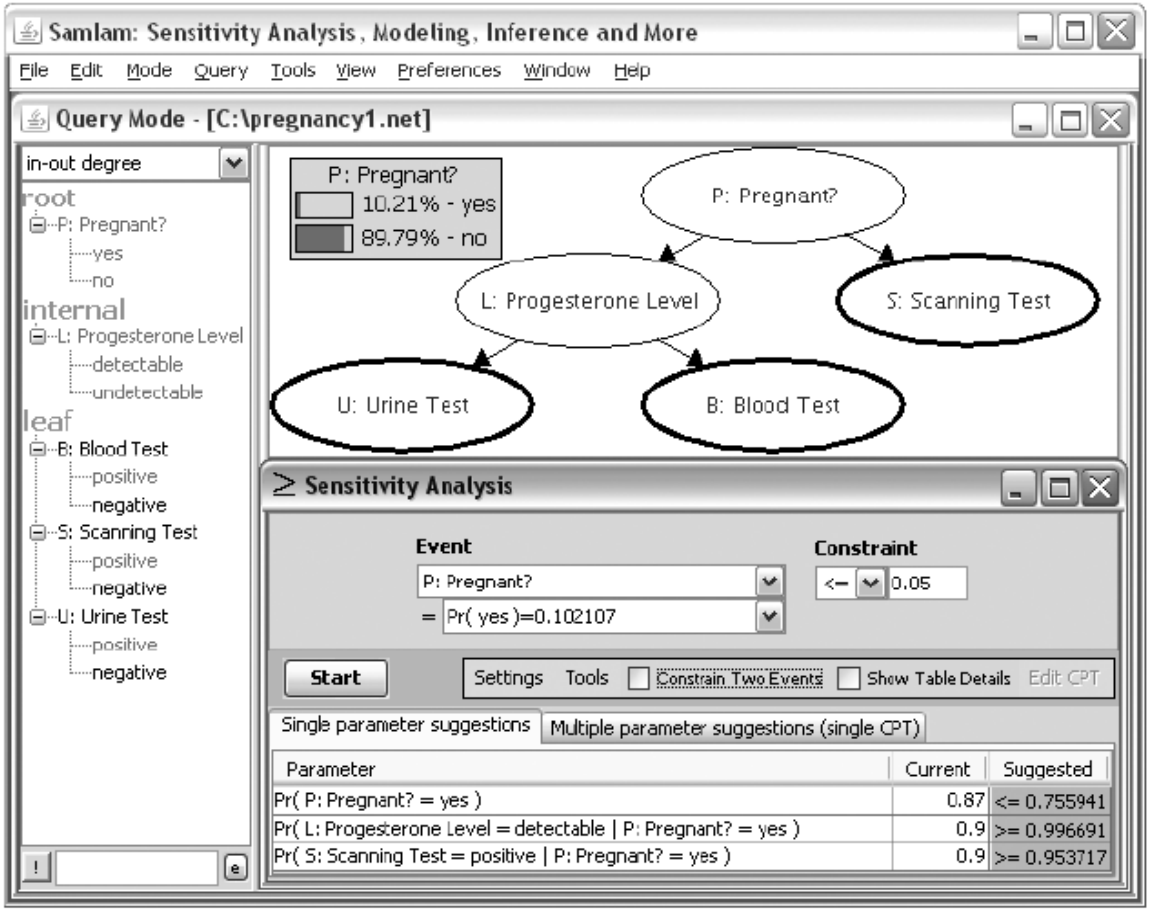
Sensitivity Analysis



Example

Which network parameter do we have to change, and by how much, so as to ensure that the probability of pregnancy would be no more than 5% given three negative tests?

Sensitivity Analysis

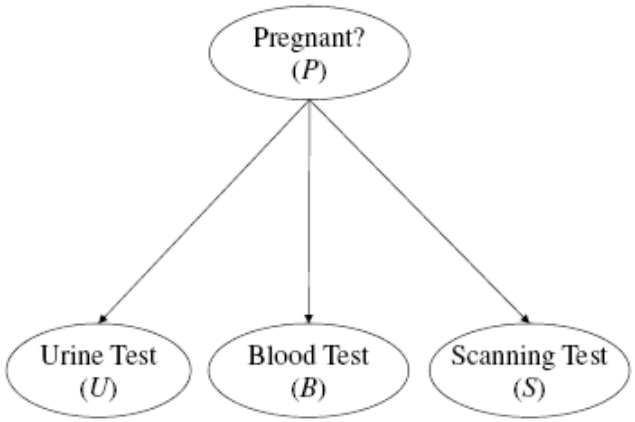
Possible (single) parameter changes:

1. If the false negative rate for the scanning test were about 4.63% instead of 10%.
2. If the probability of pregnancy given insemination were about 75.59% instead of 87%.
3. If the probability of a detectable progesterone level given pregnancy were about 99.67% instead of 90%.

The last two changes are not feasible since the farmer does not intend to change the insemination procedure, nor does he control the progesterone level.

Network Granularity

We can now build the following network in which the progesterone level is no longer represented explicitly.



P	θ_p
yes	.87

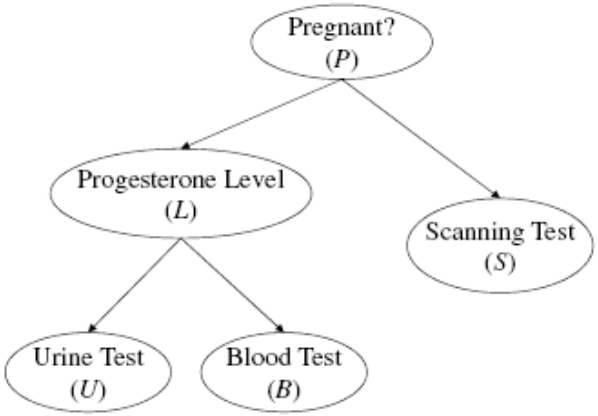
P	S	$\theta_{s\|p}$
yes	–ve	.10
no	+ve	.01

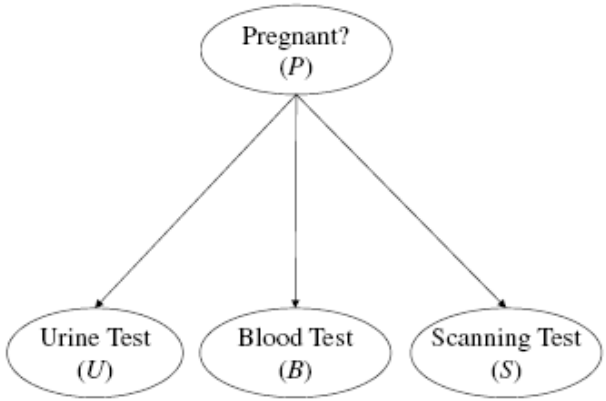
P	B	$\theta_{b\|p}$
yes	–ve	.36
no	+ve	.106

P	U	$\theta_{u\|p}$
yes	–ve	.27
no	+ve	.107

The question now is whether this simpler network is equivalent to the original one from the viewpoint of answering queries.

Network Granularity





Naive Bayes: blood and urine tests independent given pregnancy

Probability of pregnancy given two negative tests is about 45.09%, given two positive tests is about 99.61%.

Original structure

Probability of pregnancy given these two negative tests is 52.96%, given two positive tests is about 99.54%

Network Granularity

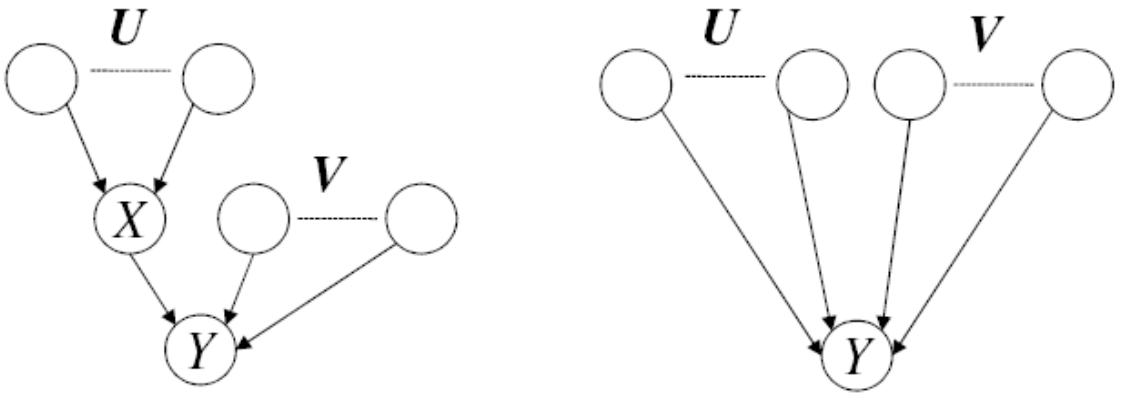
Bypassing a variable

Removing the variable, redirecting its parents to its children, and then updating the CPTs of these children (as they now have different parents).

Model accuracy

$\Pr'(.)$ is the distribution after bypassing a variable in $\Pr(.)$. The bypass procedure does not affect model accuracy in case $\Pr(\mathbf{q}, \mathbf{e}) = \Pr'(\mathbf{q}, \mathbf{e})$ for all instantiations of query variables **Q** and evidence variables **E**.

Network Granularity



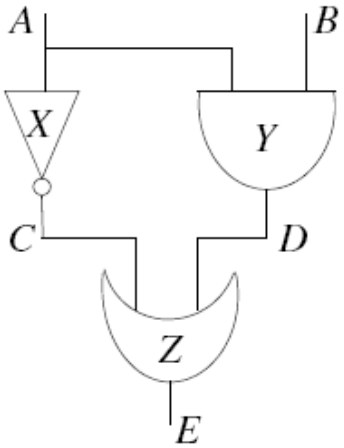
Variable X can be bypassed if it has a single child Y

The CPT for variable Y must be updated: $\theta'_{y|\mathbf{uv}} = \sum_x \theta_{y|x\mathbf{v}}\theta_{x|\mathbf{u}}$.

U are the parents of variable X.

V are the parents of variable Y other than X.

