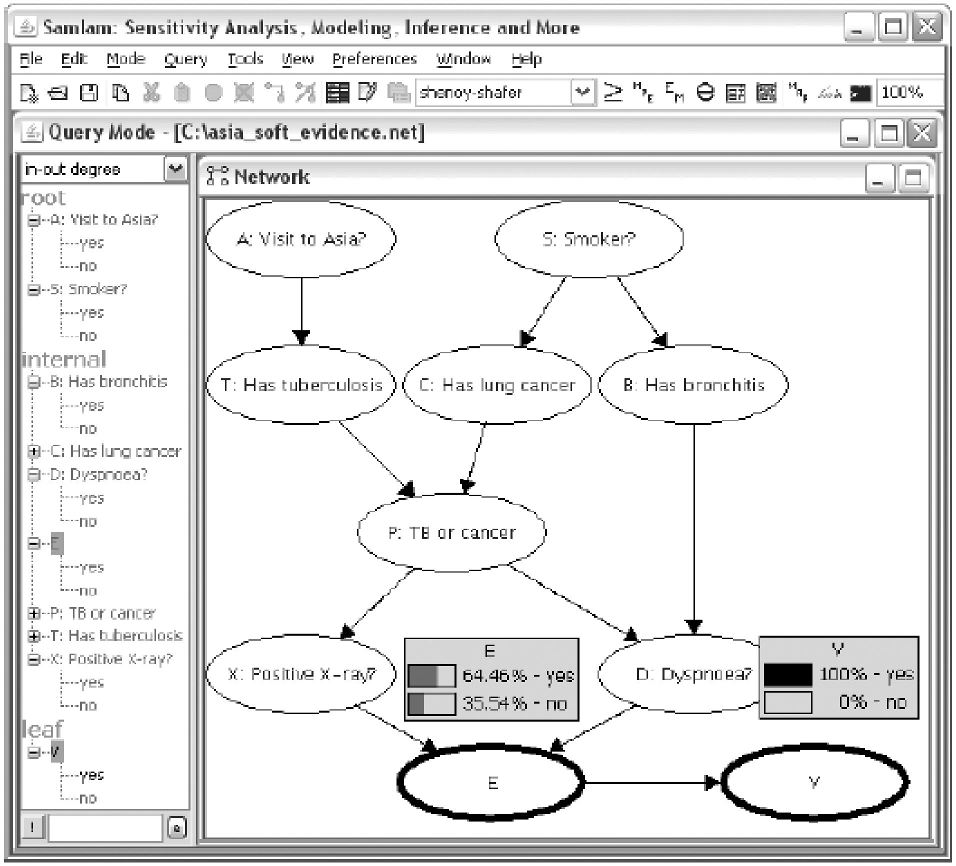


Posterior marginal

C	$\Pr(C \mid \mathbf{e})$
yes	25.23%
no	74.77%

$\mathbf{e}: X = \text{yes}, D = \text{no}$

Soft Evidence using Virtual Evidence (Noisy Sensor)



Soft evidence of Positive x-ray or Dyspnoea (X=yes or D = yes) with odds of 2 to 1.

Modelling: Add E variable and Add V to model soft evidence.

$$\frac{P(V=yes|E=yes)}{P(V=yes|E=no)} = 2$$

Define a CPT for V that satisfies this constraint

Soft evidence on E as hard evidence on auxiliary variable V.

Query: Most Probable Explanation (MPE)

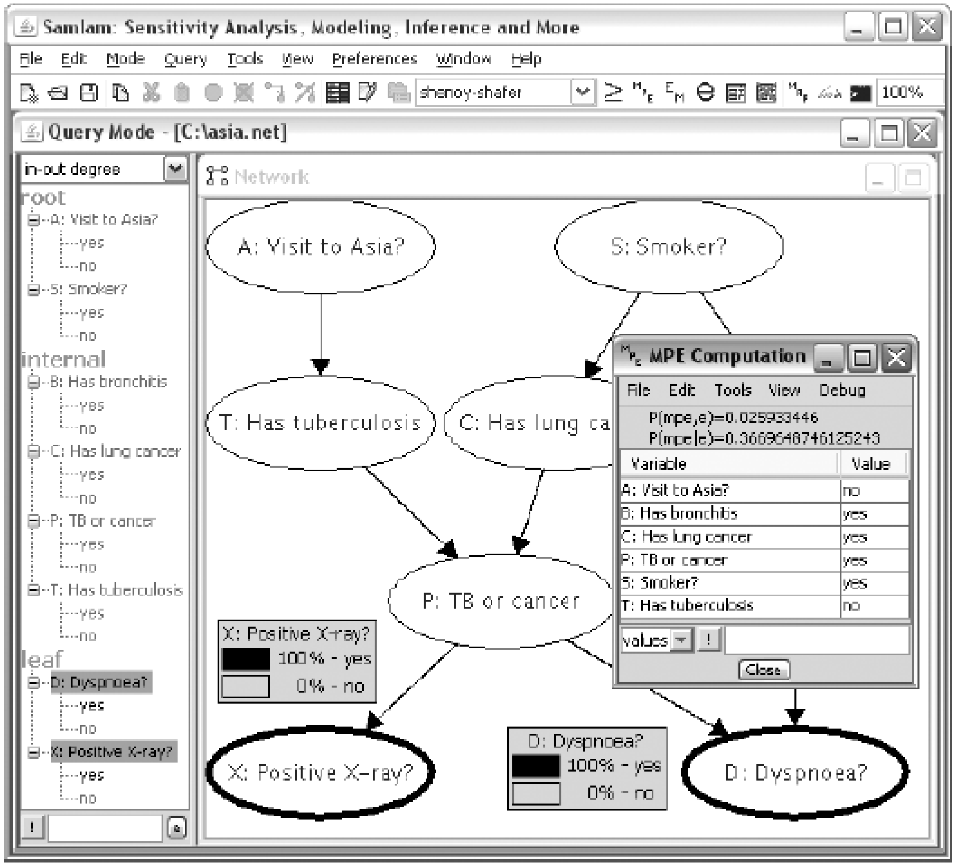
Let $X_1, \ldots, X_n$ be all network variables, and $\mathbf{e}$ be evidence. Identify an instantiation $x_1, \ldots, x_n$ that maximizes the probability $\Pr(x_1, \ldots, x_n | \mathbf{e})$. Instantiation $x_1, \ldots, x_n$ is called a most probable explanation given evidence $\mathbf{e}$.

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 $\Pr(x_i | \mathbf{e})$, then $x_1, \ldots, x_n$ is not necessarily an MPE.

MPE is also called MAP

Query: Most Probable Explanation (MPE)

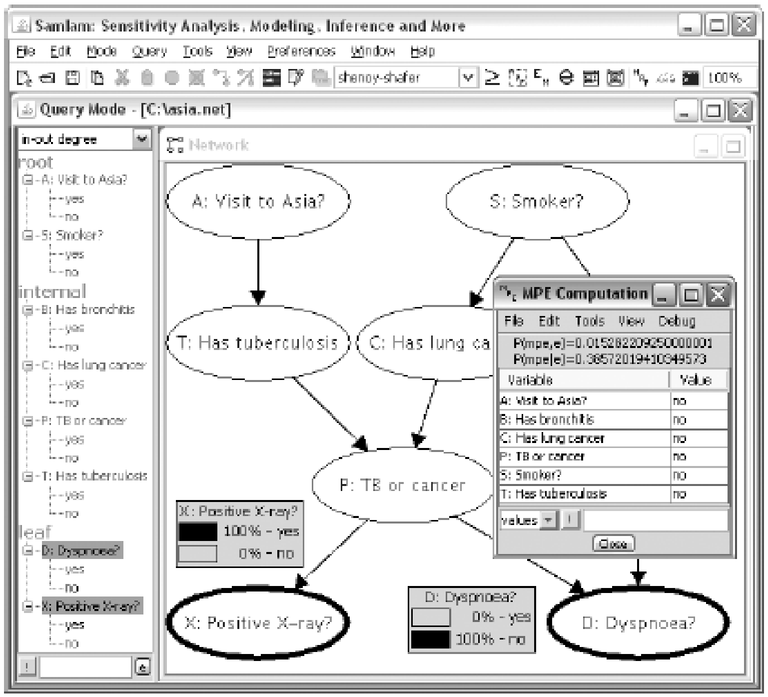


MPE given a positive X-ray and dyspnoea

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

MPE is also called MAP

Query: Most Probable Explanation (MPE)

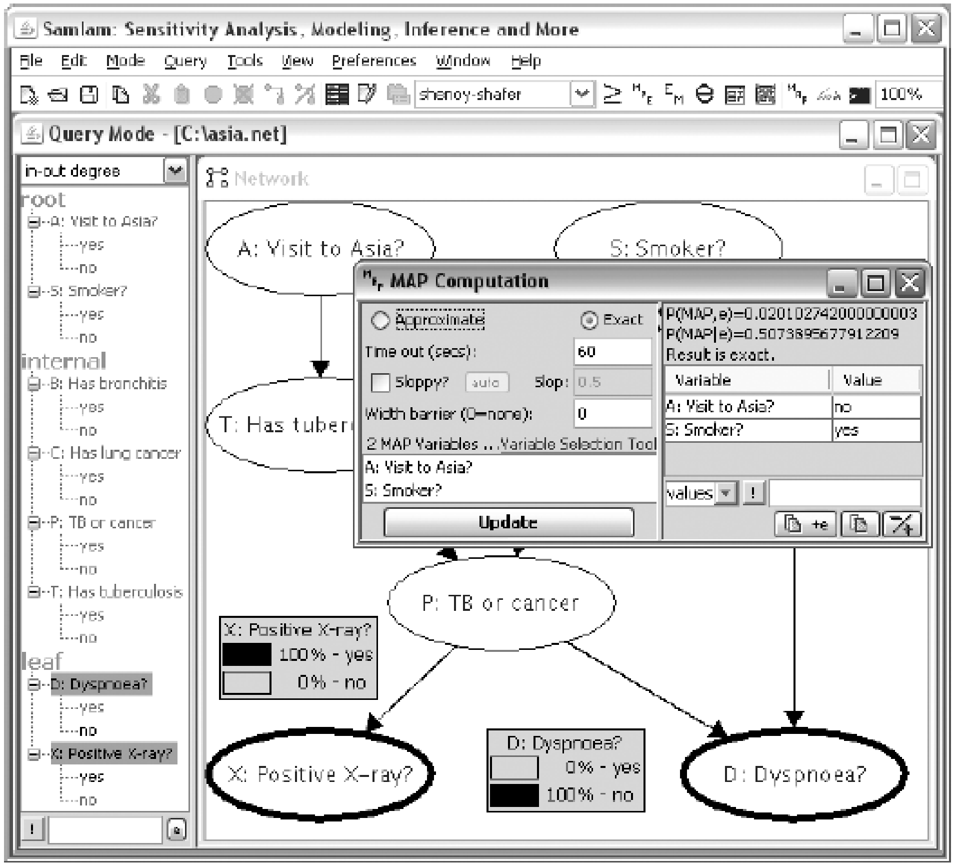


MPE given a positive X-ray and no dyspnoea ($\approx 38.57\%$)

A patient that made no visit to Asia; is not a smoker; has no lung cancer, no bronchitis and no tuberculosis.

Choosing values with maximal probability, we get:
α: $A =$ no, $S =$ yes, $T =$ no, $C =$ no, $B =$ no, $P =$ no, $X =$ yes, $D =$ no.
Probability $\approx 20.03\%$ given evidence $\mathbf{e}$: $X =$ yes, $D =$ no.

Query: Maximum a Posteriori Hypothesis (MAP)



MAP variables $\mathbf{M} = \{A, S\}$ and evidence $\mathbf{e}: X = \text{yes},\ D = \text{no}$

MAP is A=no, S=yes.

MAP has probability of $\approx 50.74\%$ given the evidence.

MAP is also called Marginal Map (MMAP)

Query: Maximum a Posteriori Hypothesis (MAP)

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 gi

Is it correct?

MA

Query: Maximum a Posteriori Hypothesis (MAP)

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:

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

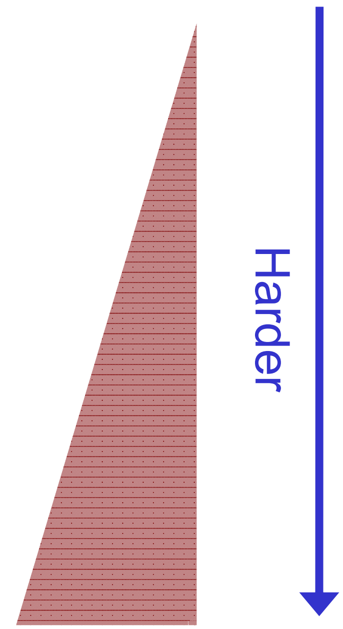
with probability $\approx 48.09\%$ given the evidence.

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

Probabilistic Reasoning Problems

Tasks:

- **Max-Inference** $$f(\mathbf{x}^*) = \max_{\mathbf{x}} \prod_{\alpha} f_\alpha(\mathbf{x}_\alpha)$$
  (most likely config, MPE.)
- **Sum-Inference** $$Z = \sum_{\mathbf{x}} \prod_{\alpha} f_\alpha(\mathbf{x}_\alpha)$$
  (data likelihood, P(evidence))
- **Mixed-Inference** $$f(\mathbf{x}_M^*) = \max_{\mathbf{x}_\mathrm{M}} \sum_{\mathbf{x}_S} \prod_{\alpha} f_\alpha(\mathbf{x}_\alpha)$$
  (optimal prediction, MAP, Marginal Map)



Combinatorial search / counting queries

Exact reasoning NP-complete (or worse)

Modeling with Bayesian Networks

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.

Modeling with Bayesian Networks

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 $P(x_1, x_2, \ldots, x_n)$ and any ordering d of the variables, the DAG created by designating as parents of X_i any minimal set Π_{X_i} of predecessors satisfying

$$P(x_i \mid \pi_{X_i}) = P(x_i \mid x_1, \ldots, x_{i-1}), \quad \Pi_{X_i} \subseteq \{X_1, X_2, \ldots, X_{i-1}\} \qquad (3.27)$$

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 Π_X, and that no proper subset of Π_X satisfy this condition.

Intuition: The causes of X can serve as the parents

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.