Diagnosis III: Model from Design

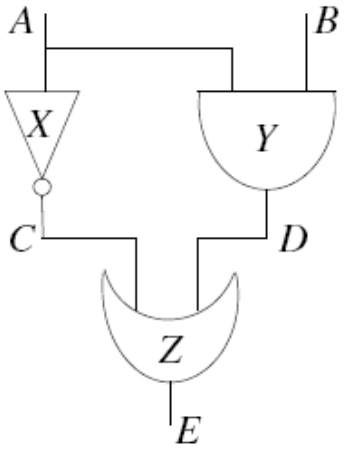


Problem statement

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



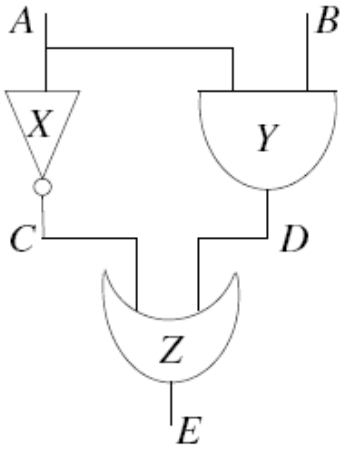
Problem statement

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.

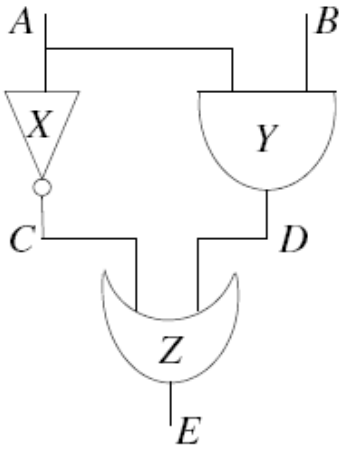
Evidence variables

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

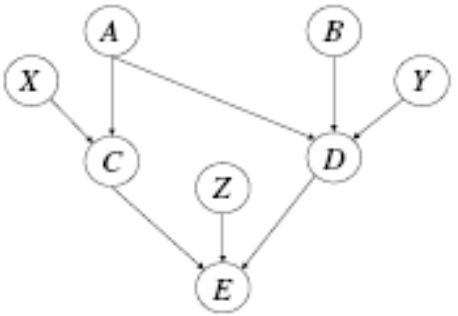
Primary inputs and output of the circuit, A, B and E.

Query variables

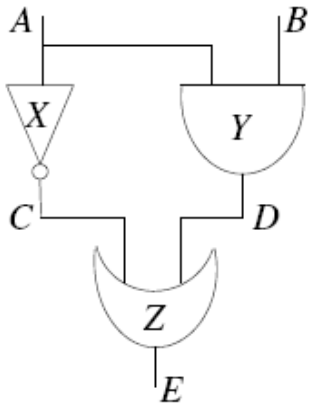
Health of components X, Y and Z.

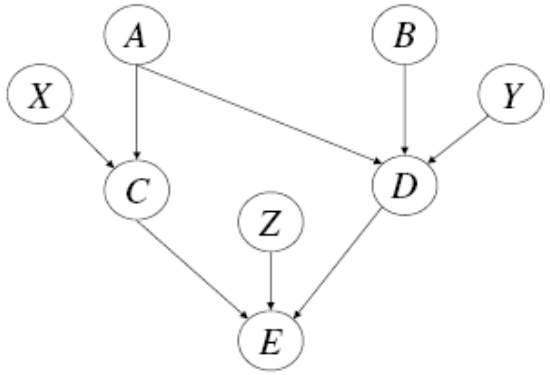
Intermediary variables

Internal wires, C and D.



Diagnosis III: Model from Design

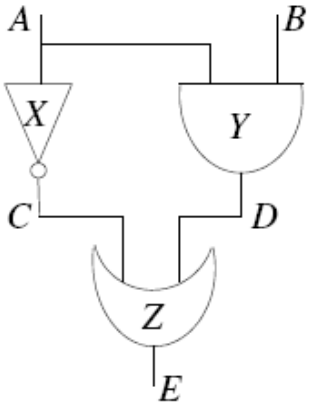


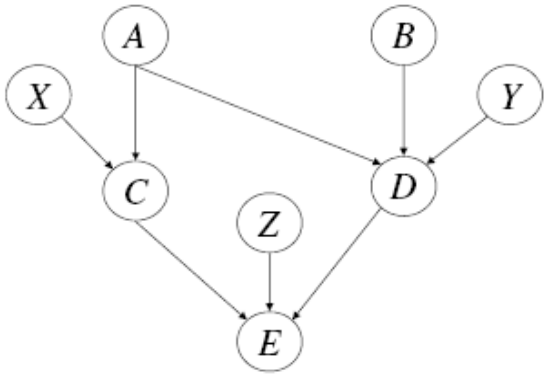


Function blocks

The outputs of each block are determined by its inputs and its state of health.

Diagnosis III: Model from Design





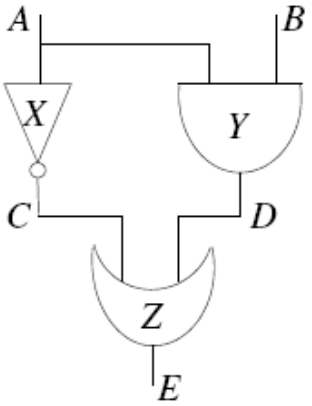
Function blocks

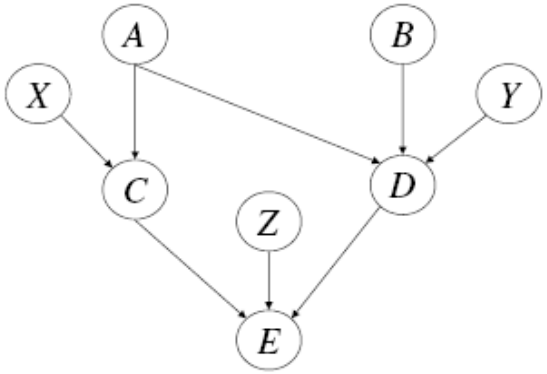
The outputs of each block are determined by its inputs and its state of health.

Primary inputs

No direct causes for primary inputs, A and B: no parents.

Diagnosis III: Model from Design





Function blocks

The outputs of each block are determined by its inputs and its state of health.

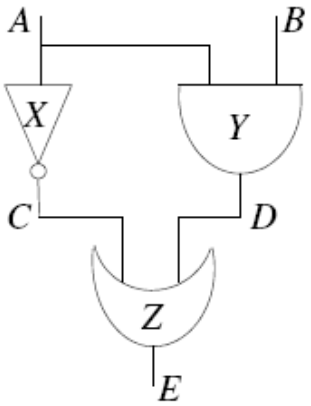
Primary inputs

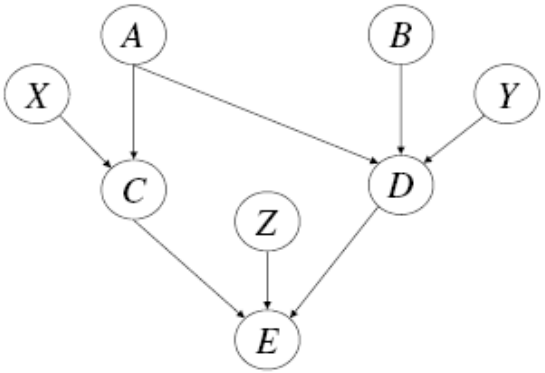
No direct causes for primary inputs, A and B: no parents.

Health states

No direct causes for health of X, Y and Z: no parents.

Diagnosis III: Model from Design





Function blocks

The outputs of each block are determined by its inputs and its state of health.

Primary inputs

No direct causes for primary inputs, A and B: no parents.

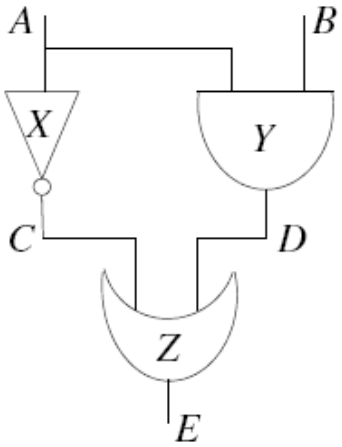
Health states

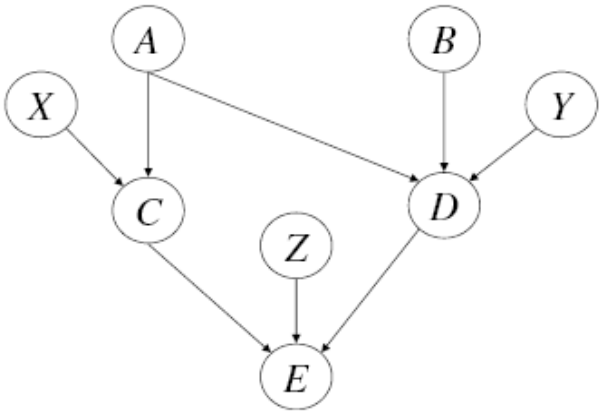
No direct causes for health of X, Y and Z: no parents.

Gate output D

Direct causes of D are gate inputs, A and B, and health of Y.

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	$\theta_{c\mid a,x}$
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 $\mathbf{e}$: $A = \text{high}$, $B = \text{high}$, $E = \text{low}$, compute MAP over health variables X, Y and Z.