Diagnosis I: Model from Expert

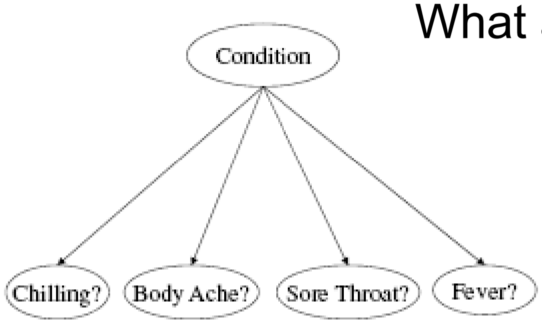
Example

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.

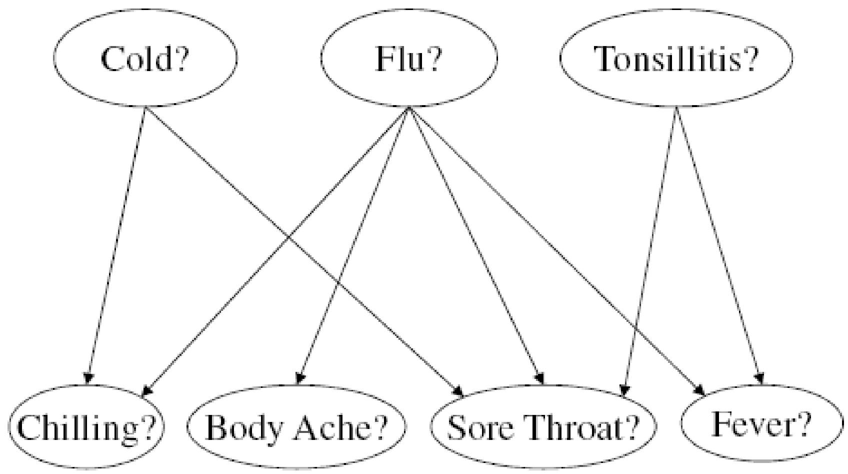
Variables? Arcs? Try it.

Diagnosis I: Model from Expert



What about?

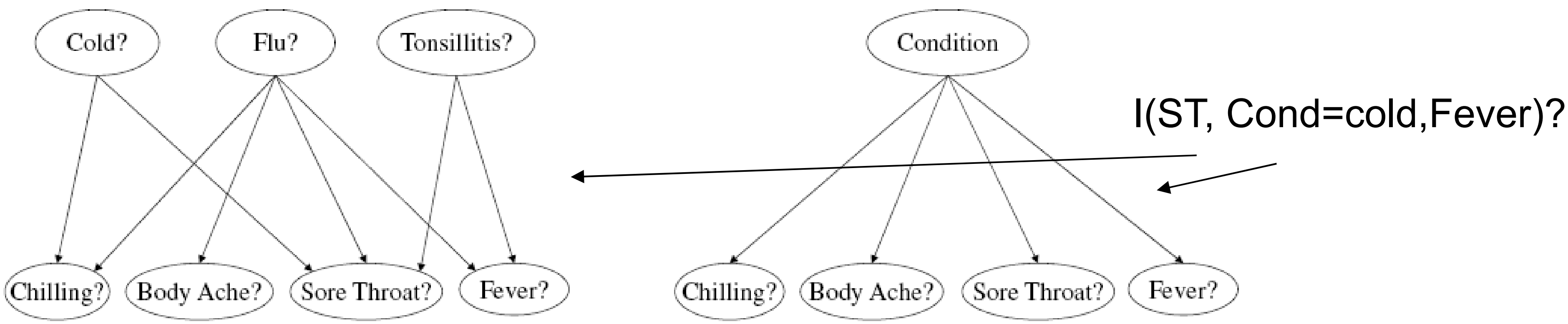
A naive Bayes structure
has the following edges C -> A1, . . . , C -> Am, where C is called the class variable and A1; : : : ;Am are called the attributes.



Variables are binary: values are either true or false. More refined information may suggest different degrees of body ache.

Diagnosis I: Model from Expert

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.

Diagnosis I: Learn the model from data

CPTs can be obtained from medical experts, who supply this information based on known medical statistics or subjective beliefs gained through practical experience.

CPTs can also be estimated from medical records of previous patients

Case	*Cold?*	*Flu?*	*Tonsillitis?*	*Chilling?*	*Bodyache?*	*Sorethroat?*	*Fever?*
1	true	false	?	true	false	false	false
2	false	true	false	true	true	false	true
3	?	?	true	false	?	true	false
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮

? indicates the unavailability of corresponding data for that patient.

Diagnosis I: M

Learning the model

- Tools for Bayesian network inference can generate a network parameterization Θ, which tries to maximize the probability of seeing the given cases.
- If each case is represented by event $\mathbf{d}_i$, such tools will generate a parametrization Θ which leads to a probability distribution $\Pr$ that attempts to maximize:

$$\prod_{i=1}^{N} \Pr(\mathbf{d}_i).$$

- Term $\Pr(\mathbf{d}_i)$ 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

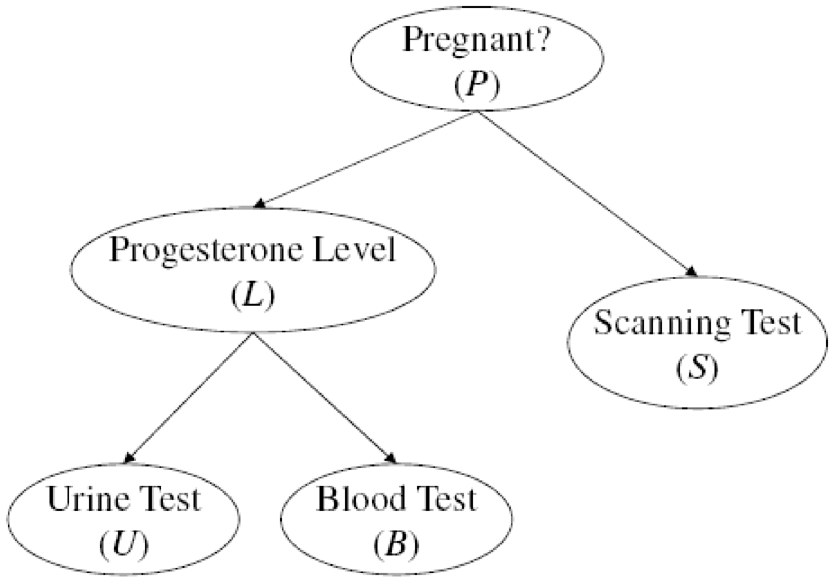
A few weeks after inseminating a cow, we have three possible tests to confirm pregnancy. The first is a scanning test which has a false positive of 1% and a false negative of 10%. The second is a blood test, which detects progesterone with a false positive of 10% and a false negative of 30%. The third 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



P	θ_p
yes	.87

P	S	$\theta_{s\|p}$
yes	−ve	.10
no	+ve	.01

P	L	$\theta_{l\|p}$
yes	undetectable	.10
no	detectable	.01

L	B	$\theta_{b\|l}$
detectable	−ve	.30
undetectable	+ve	.10

L	U	$\theta_{u\|l}$
detectable	−ve	.20
undetectable	+ve	.10

Diagnosis II: Model from Expert

Example

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

$$\mathbf{e}: \ S = -\text{ve}, \ B = -\text{ve}, \ U = -\text{ve}.$$

Posterior marginal for pregnancy given this evidence:

P	$\Pr(P \mid \mathbf{e})$
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.

Sensitivity Analysis

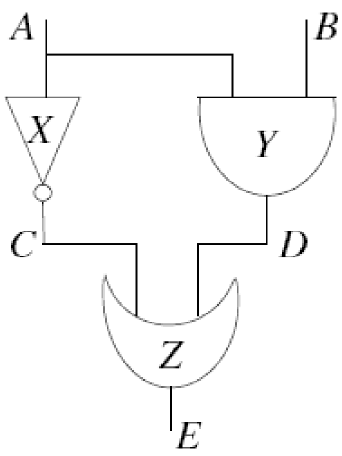
Example

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

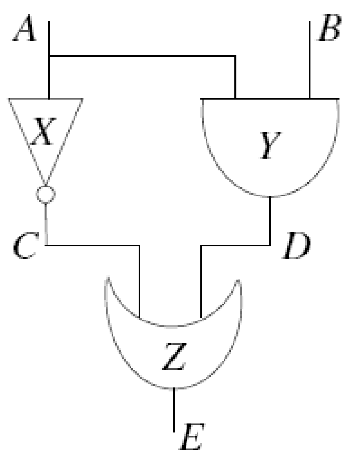


Problem statement

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



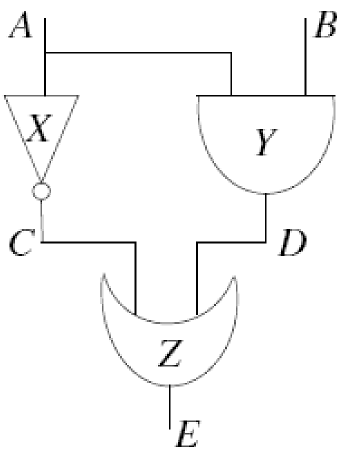
Problem statement

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.

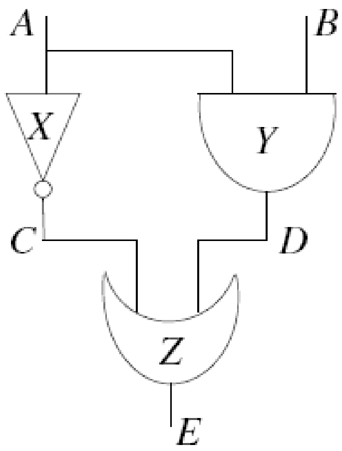
Evidence variables

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

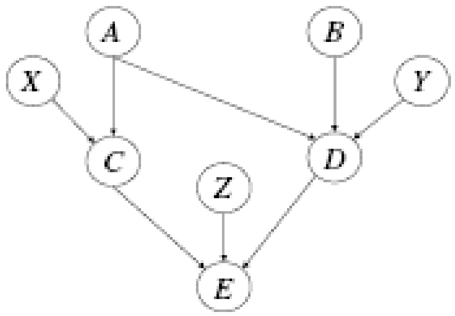
Primary inputs and output of the circuit, A, B and E.

Query variables

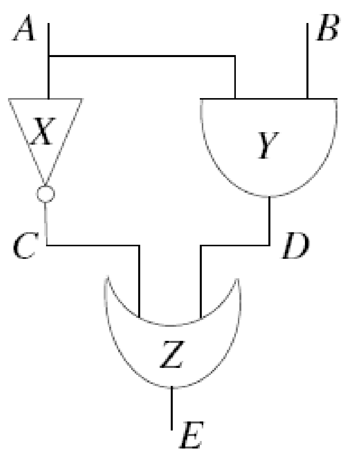
Health of components X, Y and Z.

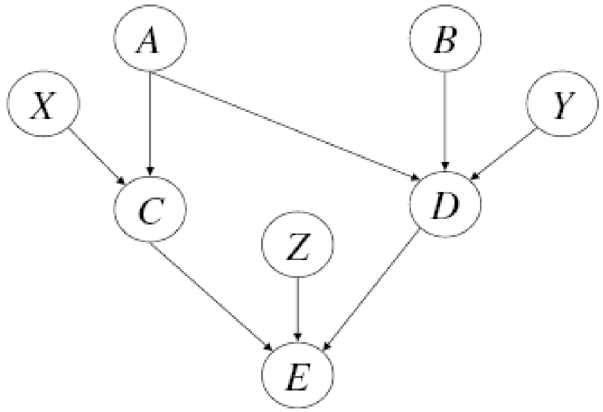
Intermediary variables

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.

Diagnosis III: Model from Design

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

X	θ_X
ok	.99
faulty	.01

X	θ_X
ok	.99
stuckat0	.005
stuckat1	.005

Need to know the probabilities of various fault modes.

Diagnosis III: Model from Design

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

Diagnosis III: Model from Design

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.

A Diagnosis Example

Example

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

A Diagnosis Example

Example

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

Network with fault modes gives two MAP instantiations:

MAP given $\mathbf{e}$	X	Y	Z	
	ok	stuckat0	ok	each probability $\approx 49.4\%$
	ok	ok	stuckat0	

A Diagnosis Example

Example

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

Network with fault modes gives two MAP instantiations:

MAP given $\mathbf{e}$	X	Y	Z
	ok	stuckat0	ok
	ok	ok	stuckat0

each probability $\approx 49.4\%$

Network with no fault modes gives two MAP instantiations:

MAP given $\mathbf{e}$	X	Y	Z
	ok	faulty	ok
	ok	ok	faulty

each probability $\approx 49.4\%$

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'

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'

Integrating Time

Suppose we have two test vectors instead of only one.

Additional evidence variables

A', B' and E'

Additional intermediary variables

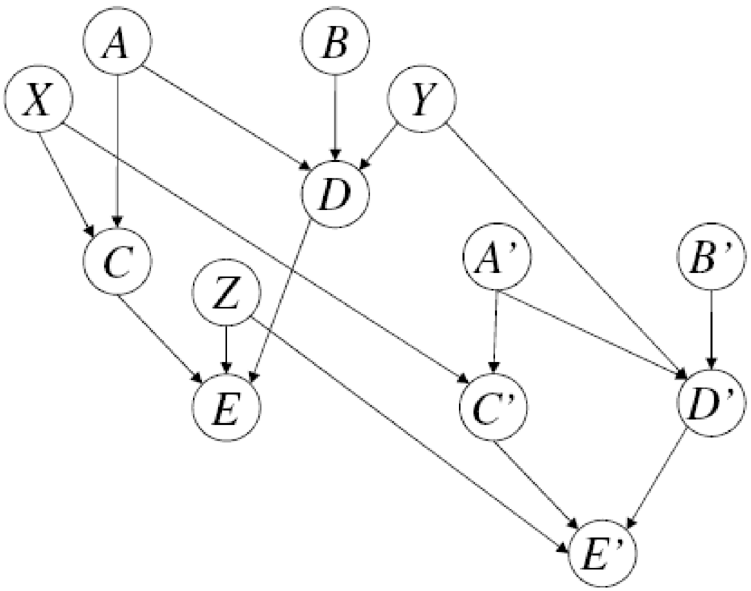
C' and D'

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.

Variables? Values? Structure?

Integrating Time: No Intermittent Faults

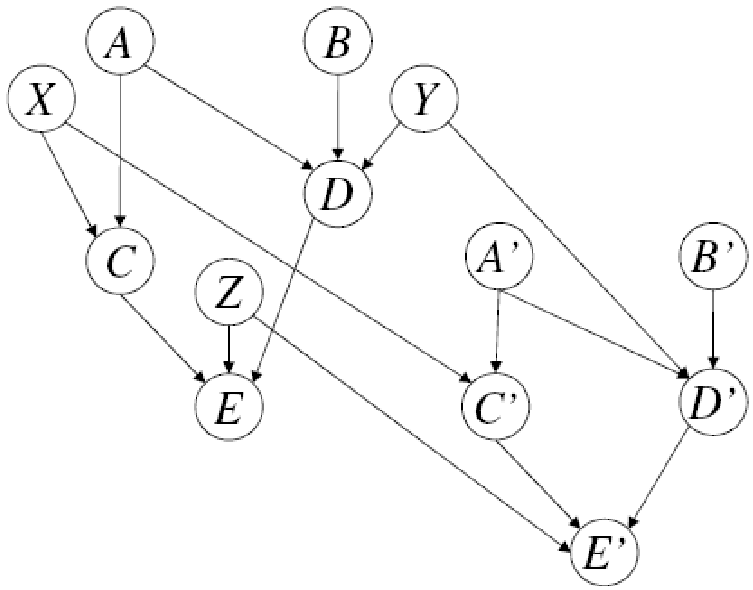


Two test vectors

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

$\mathbf{e}'$: A = low, B = low, E = low.

Integrating Time: No Intermittent Faults



Two test vectors

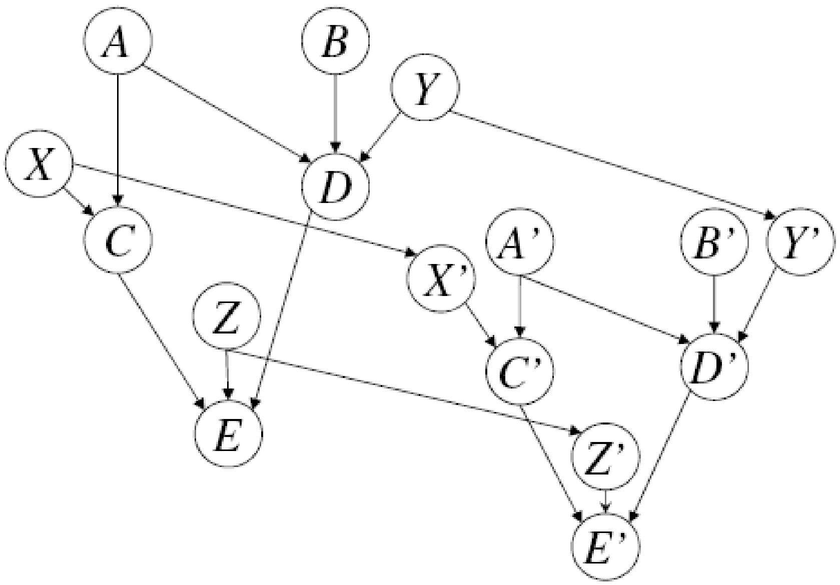
$\mathbf{e}$: $A =$ high, $B =$ high, $E =$ low
$\mathbf{e}'$: $A =$ low, $B =$ low, $E =$ low.

MAP using second structure

MAP given $\mathbf{e}, \mathbf{e}'$	X	Y	Z
	ok	ok	faulty

with probability $\approx 97.53\%$

Integrating Time: Intermittent Faults



Dynamic Bayesian network (DBN)

Two test vectors

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

Persistence model for the health of component X

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

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?

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.

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

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

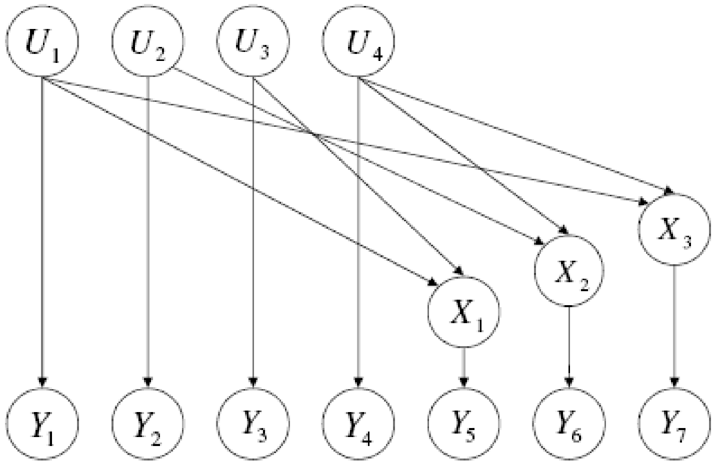
$Y_1, \ldots, Y_7$: bits received at destination D

Query variables are

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

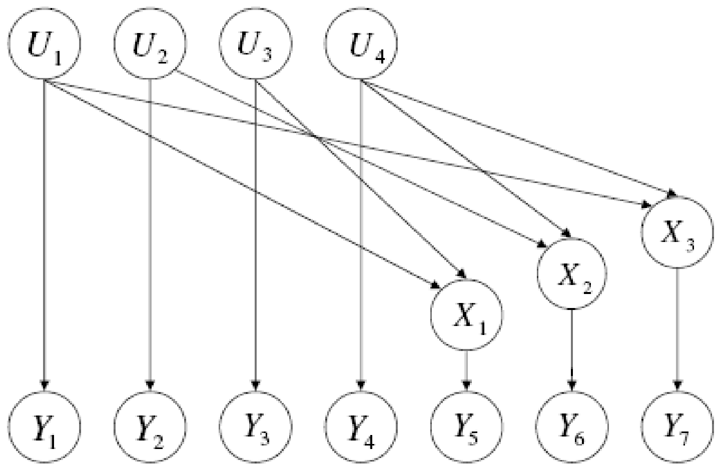
Bits $X_1, \ldots, X_3$ either query variables or intermediary variables.

Channel Coding



There are three CPT types in the problem.

Channel Coding



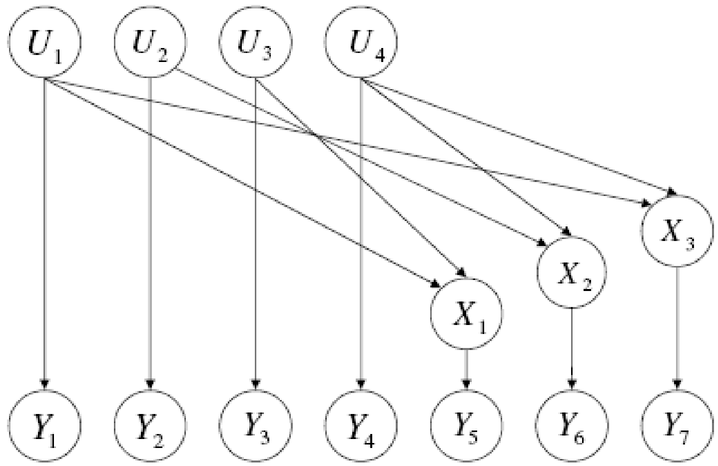
There are three CPT types in the problem.

CPT for each redundant bit, say X_1:

U_1	U_3	X_1	$\theta_{x_1 \| u_1, u_3}$
1	1	1	0
1	0	1	1
0	1	1	1
0	0	1	0

$\Pr(x_1|u_1, u_3) = 1$ iff $x_1 = u_1 \oplus u_3$ ($\oplus$ is the XOR function)

Channel Coding



There are three CPT types in the problem.

CPT for information bits, such as U_1:

U_1	θ_{u_1}
1	.5
0	.5

Captures the distribution of messages sent out from the source S

P(Y not equal U) = 0.01

What queries should we use here?

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?

Noise Models and Soft Evidence

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 σ.

Channel output Y_i is a continuous variable governed by

conditional density function $f(y_i|x_i) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-(y_i - x_i)^2/2\sigma^2}$

Can be implemented by interpreting

channel output y_i as soft evidence on the channel input $X_i = 0$ with a Bayes factor $k = e^{(1-2y_i)/2\sigma^2}$

Notice:
Odds: o(x) = P(x)\P(bar(x))
K =Bayes factor = o'(x)\o(x) … the posterior odds after observing divided by prior odds
For Gausian x: evidence on Y=y can be emulated with soft evidence on x with
K =f(y|x) \f(y|bar(x)) = the expression above. Read chapter 5

Convolutional Codes

Convolutional and turbo codes

correspond to different methods for generating redundant bits.

Convolutional Codes

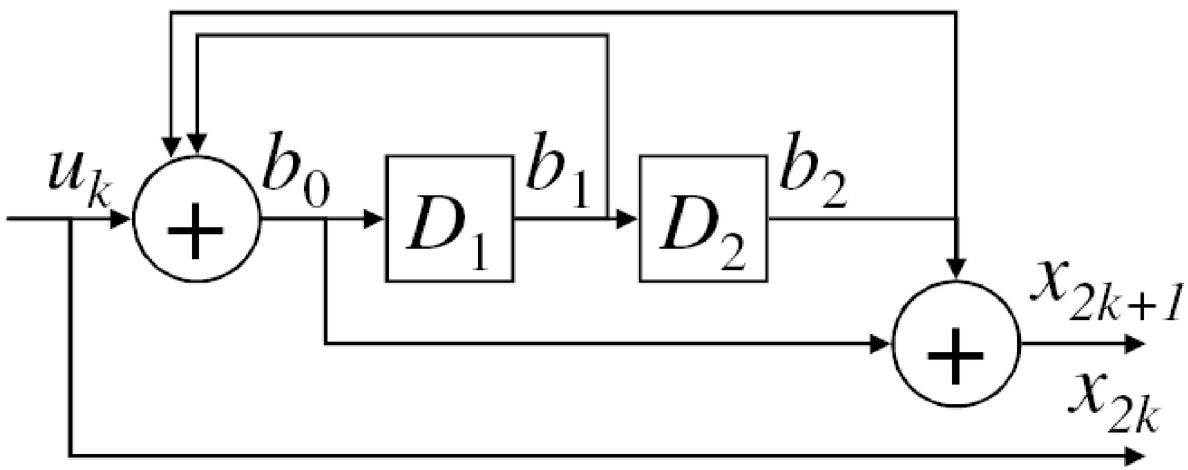
Convolutional and turbo codes

correspond to different methods for generating redundant bits.

Convolutional and turbo codes

provide examples of modeling systems with feedback loops using dynamic Bayesian networks.

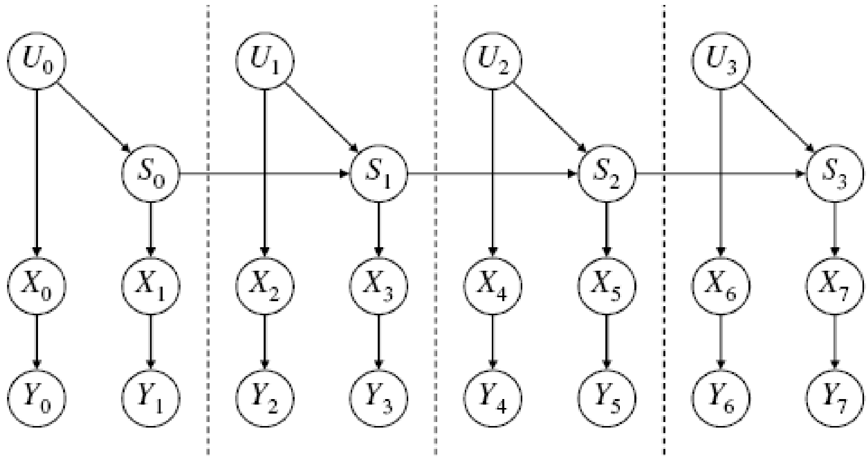
Convolutional Codes



An example convolutional encoder

Each node denoted with a "+" represents a binary addition, and each box D_i represents a delay where the output of D_i is the input of D_i from the previous encoder state.

Convolutional Codes

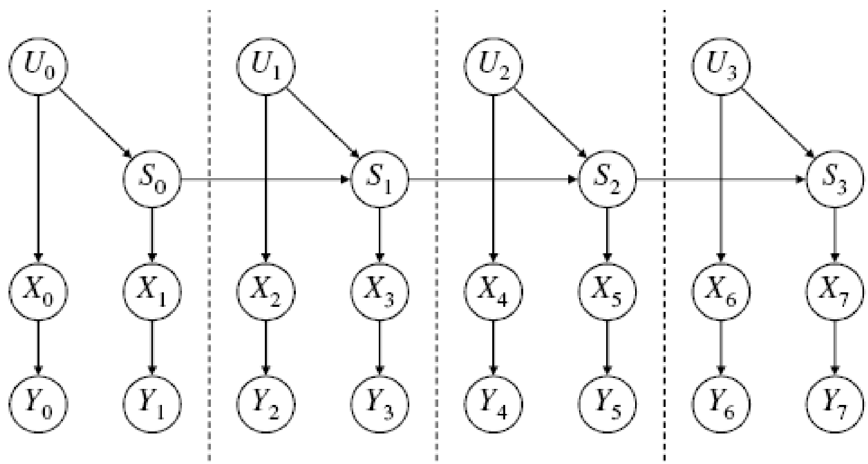


Dynamic Bayesian network for a convolutional code.

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



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.

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.

Turbo Codes

Given four information bits $u_0, \ldots, u_3$.

Turbo Codes

Given four information bits $u_0, \ldots, u_3$.

In a convolutional code

we generate 4 redundant bits leading to an 8-bit codeword.

Turbo Codes

Given four information bits $u_0, \ldots, u_3$.