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:

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

Network with fault modes gives two MAP instantiations:

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

Network with no fault modes gives two MAP instantiations:

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

Posterior Marginals

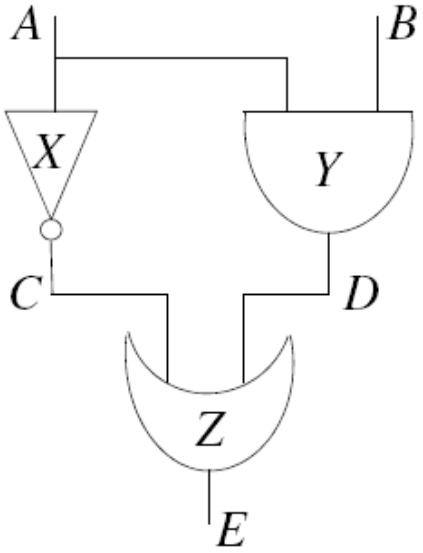
Consider the posterior marginals over the health variables X, Y, Z:

State	X	Y	Z	$\Pr(X, Y, Z\|\mathbf{e})$
1	ok	ok	ok	0
2	faulty	ok	ok	0
3	ok	faulty	ok	.49374
4	ok	ok	faulty	.49374
5	ok	faulty	faulty	.00499
6	faulty	ok	faulty	.00499
7	faulty	faulty	ok	.00249
8	faulty	faulty	faulty	.00005

- State 2 is impossible.
- Y and Z more likely to be faulty together than Y and X.
- States with faulty Z more likely than states with faulty Y:

$$\Pr(Z = \text{faulty}|\mathbf{e}) \approx 50.38\% > \Pr(Y = \text{faulty}|\mathbf{e}) \approx 50.13\%.$$

Lack of Symmetry for Double Faults



Test vector

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

- If Y and Z are faulty, we have two possible states for C and D: $C = \text{low}$, D either low or high.
- If Y and X are faulty, we have only one possible state for C and D: $C = \text{low}$ and $D = \text{low}$.

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

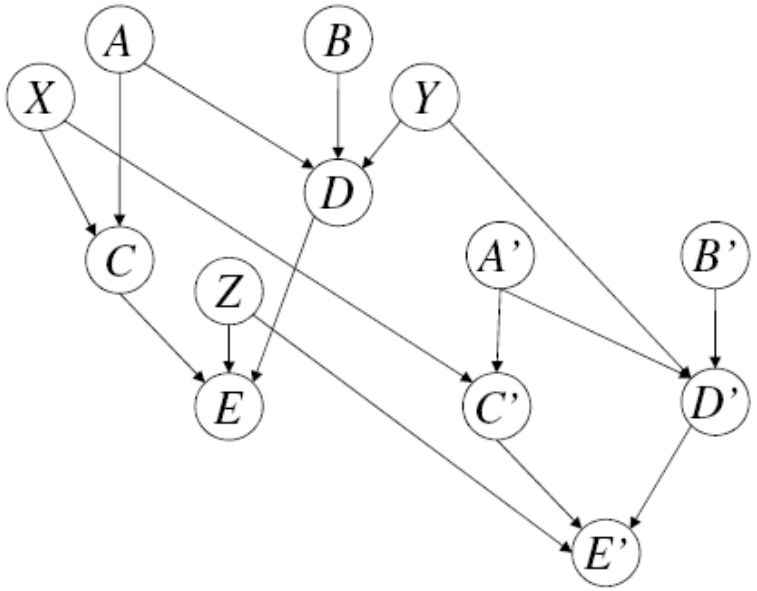
C' and D'

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

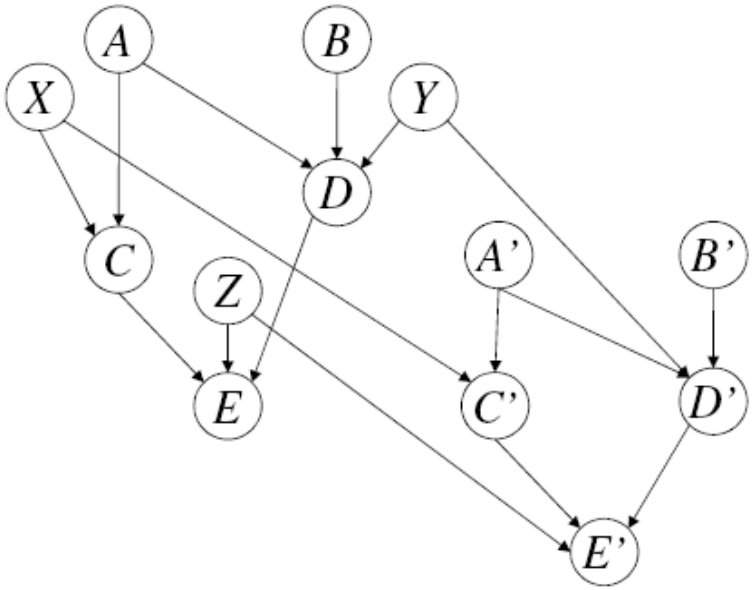


Two test vectors

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

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

Integrating Time: No Intermittent Faults



Two test vectors

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

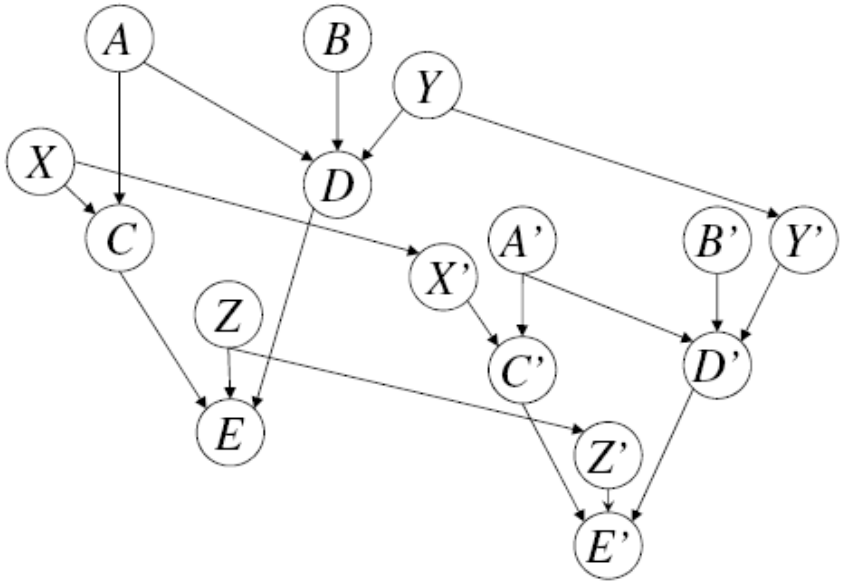
$\mathbf{e}'$: $A = \text{low}$, $B = \text{low}$, $E = \text{low}$.

MAP using second structure

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

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

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

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

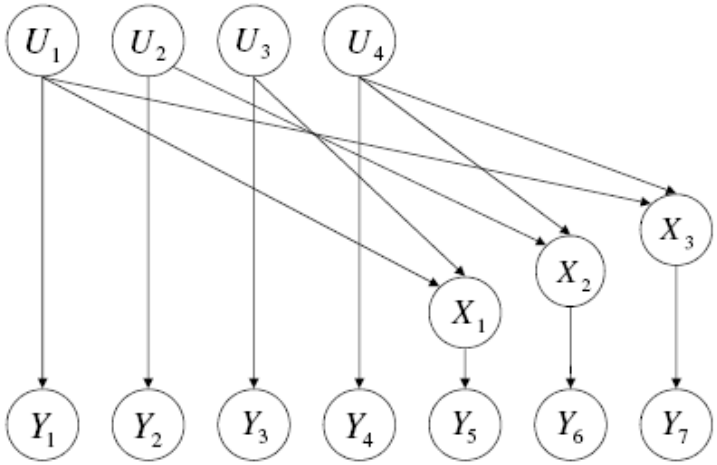
$Y_1, \ldots, Y_7$: bits received at destination D

Query variables are

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

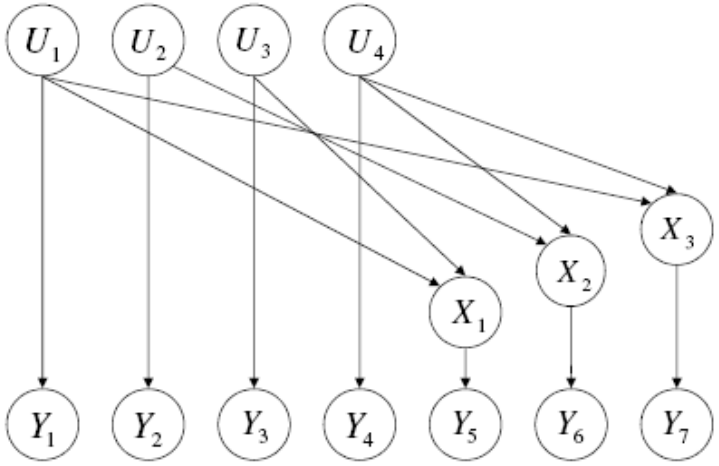
Bits $X_1, \ldots, X_3$ either query variables or intermediary variables.

Channel Coding



There are three CPT types in the problem.

Channel Coding



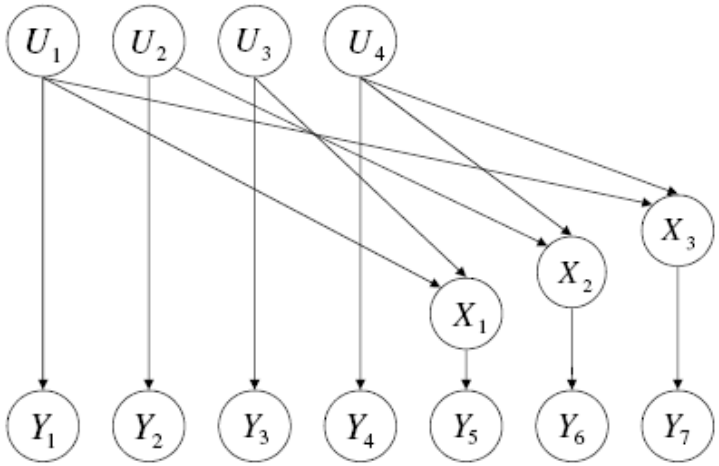
There are three CPT types in the problem.