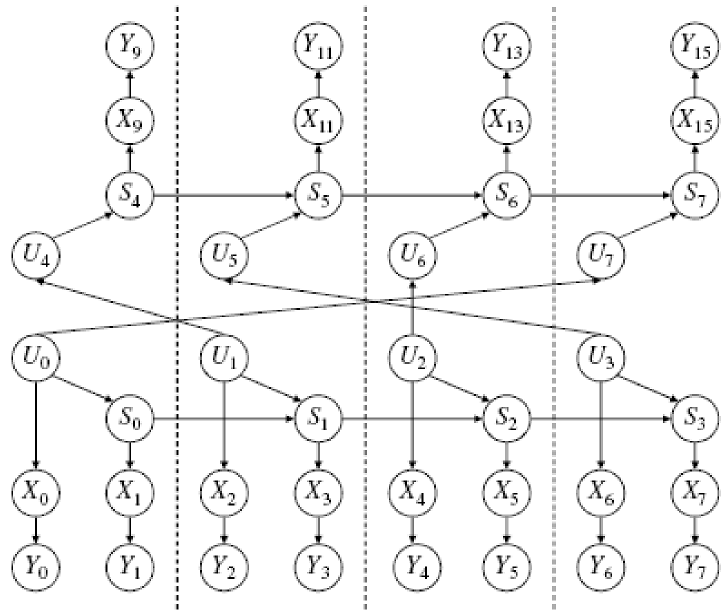
In a convolutional code

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.

Turbo Codes



Lower network represents a convolutional code

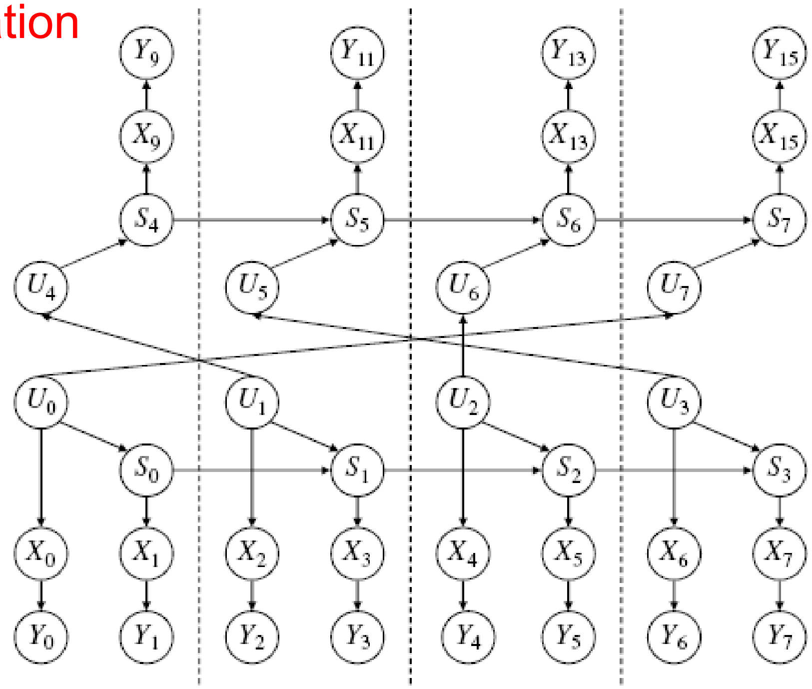
for the bit sequence $u_0, \ldots, u_3$.

Upper network represents a convolutional code

for the bit sequence $u_4, \ldots, u_7$.

Turbo Codes

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).

Read on your own

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 probablyhear 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.

Genetic Linkage Analysis

The *ABO* gene

is responsible for determining blood type. This gene has three alleles: A, B and O. Since each individual must have two alleles for this gene, we have six possible genotypes in this case.

Genetic Linkage Analysis

The *ABO* gene

is responsible for determining blood type. This gene has three alleles: A, B and O. Since each individual must have two alleles for this gene, we have six possible genotypes in this case.

There are only four different blood types

Genotype	Phenotype
A/A	Blood type A
A/B	Blood type AB
A/O	Blood type A
B/B	Blood type B
B/O	Blood type B
O/O	Blood type O

If someone has the blood type A, they could have the pair of alleles A/A or the pair A/O for their genotype.

Genetic Linkage Analysis

The phenotype is not always determined precisely by the genotype.

Genetic Linkage Analysis

The phenotype is not always determined precisely by the genotype.

A disease gene with two alleles H and D

Genotype	Phenotype
H/H	healthy
H/D	healthy
D/D	ill with probability .9

Genetic Linkage Analysis

The phenotype is not always determined precisely by the genotype.

A disease gene with two alleles H and D

Genotype	Phenotype
H/H	healthy
H/D	healthy
D/D	ill with probability .9

Penetrance

The conditional probability of observing a phenotype (e.g., healthy, ill) given the genotype (e.g., H/H, H/D, D/D).

Genetic Linkage Analysis

The phenotype is not always determined precisely by the genotype.

A disease gene with two alleles H and D

Genotype	Phenotype
H/H	healthy
H/D	healthy
D/D	ill with probability .9

Penetrance

The conditional probability of observing a phenotype (e.g., healthy, ill) given the genotype (e.g., H/H, H/D, D/D).

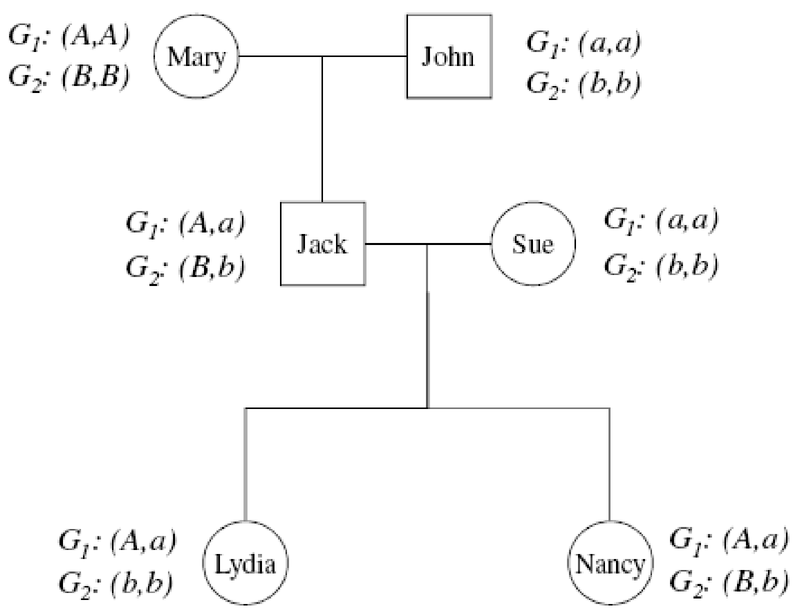
Example

Penetrance is always 0 or 1 for the ABO gene.
Penetrance is .9 for the phenotype ill given the genotype D/D.

Recombination Events

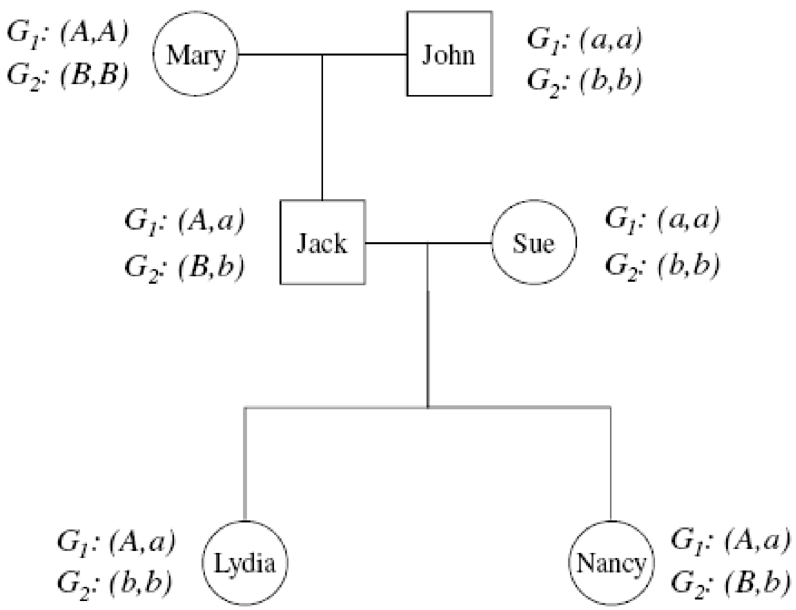
Haplotype

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.

Recombination Events



- 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.

If G1 and G2 are close then they are likely to pass down from the same haplotype (grandmother or grandfather)

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

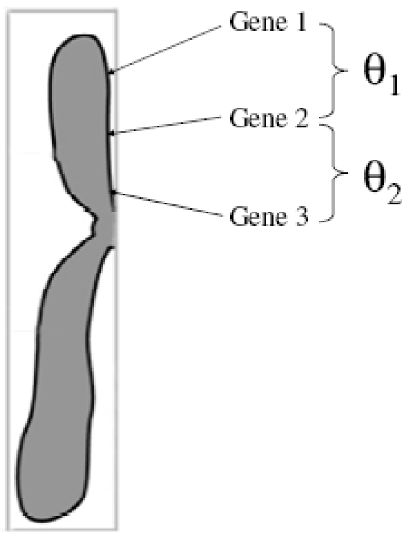
The extent to which genes G_1 and G_2 are linked

is measured by a recombination fraction or frequency, θ, 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 $\theta < 1/2$.

Genetic Linkage and Gene Maps



Linkage between genes

is related to their locations on a chromosome within the cell nucleus. These locations are typically referred to as loci (singular: locus).

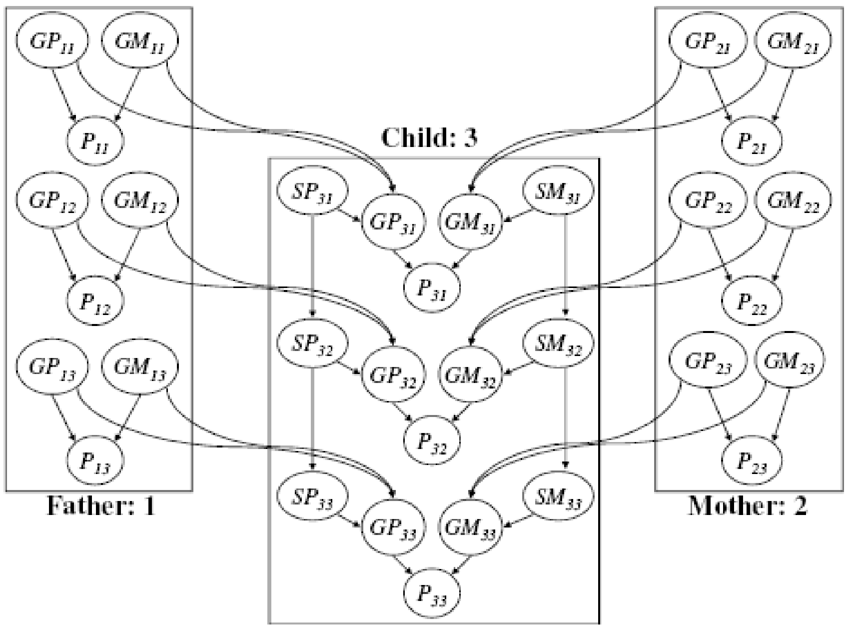
For genes that are closely located on a chromosome

linkage is inversely proportional to distance between their locations.

The recombination frequency can provide direct evidence

on the distance between genes on a chromosome.

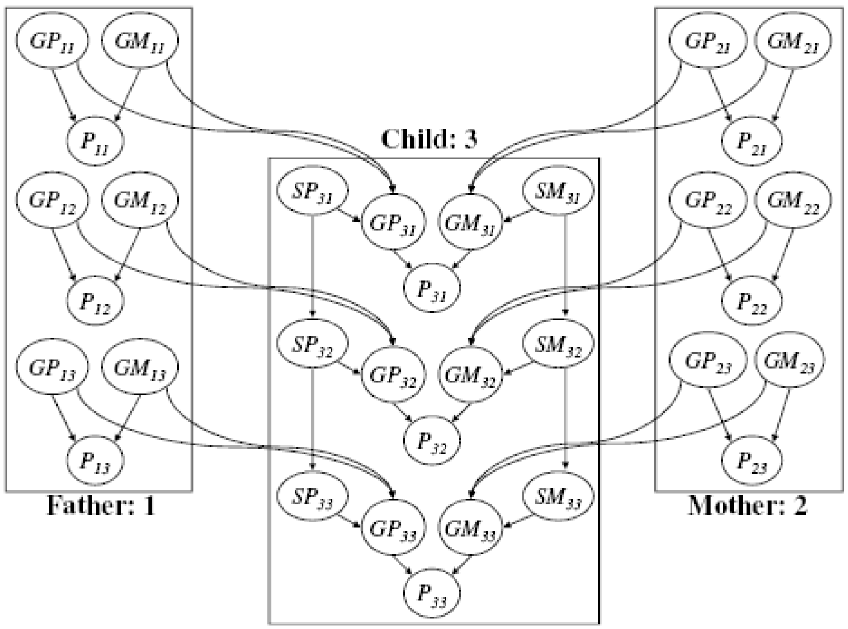
From Pedigrees to Bayesian Networks



Genotype and phenotype

—GP_{ij}: paternal allele for individual i and gene j
—GM_{ij}: maternal allele for individual i and gene j
—P_{ij}: phenotype for individual i and gene j

From Pedigrees to Bayesian Networks

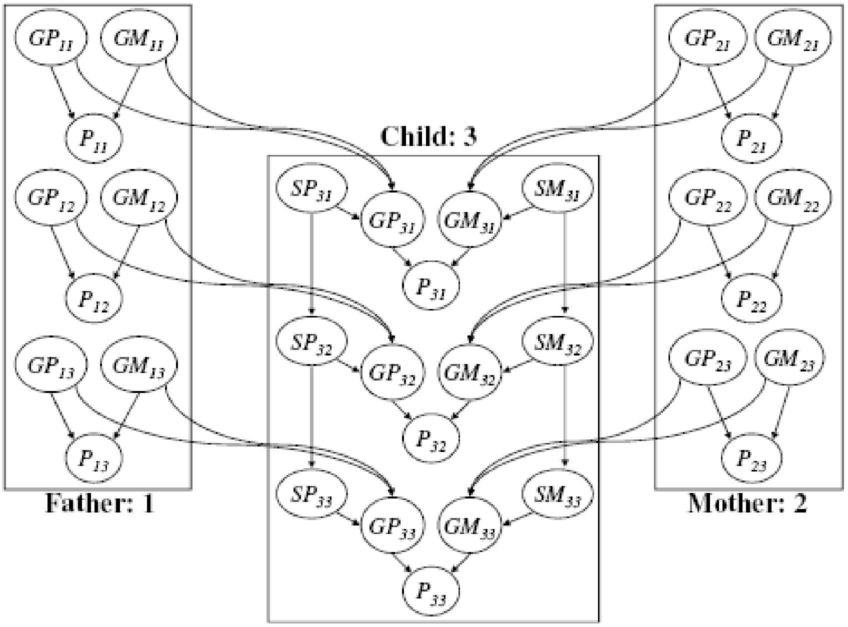


Selector variables

—SP_{ij}: determines how individual i inherits alleles of gene j from his father

—SM_{ij}: determines how individual i inherits alleles of gene j from his mother

From Pedigrees to Bayesian Networks



Selector variables

—SP_{ij}: determines how individual i inherits alleles of gene j from his father
—SM_{ij}: determines how individual i inherits alleles of gene j from his mother

If $SP_{ij} = p$ then individual i will inherit the allele of gene j

that his father obtained from the grandfather.