CPT for each redundant bit, say X_1:

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

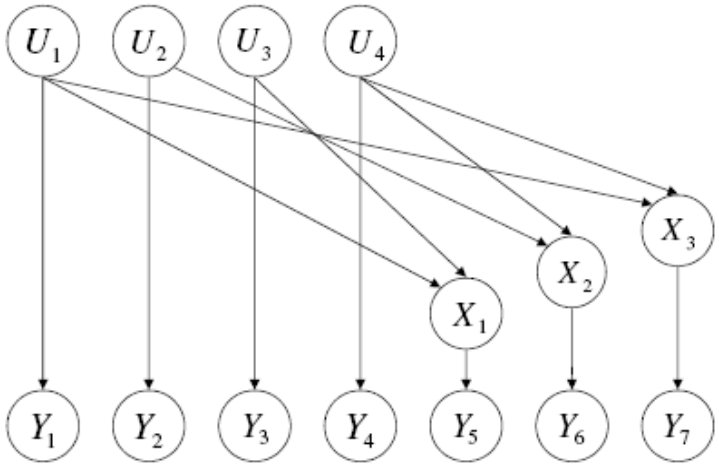
$\Pr(x_1|u_1, u_3) = 1$ iff $x_1 = u_1 \oplus u_3$ ($\oplus$ is the XOR function)

Channel Coding



There are three CPT types in the problem.

Channel Coding



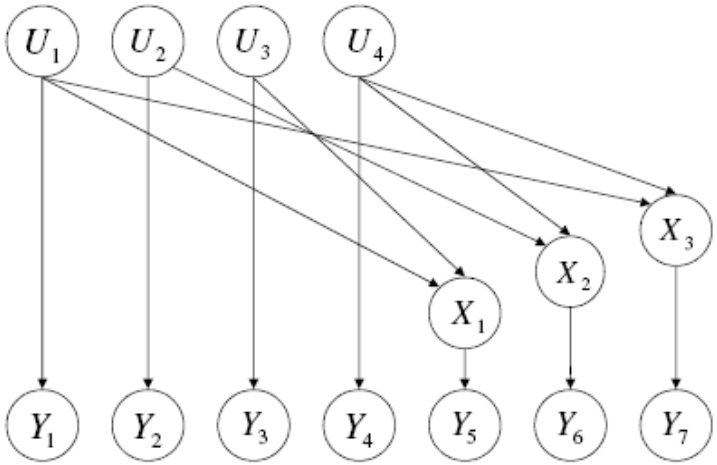
There are three CPT types in the problem.

CPT for a channel output bit, say Y_1:

U_1	Y_1	$\theta_{y_1 \mid u_1}$
1	0	.01
0	1	.01

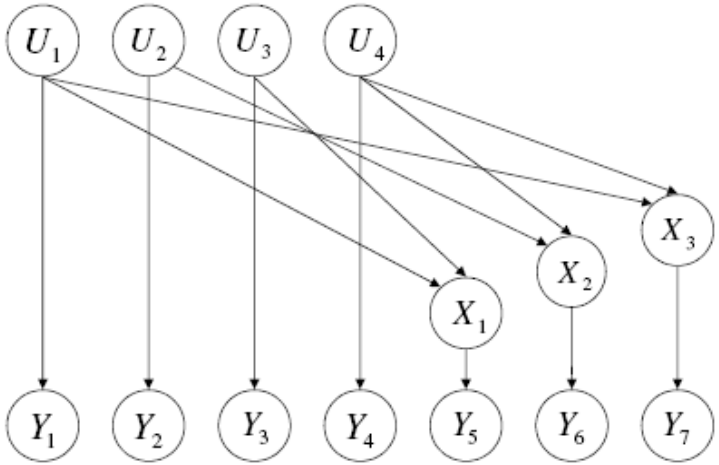
CPT captures the simple noise model given in the problem statement.

Channel Coding



There are three CPT types in the problem.

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

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

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 $k = e^{(1-2y_i)/2\sigma^2}$

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 $k = e^{(1-2y_i)/2\sigma^2}$

Example

If $\sigma = .5$ and channel output $y_i = .1$, we interpret as a soft evidence on channel input $X_i = 0$ with a Bayes factor $k \approx 5$.

Convolutional Codes

Convolutional and turbo codes

correspond to different methods for generating redundant bits.

Convolutional Codes

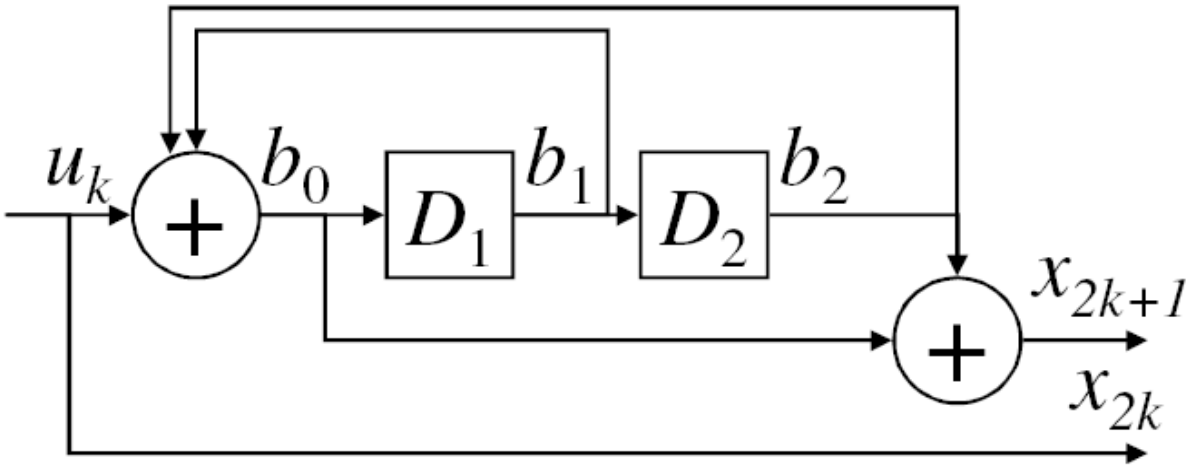
Convolutional and turbo codes

correspond to different methods for generating redundant bits.

Convolutional and turbo codes

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

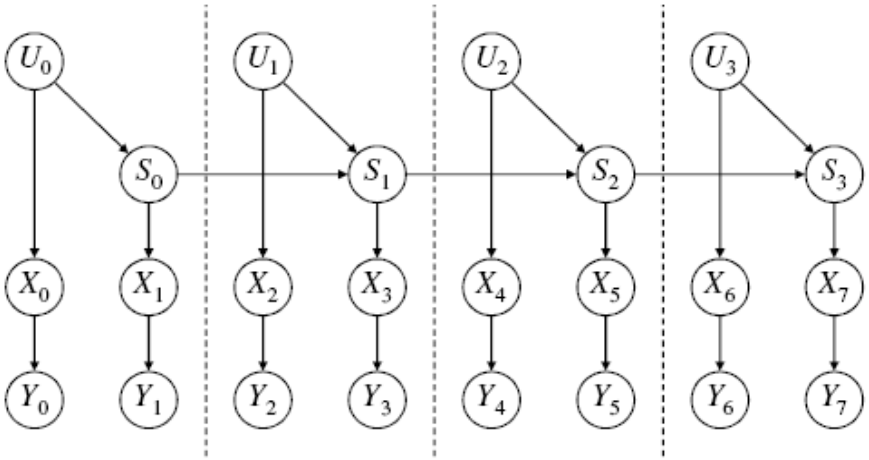
Convolutional Codes



An example convolutional encoder

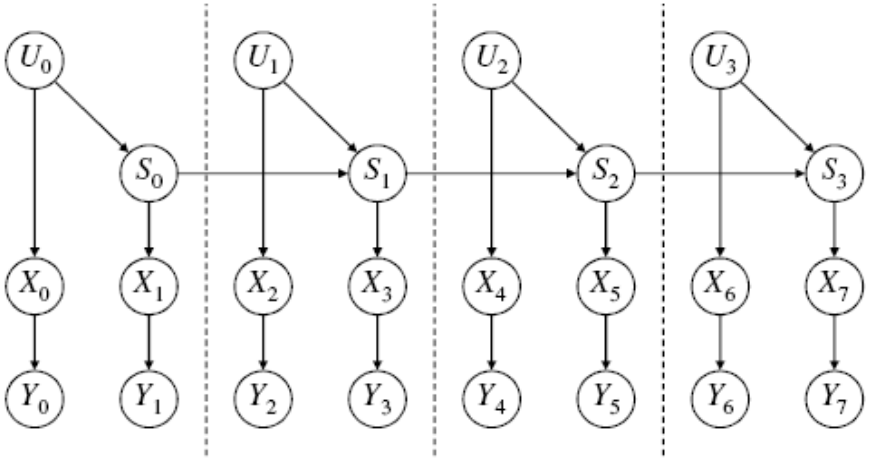
Each node denoted with a "+" represents a binary addition, and each box D_i represents a delay where the output of D_i is the input of D_i from the previous encoder state.

Convolutional Codes



Dynamic Bayesian network for a convolutional code.

Convolutional Codes

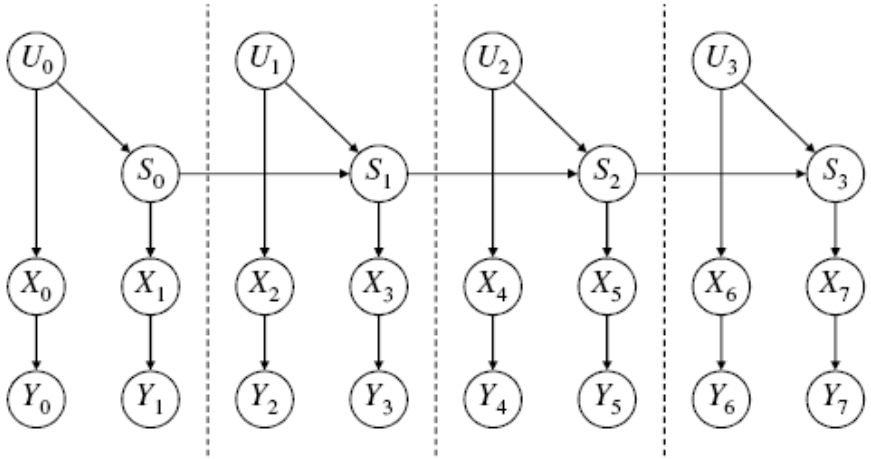


Dynamic Bayesian network for a convolutional code.

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 $u_0, \ldots, u_3$.

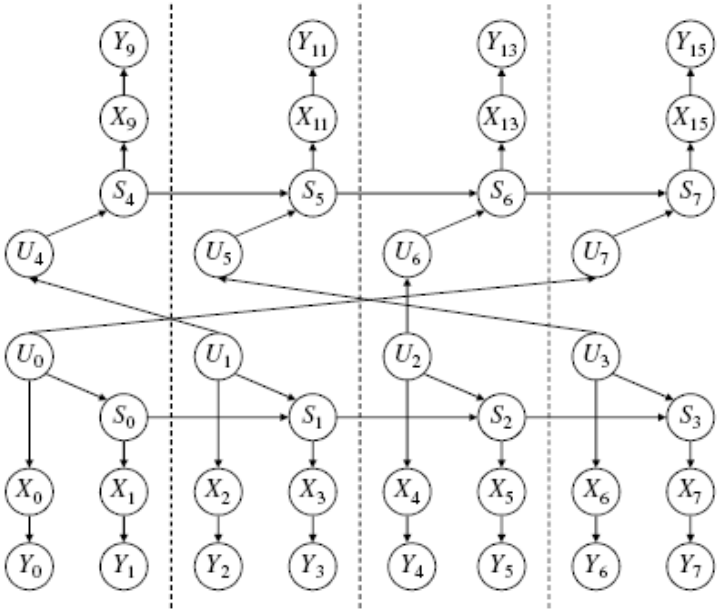
In a convolutional code

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



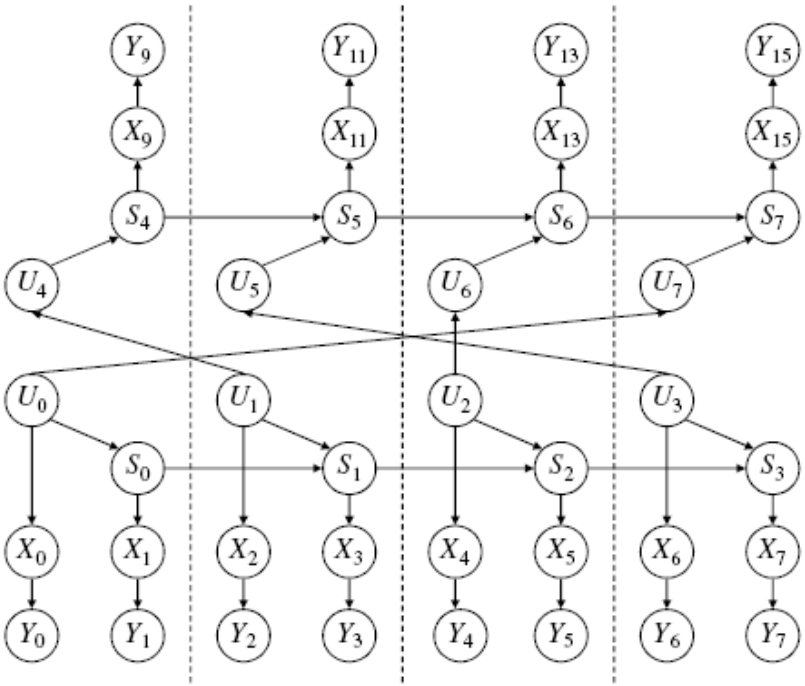
Lower network represents a convolutional code

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

Upper network represents a convolutional code

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

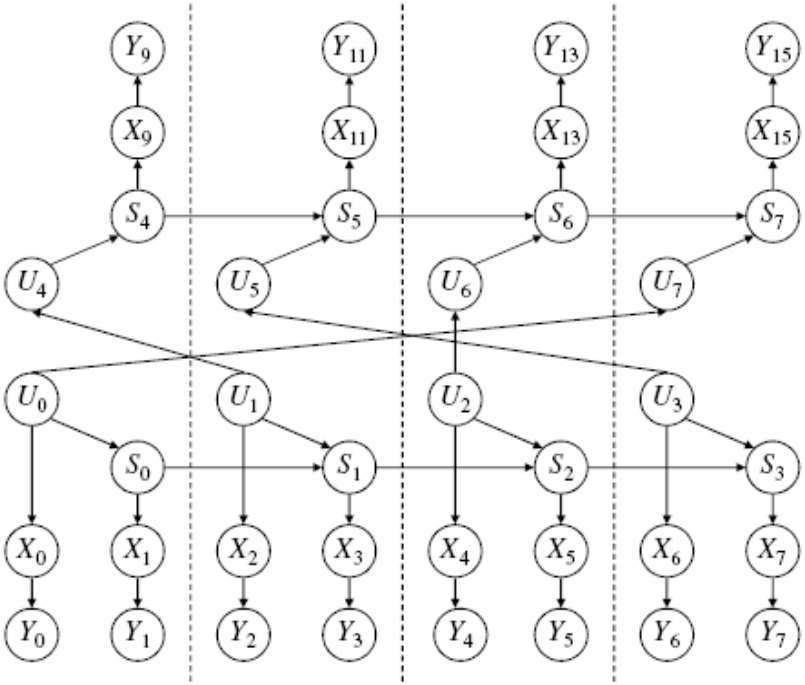
Turbo Codes



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).

Turbo Codes

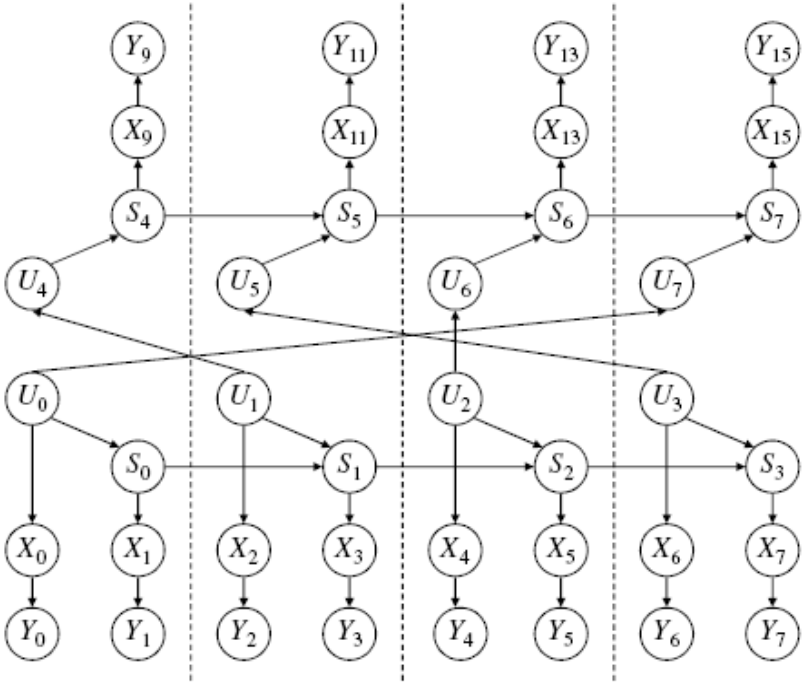


CPTs for the bit sequence $u_4, \ldots, u_7$

$$\theta_{u_k|u_j} = \begin{cases} 1, & \text{if } u_k = u_j; \\ 0, & \text{otherwise.} \end{cases}$$

Establishes equivalence between U_k in the upper network and U_j in

Turbo Codes

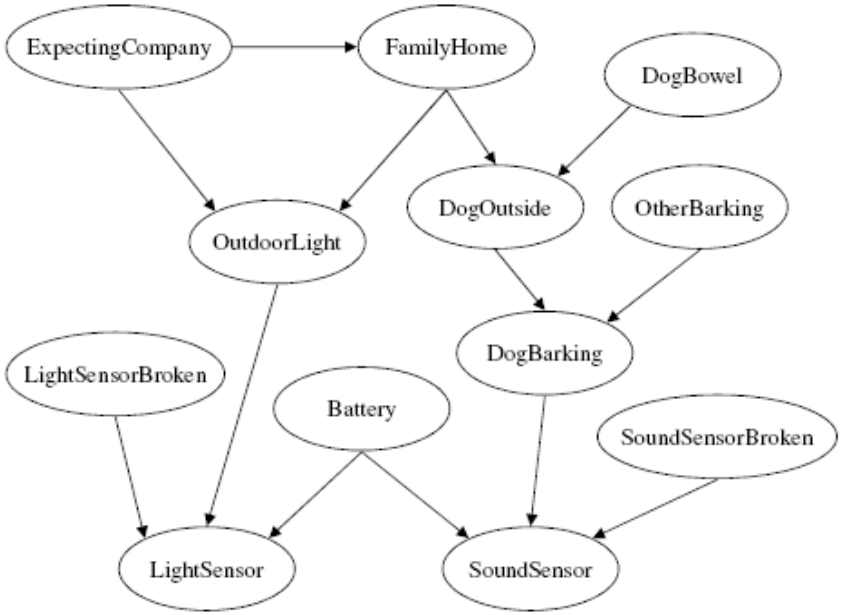


Networks corresponding to convolutional codes are

singly-connected: there is only one (undirected) path between any two variables in the network.

Networks corresponding to turbo codes are Multiply-connected

Commonsense Knowledge

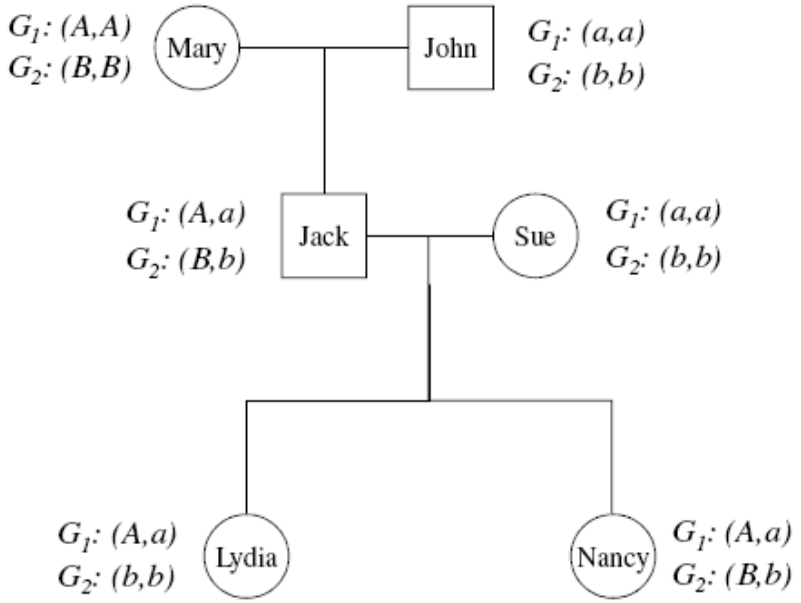


Parameters based on a combination of sources

- Statistical information such as reliabilities of sensors and battery.
- Subjective beliefs relating to how often the wife goes out, guests are expected, the dog has bowel trouble, etc.
- Objective beliefs regarding the functionality of sensors.