If $SP_{ij} = m$ then individual i will inherit the allele of gene j

that his father obtained from the grandmother.

From Pedigrees to Bayesian Networks

$$\theta_{gp_{ij}|gp_{kj},gm_{kj},sp_{ij}} = \begin{cases} 1, & \text{if } sp_{ij} = p \text{ and } gp_{ij} = gp_{kj}; \\ 1, & \text{if } sp_{ij} = m \text{ and } gp_{ij} = gm_{kj}; \\ 0, & \text{otherwise.} \end{cases}$$

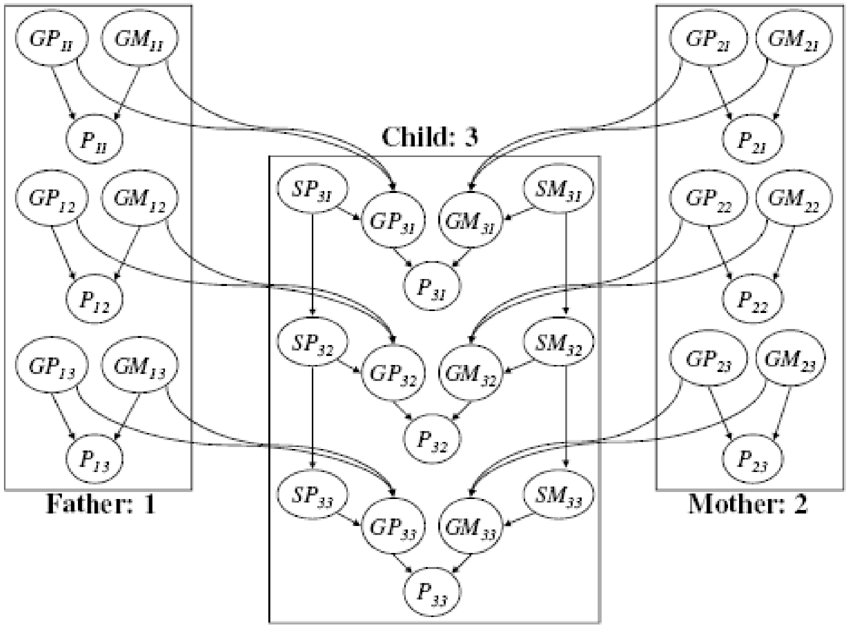
If $SP_{ij} = p$ then the allele GP_{ij} for individual i and gene j

will be inherited from the paternal haplotype of his father k, GP_{kj}

If $SP_{ij} = m$ then the allele GP_{ij} for individual i and gene j

will be inherited from the maternal haplotype of his father k, GM_{kj}

From Pedigrees to Bayesian Networks



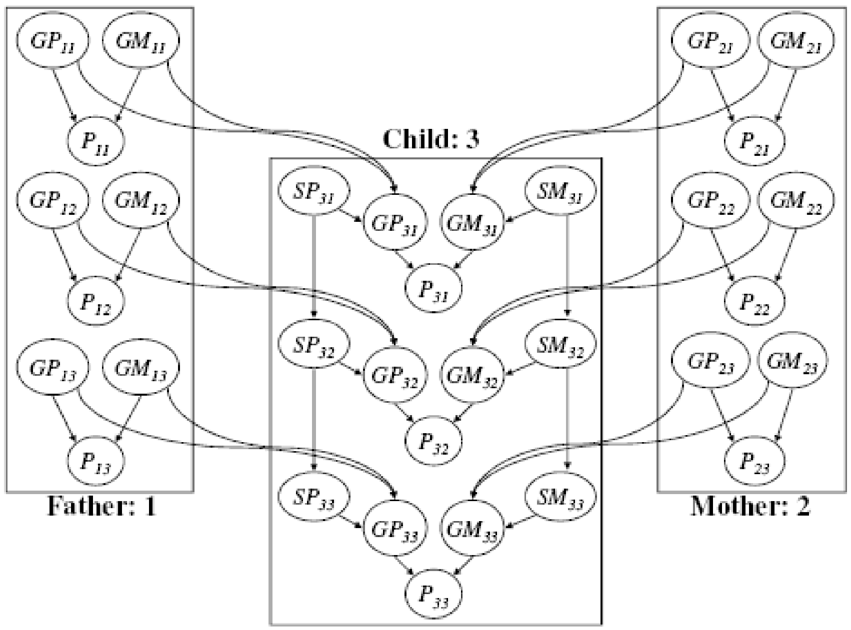
Selectors of second gene SP_{32} and SM_{32} have CPTs

that are a function of recombination frequency θ_{12}

Selectors of third gene SP_{33} and SM_{33} have CPTs

that are a function of recombination frequency θ_{23}

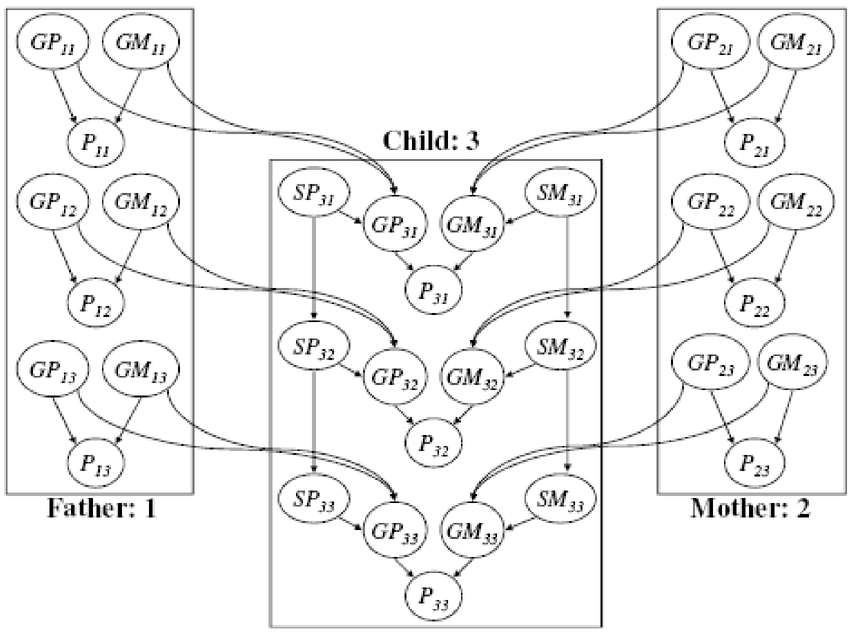
From Pedigrees to Bayesian Networks



CPT for selector variable SP_{32}

encodes the recombination frequency θ_{12}

From Pedigrees to Bayesian Networks

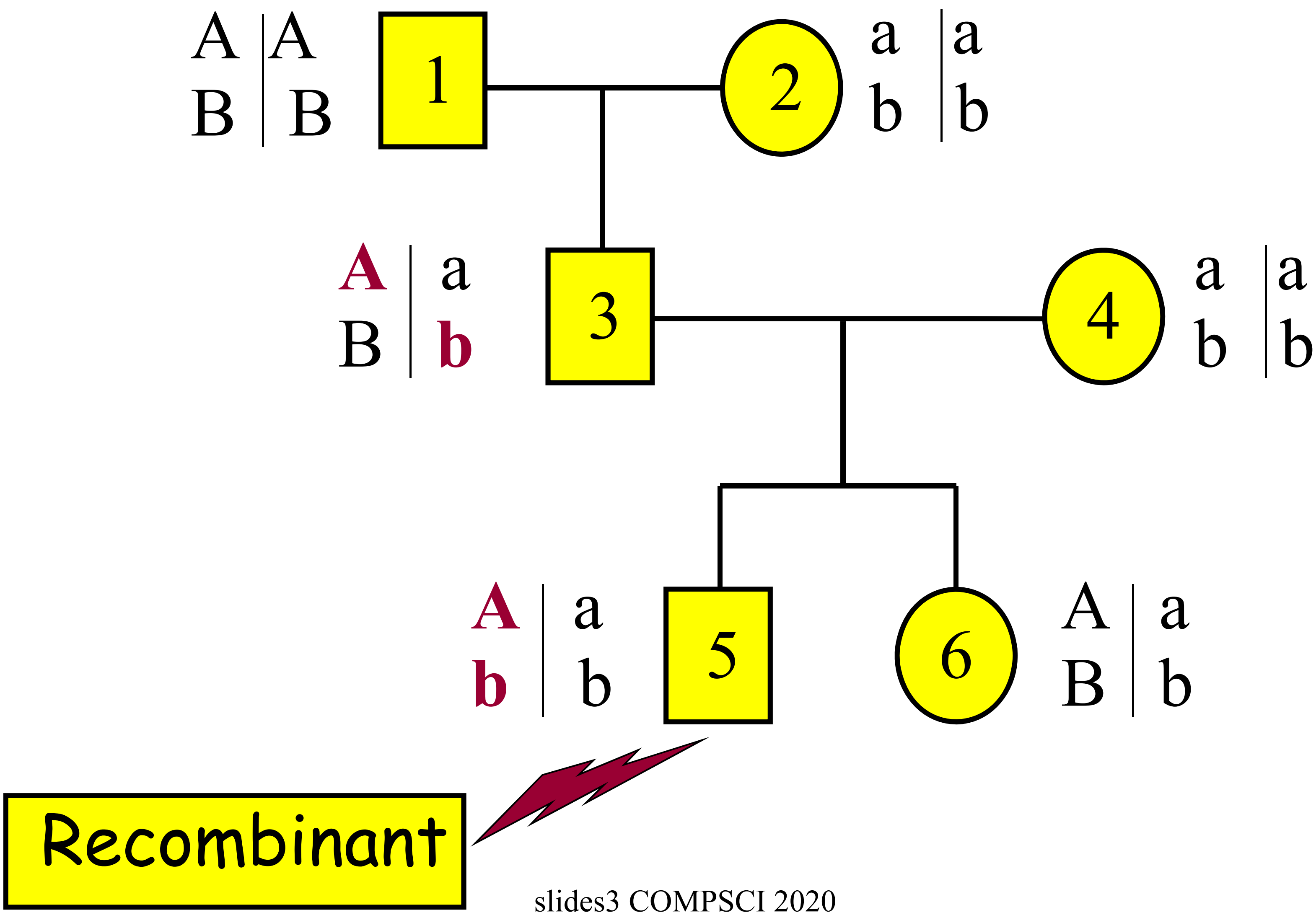


CPT for selector variable SP_{32}

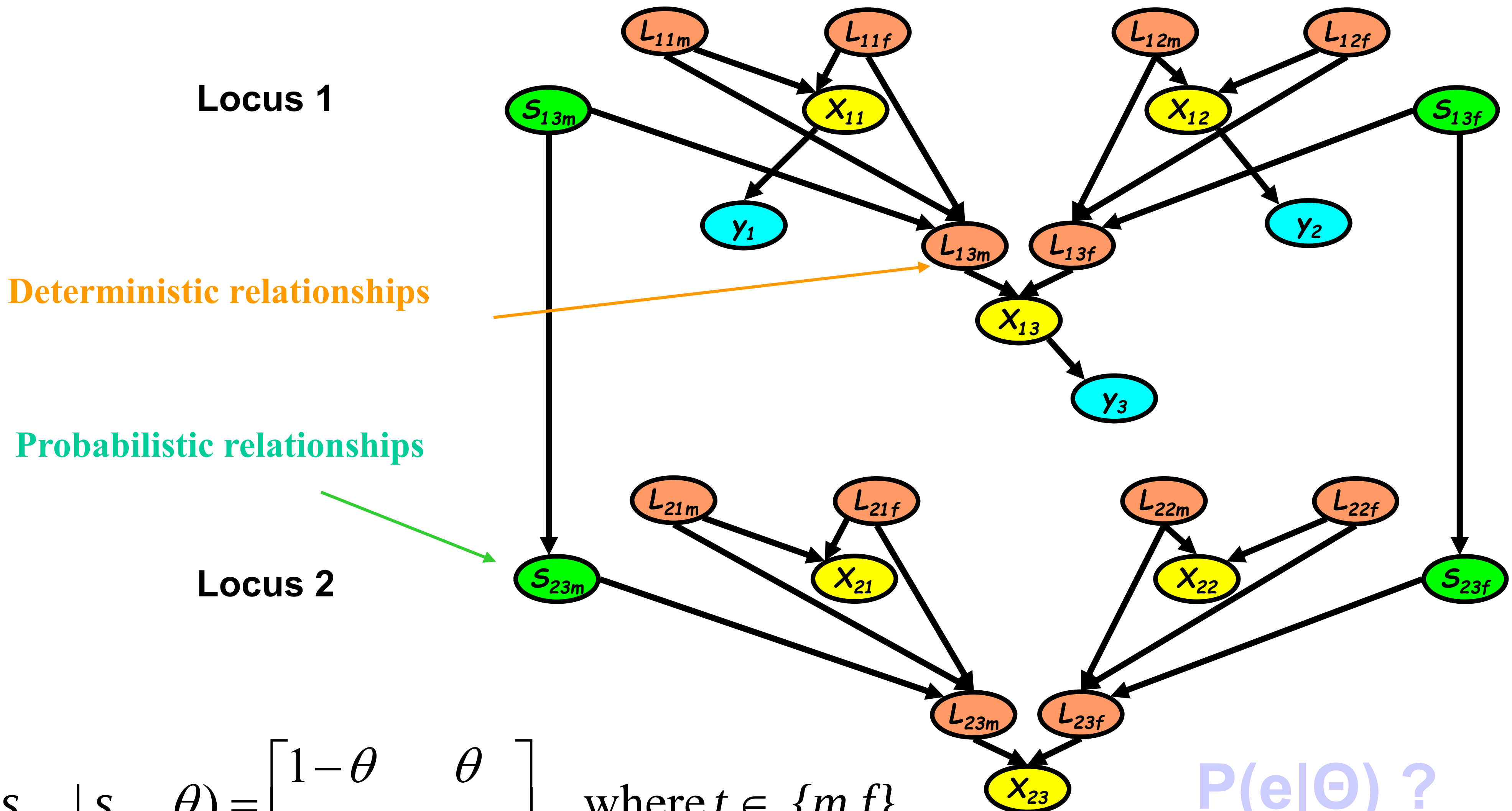
encodes the recombination frequency θ_{12}