Genetic Linkage Analysis

Variables, values, structure?



A pedigree involving six individuals

Squares represent males, circles represent females. Horizontal edges connect spouses, while vertical edges connect couples to their children. For example, Jack and Sue are a couple with two daughters, Lydia and Nancy.

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

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).

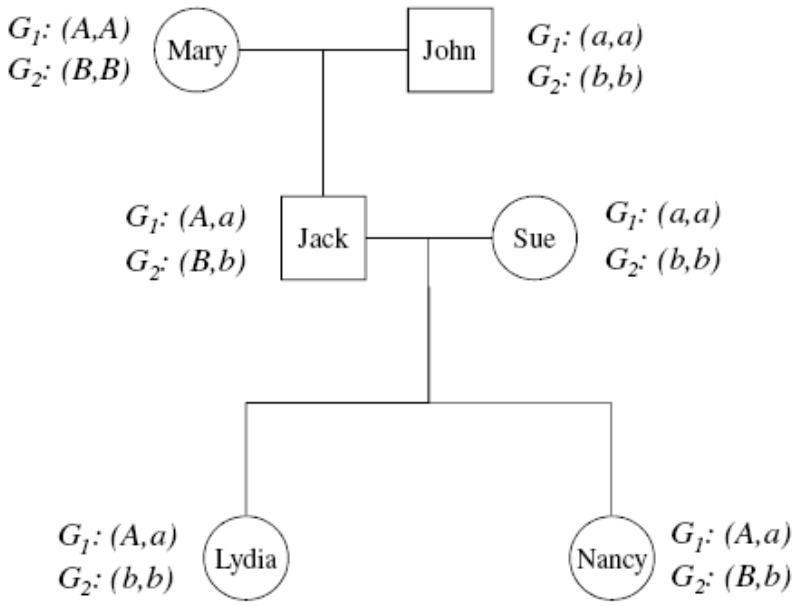
Example

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

Recombination Events

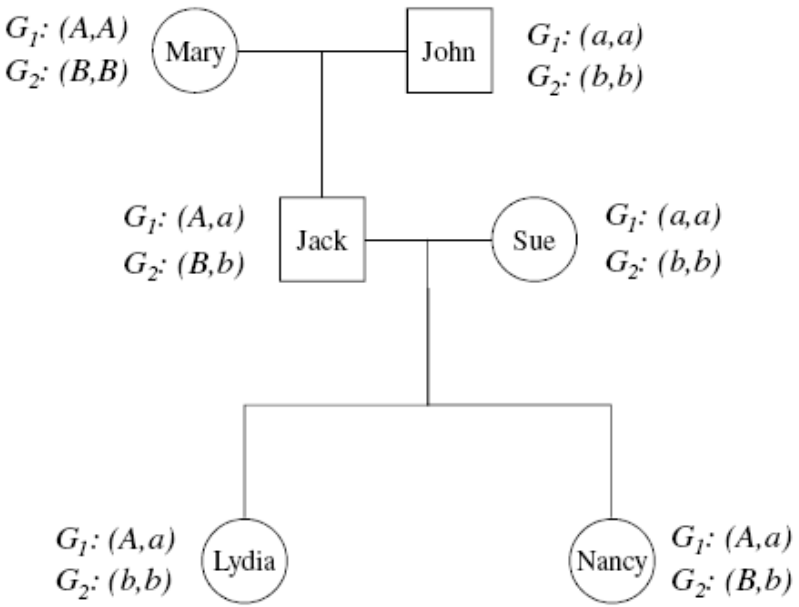
Haplotype

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.

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

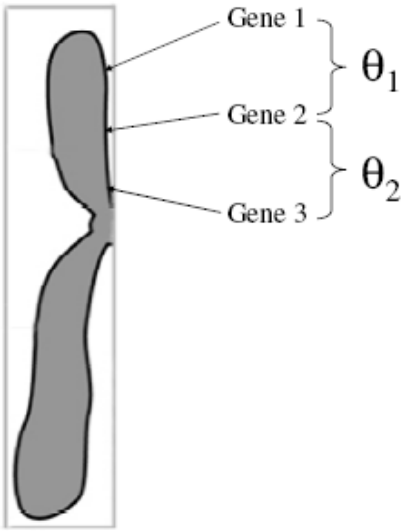
The extent to which genes G_1 and G_2 are linked

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 $\theta < 1/2$.

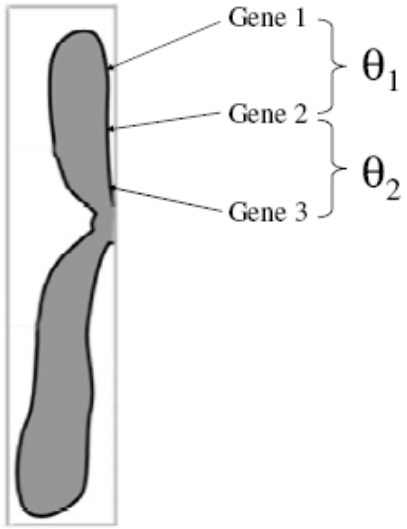
Genetic Linkage and Gene Maps



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).

Genetic Linkage and Gene Maps



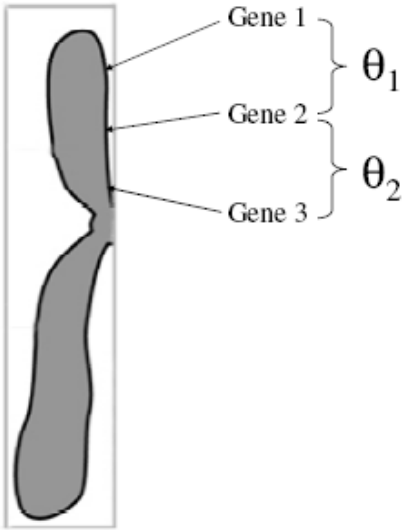
Linkage between genes

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.

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

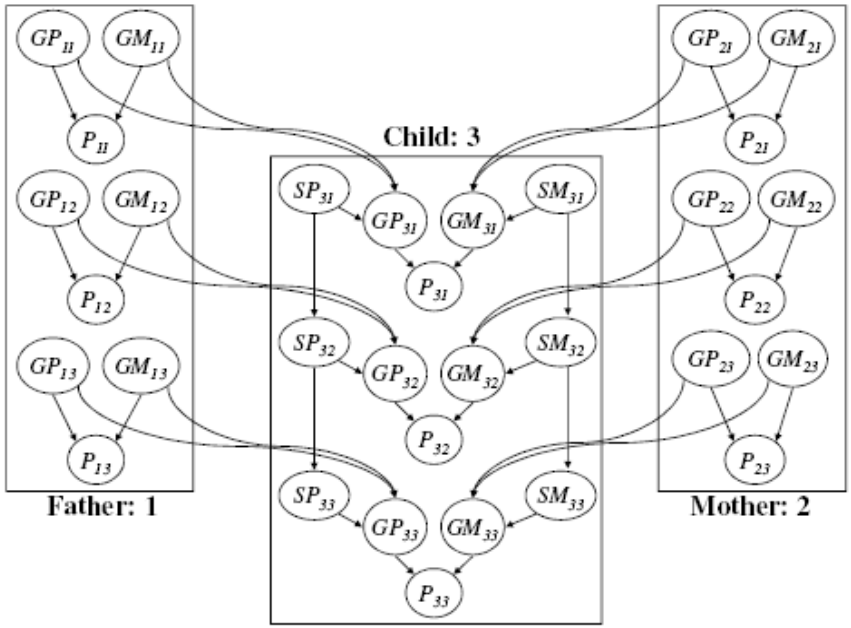
on the distance between genes on a chromosome.

The Likelihood of a Hypothesis

Given a pedigree, together with some information about the genotypes and phenotypes of involved individuals

we want to develop a Bayesian network which can be used to assess the likelihood of a particular recombination frequency.

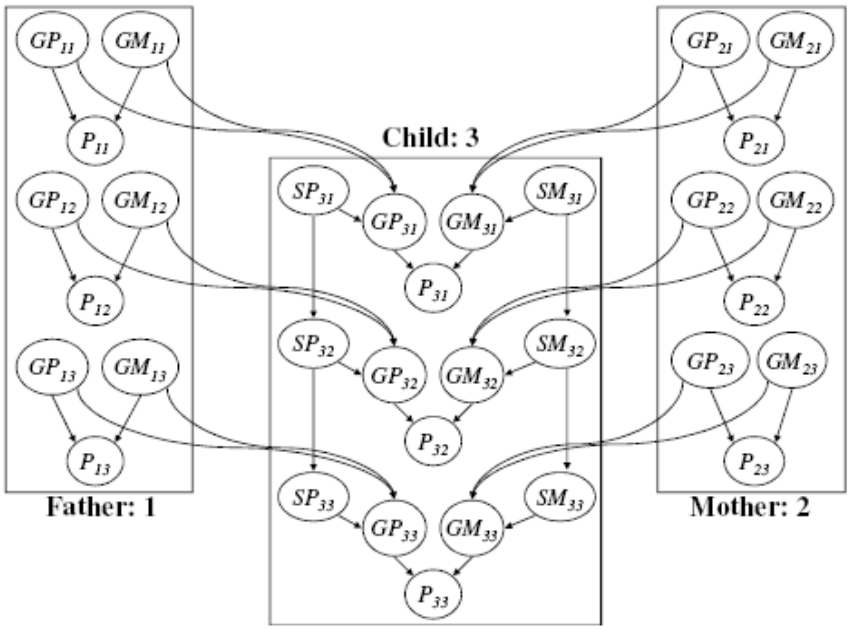
From Pedigrees to Bayesian Networks



A Bayesian network structure corresponding to a simple pedigree

involving three individuals numbered $1, 2$ and 3. Each individual has three genes numbered $1, 2$ and 3, which are assumed to be in this order on a chromosome.

From Pedigrees to Bayesian Networks



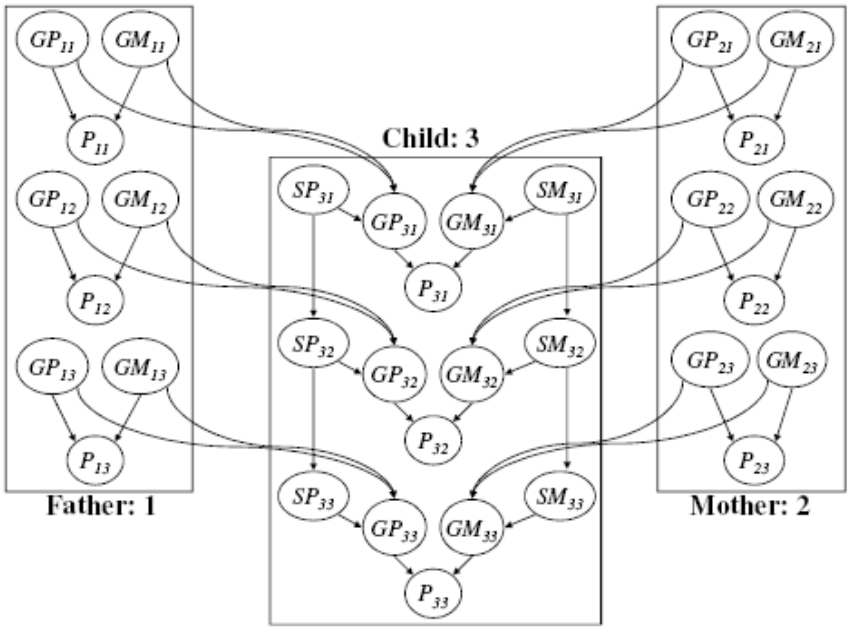
Genotype and phenotype

—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

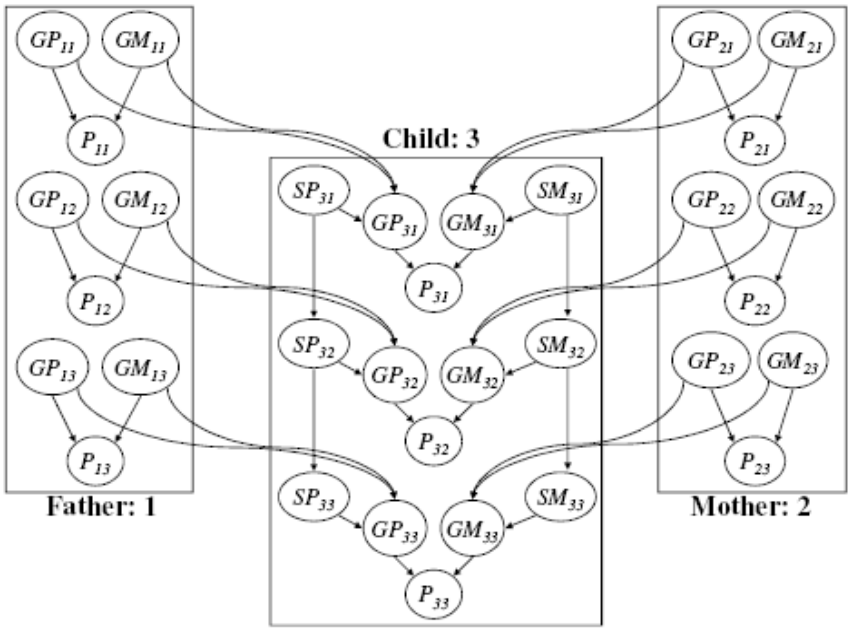


Selector variables

—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

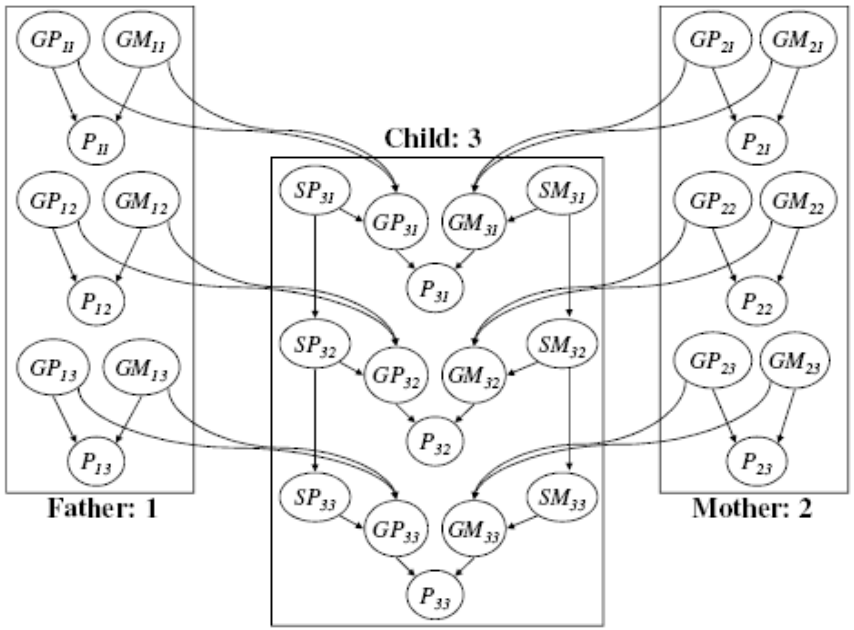
If $SP_{ij} = p$ then individual i will inherit the allele of gene j

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.

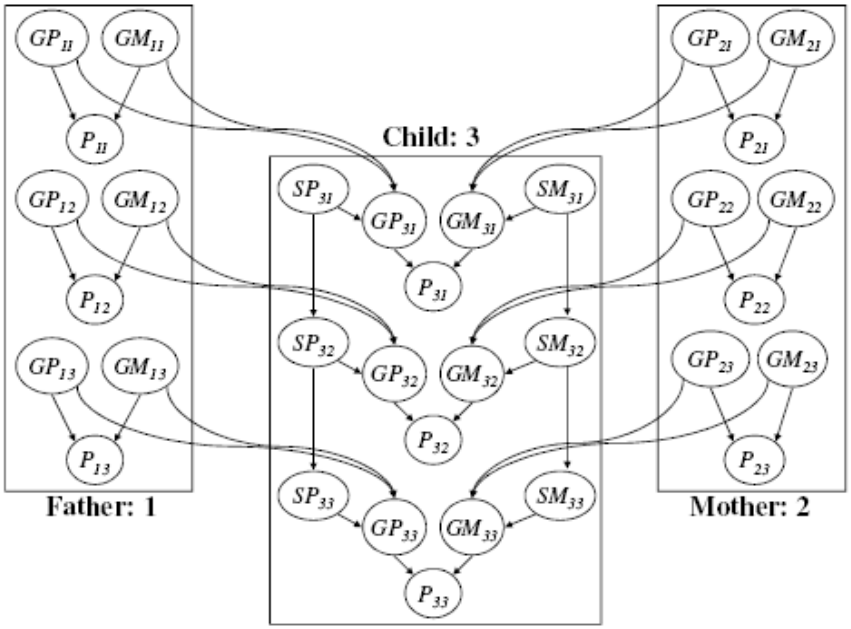
From Pedigrees to Bayesian Networks



For each founder i and gene j, the CPTs for genotype variables GP_{ij} and GM_{ij}

are usually obtained from population statistics collected by geneticists.

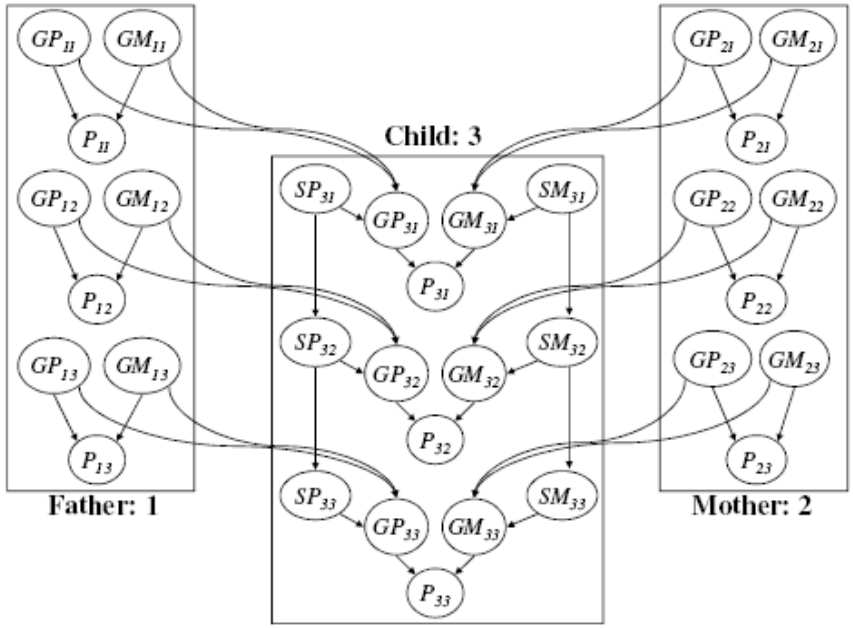
From Pedigrees to Bayesian Networks



For each individual i and gene j, the CPT for the phenotype P_{ij}

may be deterministic or probabilistic as we have seen earlier.