SP_{31}	SP_{32}	$\theta_{sp_{32}\mid sp_{31}}$	
p	p	$1-\theta_{12}$	
p	m	θ_{12}	recombination between genes 1 and 2
m	p	θ_{12}	recombination between genes 1 and 2
m	m	$1-\theta_{12}$	

Two Loci Inheritance



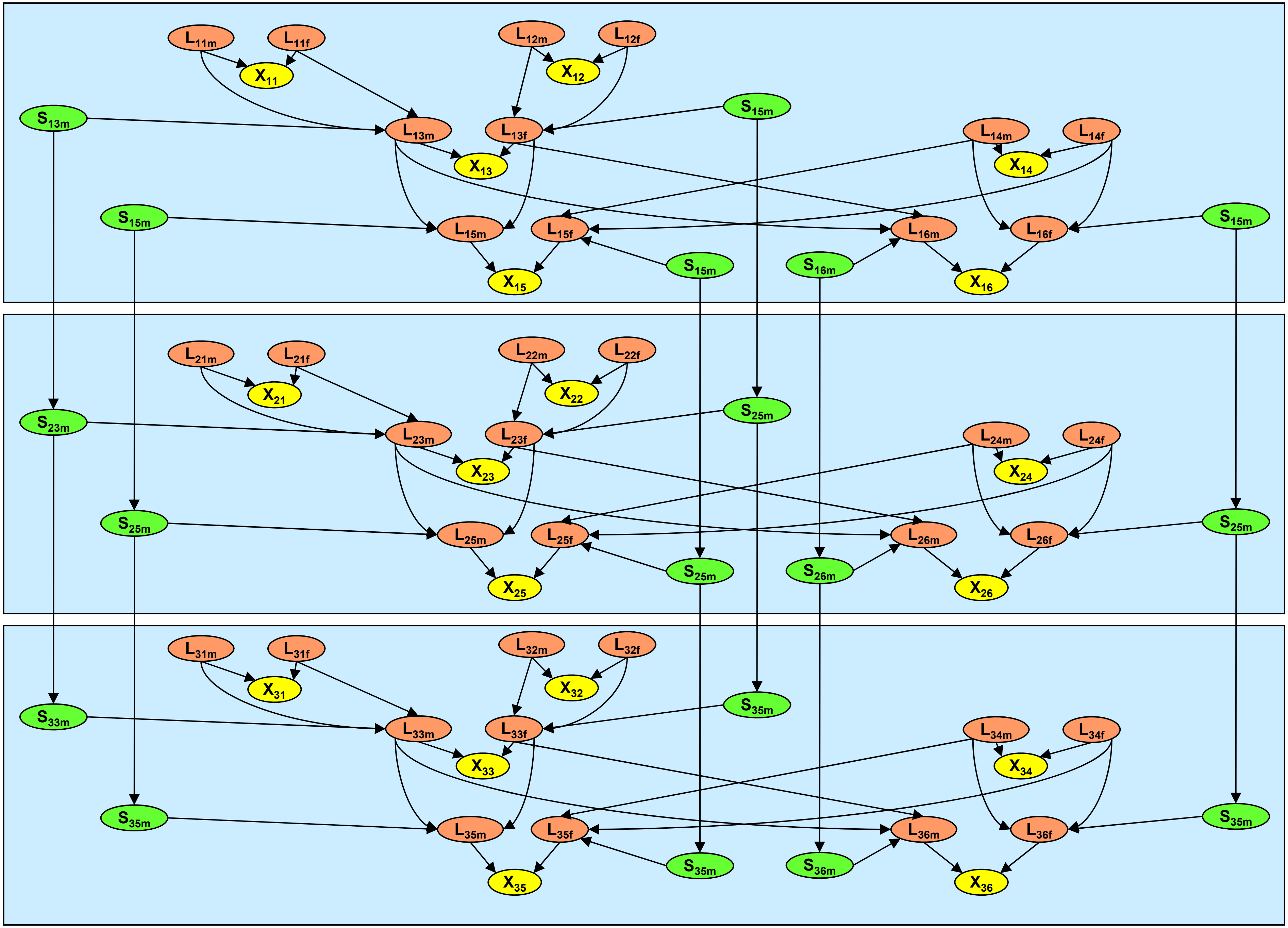
Bayesian Network for Recombination



$$P(s_{23t} \mid s_{13t}, \theta) = \begin{bmatrix} 1-\theta & \theta \\ \theta & 1-\theta \end{bmatrix}$$ where $t \in \{m,f\}$

P(e|Θ) ?

Linkage analysis:
6 people, 3 markers



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

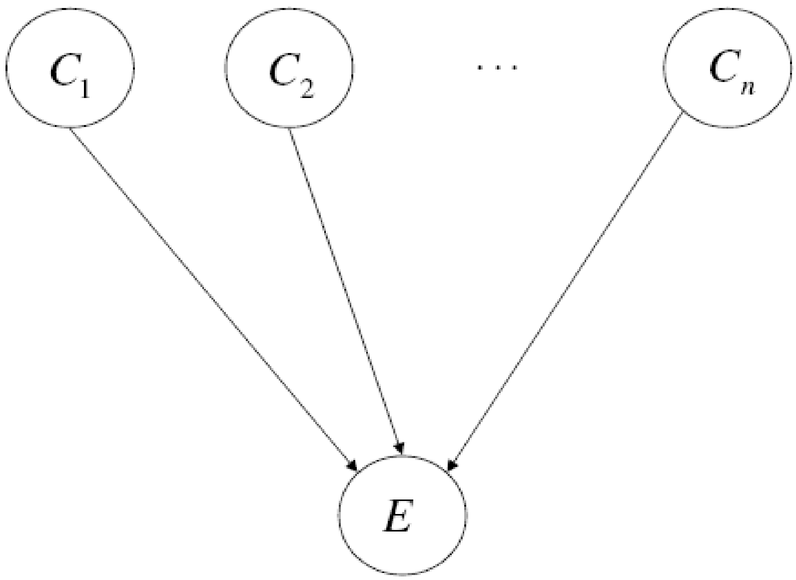
The size of a CPT

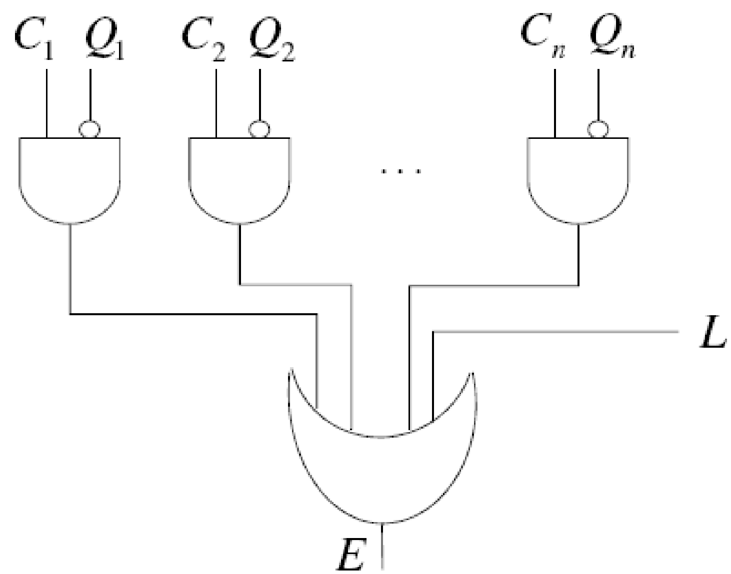
for binary variable E with binary parents $C_1, \ldots, C_n$

Number of Parents: n	Parameter Count: 2^n
2	4
3	8
6	64
10	1024
20	$1,048,576$
30	$1,073,741,824$

Micro Model

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





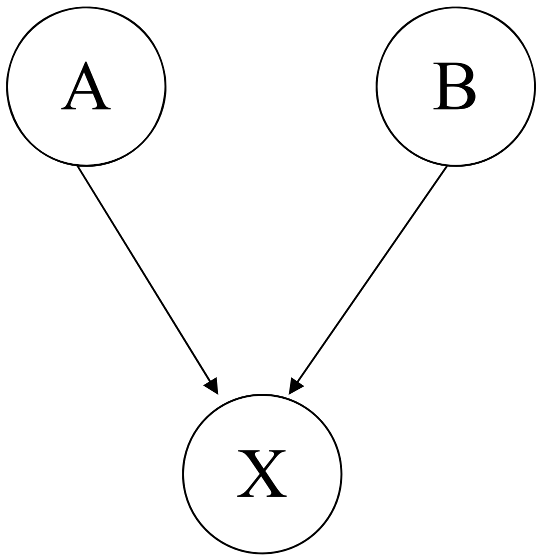
A noisy-or circuit

A micro model

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

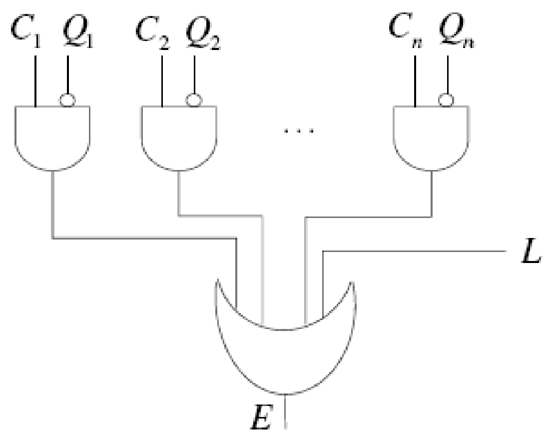
We wish to specify cpt with less parameters

Binary OR



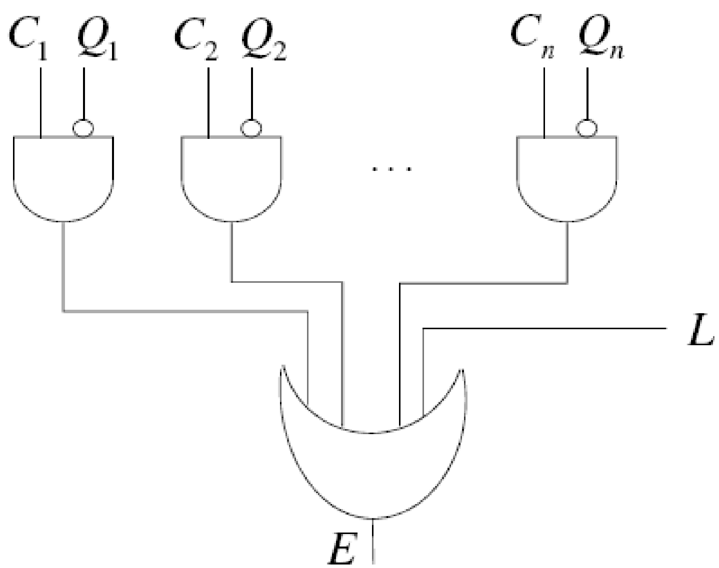
A	B	P(X=0|A,B)	P(X=1|A,B)
0	0	1	0
0	1	0	1
1	0	0	1
1	1	0	1

Noisy-or Model



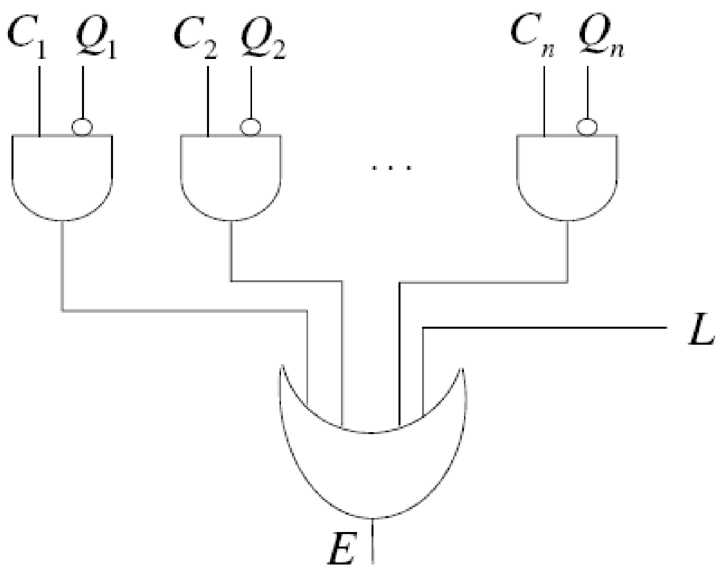
- Cause C_i is capable of establishing effect E, except under some unusual circumstances summarized by suppressor Q_i.
- When suppressor Q_i is active, C_i is no longer able to establish E.
- The leak variable L represents all other causes of E which were not modeled explicitly.
- When none of the causes 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 I_α be the indices of causes that are active in α.

Noisy-or Model

- Let I_α be the indices of causes that are active in α.
- If

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

then $I_\alpha = \{1, 2, 5\}$.

Noisy-or Model

- Let I_α be the indices of causes that are active in α.
- If

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

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

$$\begin{aligned} \Pr(E = \text{passive} | \alpha) &= (1 - \theta_l) \prod_{i \in I_\alpha} \theta_{q_i} \\ \Pr(E = \text{active} | \alpha) &= 1 - \Pr(E = \text{passive} | \alpha). \end{aligned}$$

Noisy-or Model

- Let I_α be the indices of causes that are active in α.
- If

  α: $C_1 =$ active, $C_2 =$ active, $C_3 =$ passive, $C_4 =$ passive, $C_5 =$ active,

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

$$\begin{aligned}
\Pr(E = \text{passive}|\alpha) &= (1 - \theta_l) \prod_{i \in I_\alpha} \theta_{q_i} \\
\Pr(E = \text{active}|\alpha) &= 1 - \Pr(E = \text{passive}|\alpha).
\end{aligned}$$

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 .15
- The suppressor probability for flu, say, .01
- The suppressor probability for tonsillitis, say .05
- The leak probability, say .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	$\theta_{s\|c,f,t}$	
true	true	true	true	0.9999265	$1-(1-.02)(.15)(.01)(.05)$
true	true	false	true	0.99853	$1-(1-.02)(.15)(.01)$
true	false	true	true	0.99265	$1-(1-.02)(.15)(.05)$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	
false	false	false	true	.02	$1-(1-.02)$

Noisy/OR CPDs

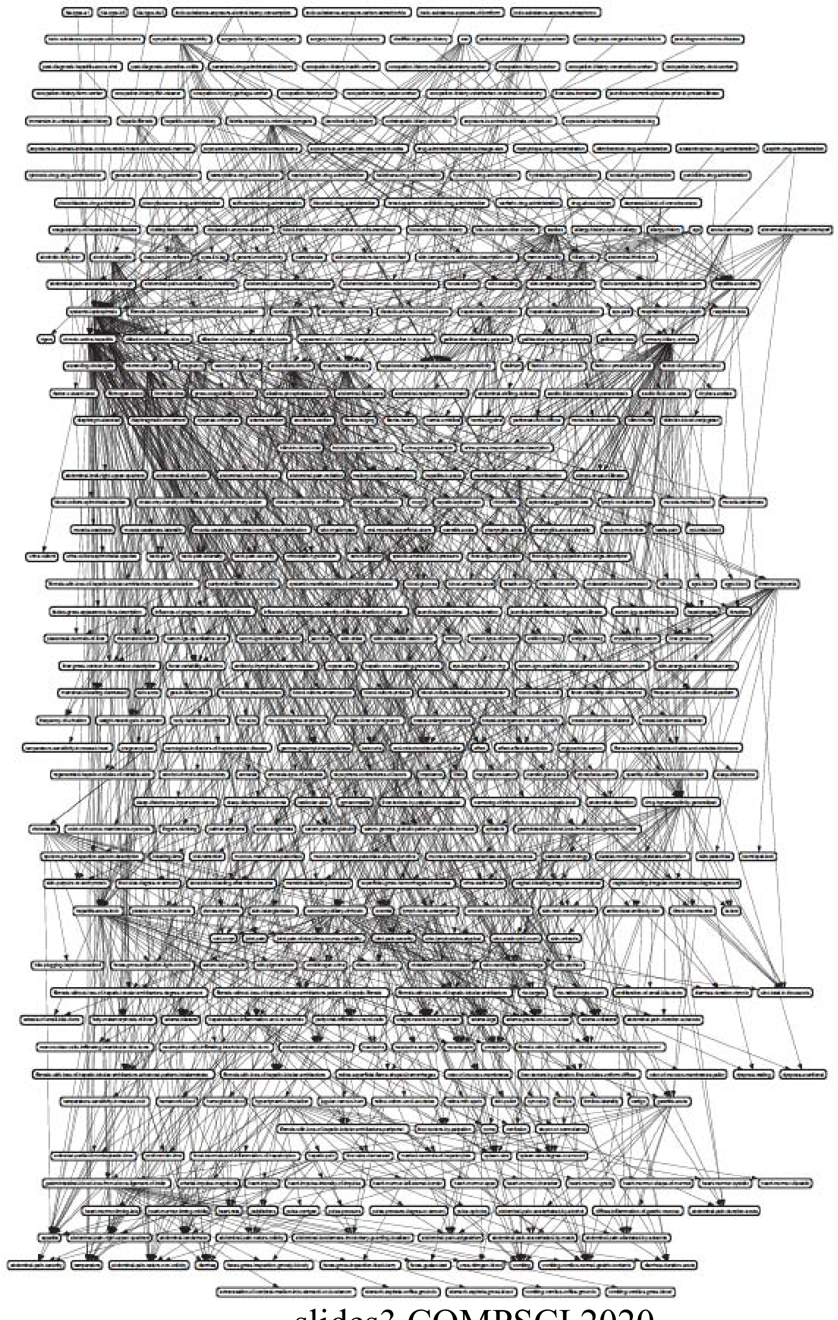
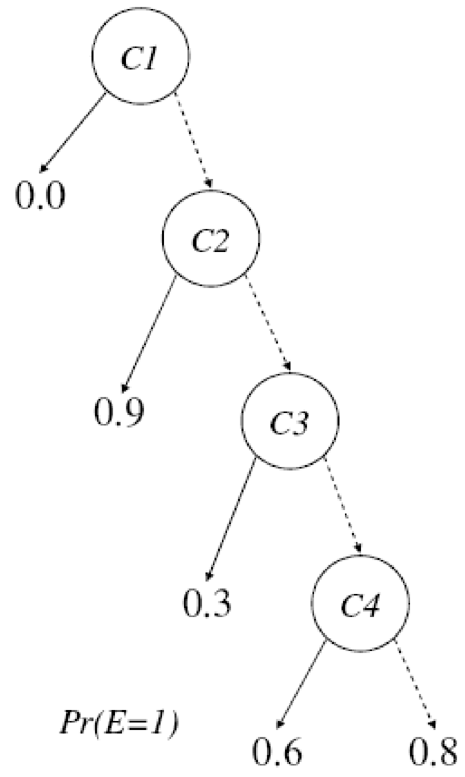


Figure 11: the CPCS network for diagnosis of internal diseases. The network contains 448 nodes, 906 links.

Decision Trees

C1	*C2*	*C3*	*C4*	*Pr(E=1)*
1	1	1	1	0.0
1	1	1	0	0.0
1	1	0	1	0.0
1	1	0	0	0.0
1	0	1	1	0.0
1	0	1	0	0.0
1	0	0	1	0.0
1	0	0	0	0.0
0	1	1	1	0.9
0	1	1	0	0.9
0	1	0	1	0.9
0	1	0	0	0.9
0	0	1	1	0.3
0	0	1	0	0.3
0	0	0	1	0.6
0	0	0	0	0.8



If-Then Rules

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

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

If $C_1 = 1$	then	$\Pr(E = 1) = 0.0$
If $C_1 = 0 \wedge C_2 = 1$	then	$\Pr(E = 1) = 0.9$
If $C_1 = 0 \wedge C_2 = 0 \wedge C_3 = 1$	then	$\Pr(E = 1) = 0.3$
If $C_1 = 0 \wedge C_2 = 0 \wedge C_3 = 0 \wedge C_4 = 1$	then	$\Pr(E = 1) = 0.6$
If $C_1 = 0 \wedge C_2 = 0 \wedge C_3 = 0 \wedge C_4 = 0$	then	$\Pr(E = 1) = 0.8$

If-Then Rules

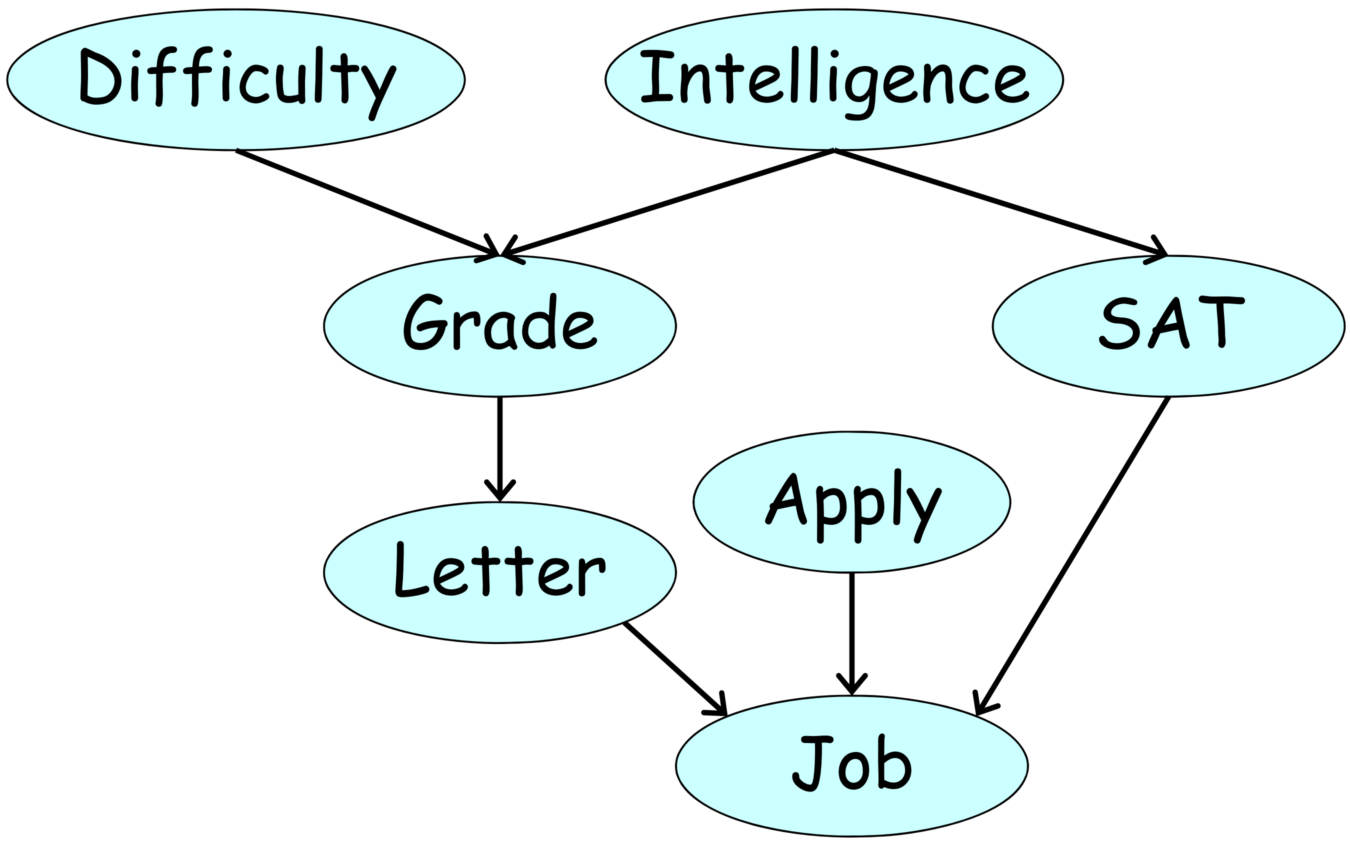
A CPT for variable E can be represented using a set of if-then rules of the form

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

For the rule-based representation to be complete and consistent

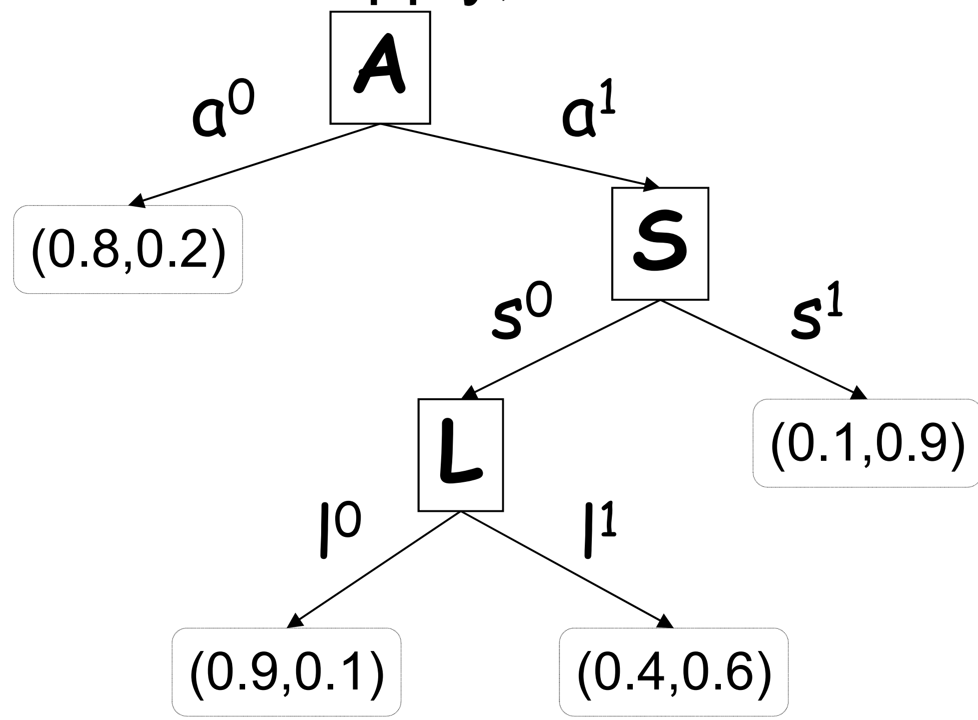
- The premises α_i must be mutually exclusive. That is, $\alpha_i \wedge \alpha_j$ is inconsistent for $i \neq j$. This ensures that the rules will not conflict with each other.
- The premises α_i must be exhaustive. That is, $\bigvee_i \alpha_i$ must be valid. This ensures that every CPT parameter $\theta_{e|\ldots}$ is implied by the rules.

A student's example



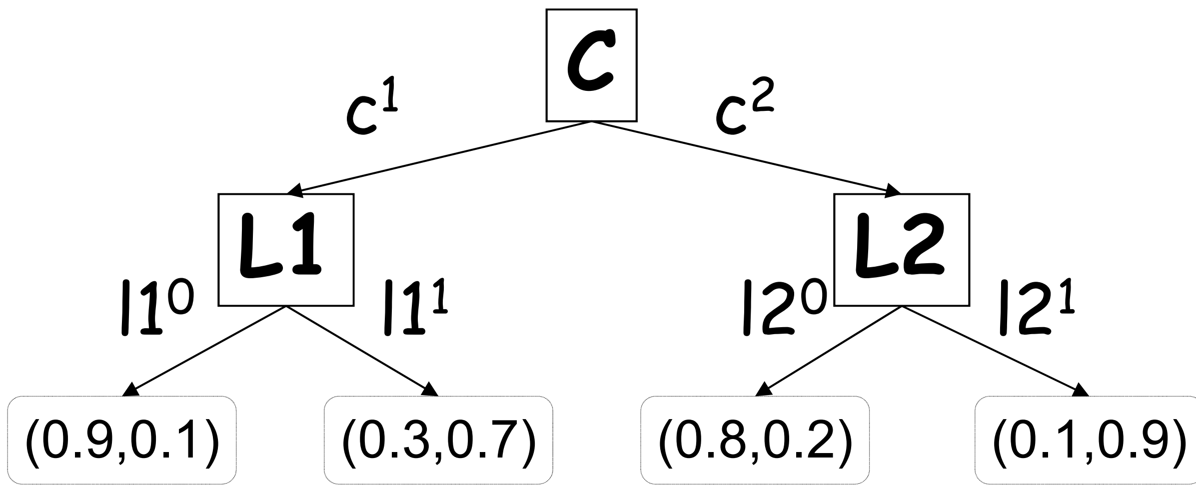
Tree CPD

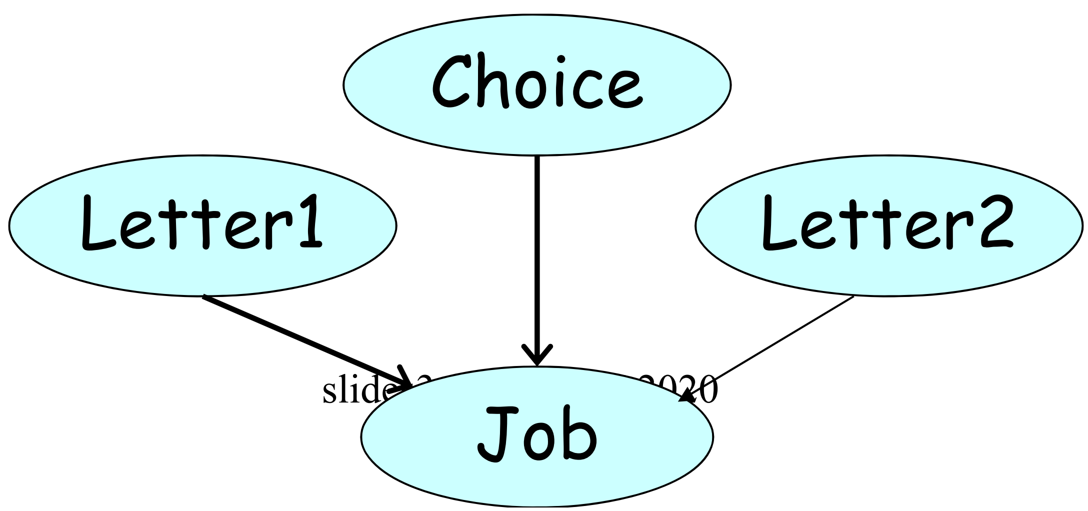
If the student does not *A*pply, *S*AT and *L* are irrelevant



Tree-CPD for job

Captures irrelevant variables

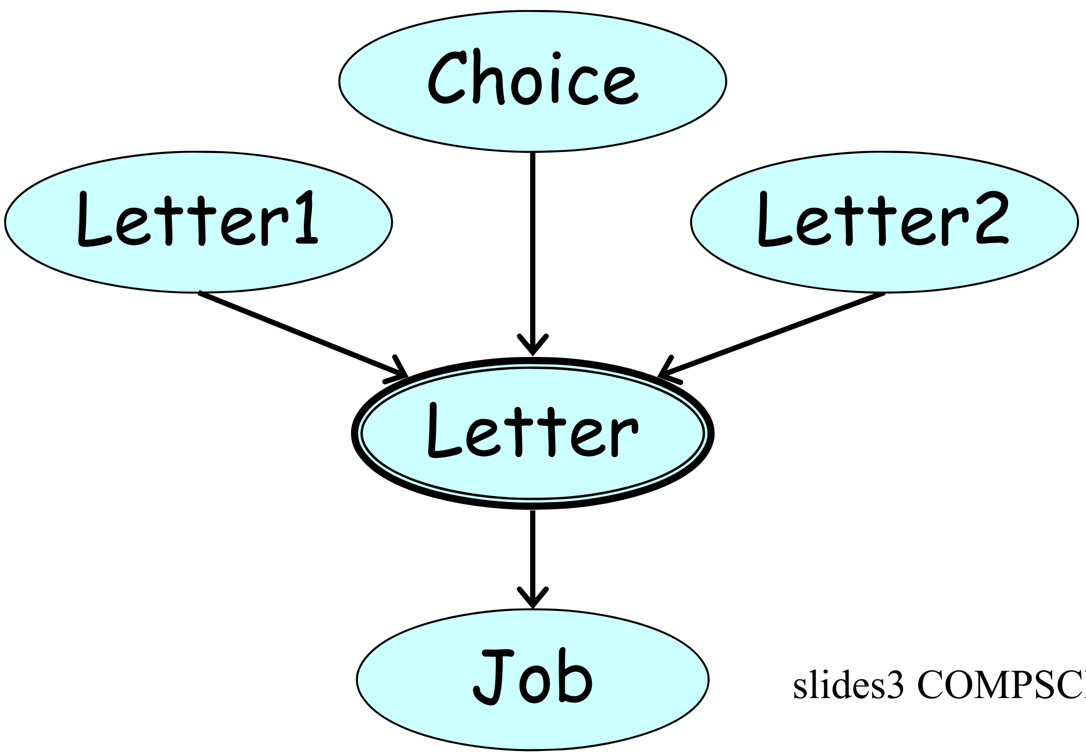




Multiplexer CPD

A CPD P(Y|A,Z1,Z2,…,Zk) is a multiplexer iff Val(A)=1,2,…k, and

P(Y|A,Z1,…Zk)=Z_a



Mixture of trees

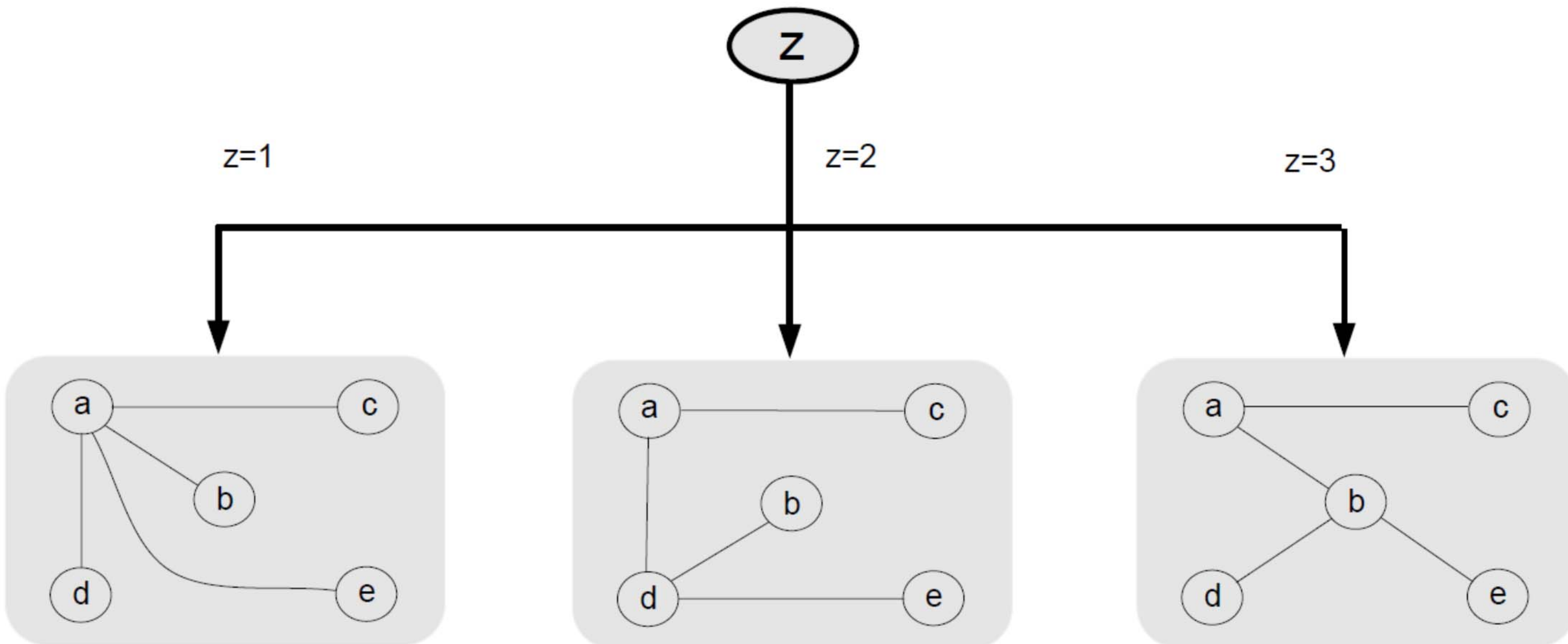


Figure 1: A mixture of trees over a domain consisting of random variables $V = \{a, b, c, d, e\}$, where z is a hidden *choice variable*. Conditional on the value of z, the dependency structure is a tree. A detailed presentation of the mixture-of-trees model is provided in Section 3.

Meila and Jordan, 2000

Mixture model with shared structure

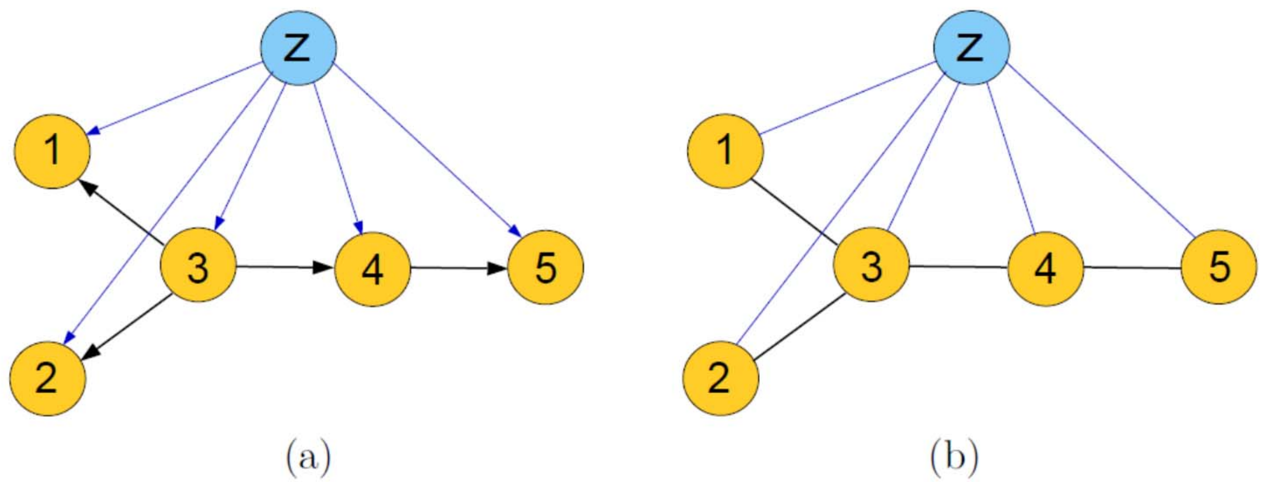


Figure 4: A mixture of trees with shared structure (MTSS) represented as a Bayes net (a) and as a Markov random field (b).

Meila and Jordan, 2000

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

$$(X = \text{ok} \wedge A = \text{high}) \vee X = \text{stuckat0} \iff C = \text{low}$$
$$(X = \text{ok} \wedge A = \text{low}) \vee X = \text{stuckat1} \iff C = \text{high}$$

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

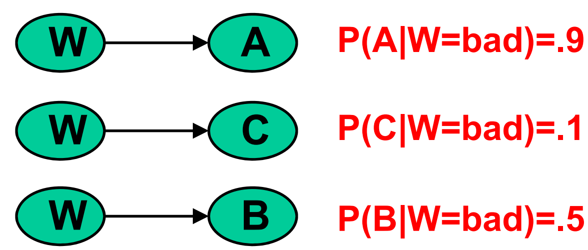
Probabilistic Reasoning

Party example: the weather effect

Alex is-likely-to-go in bad weather

Chris rarely-goes in bad weather

Becky is indifferent but unpredictable

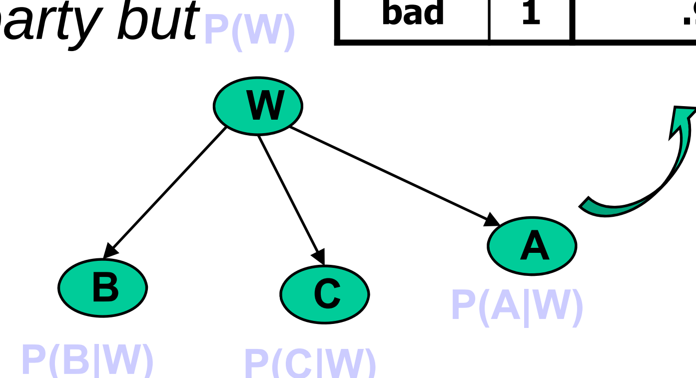


Questions:

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?

W	A	P(A|W)
good	0	.01
good	1	.99
bad	0	.1
bad	1	.9

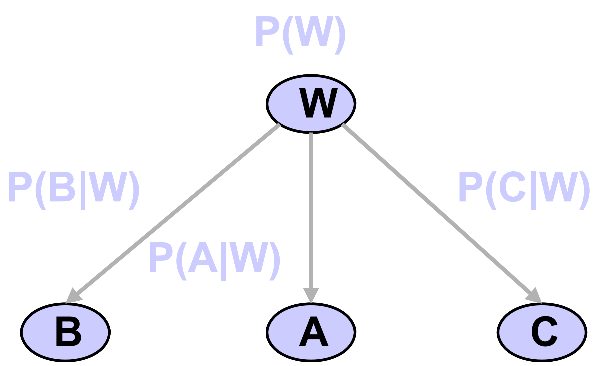


$P(W,A,C,B) = P(B|W) \cdot P(C|W) \cdot P(A|W) \cdot P(W)$

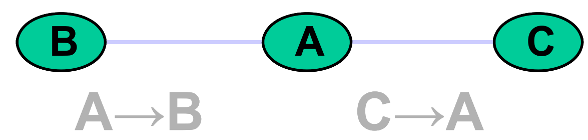
$P(A,C,B|W=bad) = 0.9 \cdot 0.1 \cdot 0.5$

Party Example Again

Bayes Network



Constraint Network



Semantics?

Algorithms?

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

$$P(C, \neg B \mid w = bad, A \rightarrow B, C \rightarrow A)$$