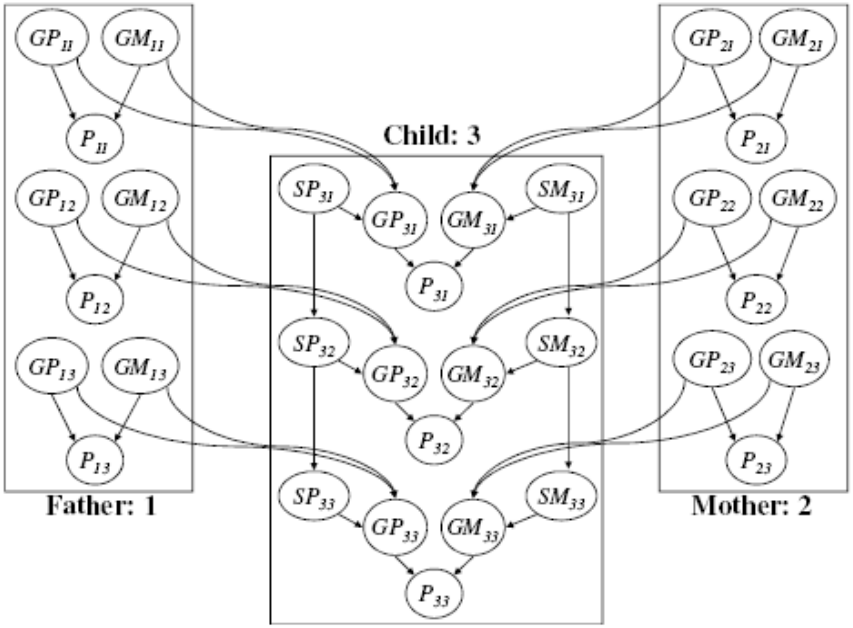
From Pedigrees to Bayesian Networks



For each non-founder i and gene j, the CPTs for genotype variables GP_{ij} and GM_{ij}

follow deterministically from the semantics of selector variables.

From Pedigrees to Bayesian Networks



If individual i has father k, the CPT for GP_{ij} is given by

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

From Pedigrees to Bayesian Networks

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

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}

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

CPTs of selector variables

host our hypotheses about recombination frequencies.

From Pedigrees to Bayesian Networks

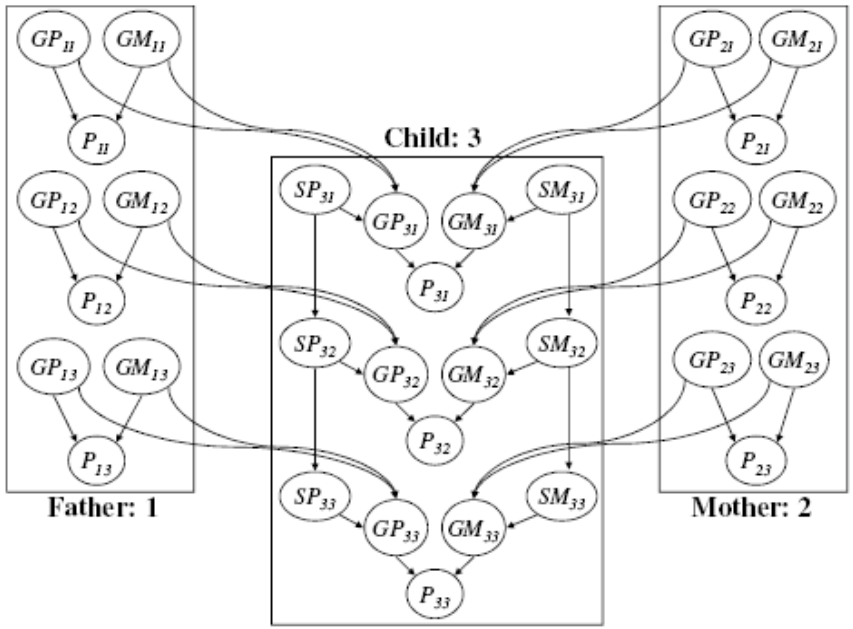
CPTs of selector variables

host our hypotheses about recombination frequencies.

To produce a distance map for genes

we need the distance between genes 1 and 2, and the distance between genes 2 and 3 which are indicated by recombination frequencies θ_{12} and θ_{23}.

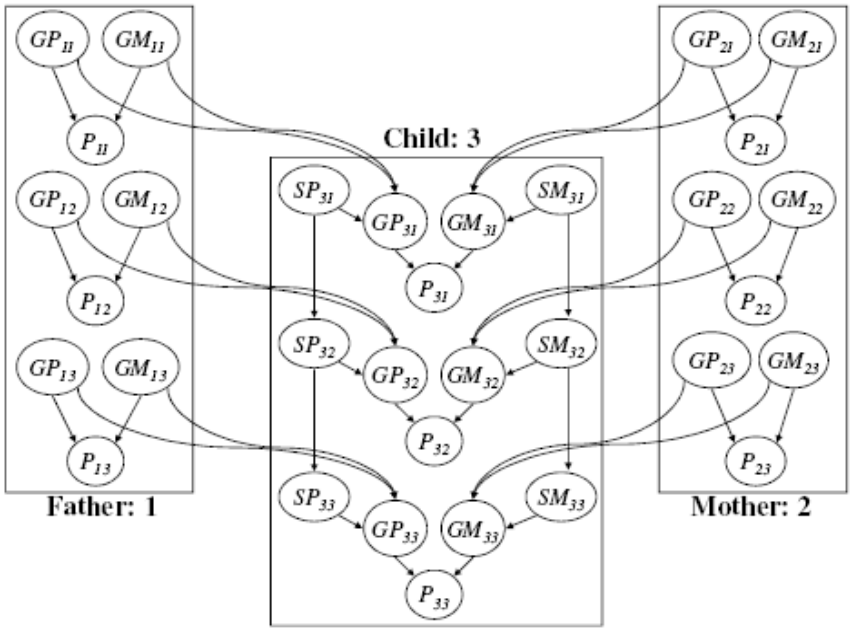
From Pedigrees to Bayesian Networks



Selectors of first gene SP_{31} and SM_{31} have uniform CPTs

This means that parents pass paternal or maternal alleles with equal probability for this gene.

From Pedigrees to Bayesian Networks



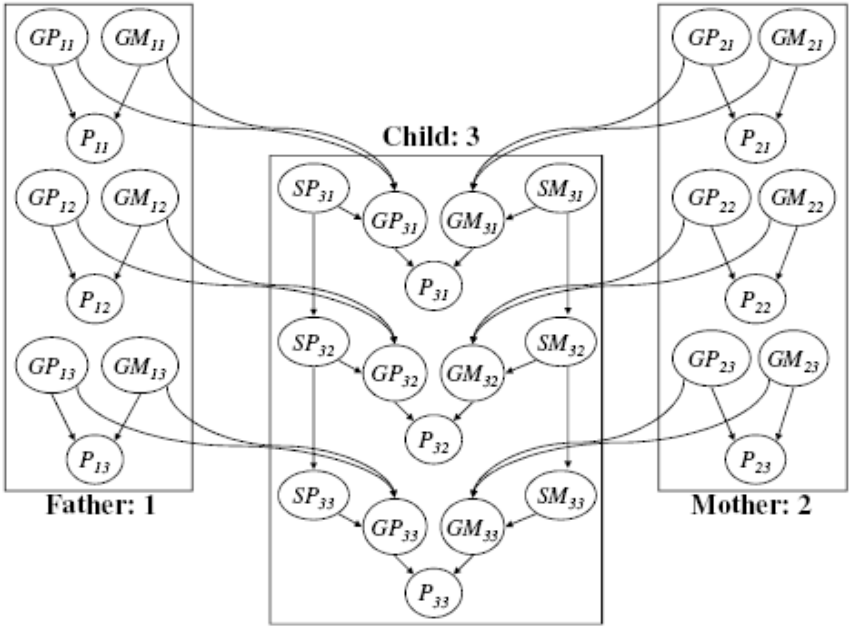
Selectors of second gene SP_{32} and SM_{32} have CPTs

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}

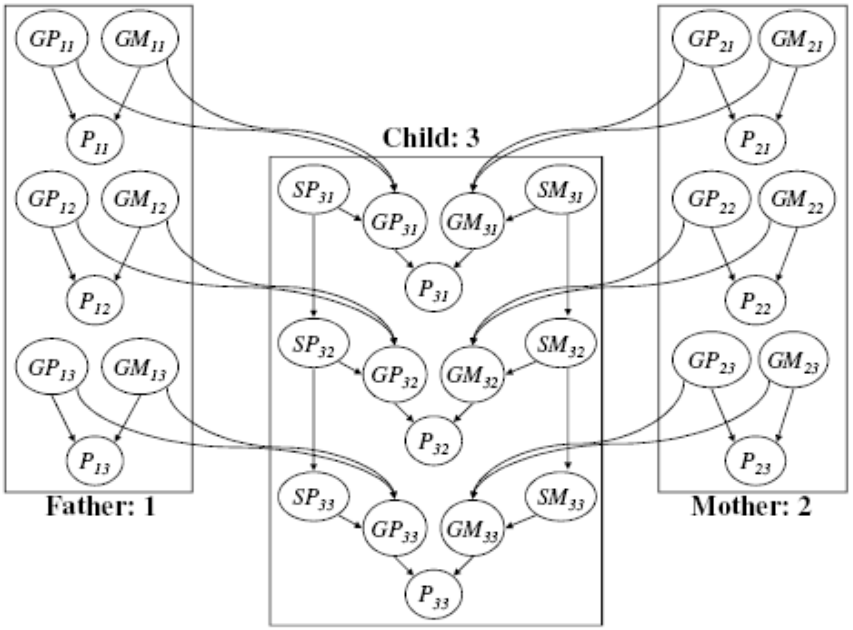
From Pedigrees to Bayesian Networks



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}	$\theta_{sp_{32}\|sp_{31}}$	
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	$1-\theta_{12}$	

Putting the Network to Use

Given network that induces distribution $\Pr(.)$

If $\mathbf{g}$ is evidence about the genotype and $\mathbf{p}$ is evidence about the phenotype, then $\Pr(\mathbf{g}, \mathbf{p})$ represents the likelihood of recombination frequencies included in the network CPTs.

Putting the Network to Use

Given network that induces distribution $\Pr(.)$

If $\mathbf{g}$ is evidence about the genotype and $\mathbf{p}$ is evidence about the phenotype, then $\Pr(\mathbf{g}, \mathbf{p})$ represents the likelihood of recombination frequencies included in the network CPTs.

By changing the CPTs for selector variables (which host the recombination frequencies) and recomputing $\Pr(\mathbf{g}, \mathbf{p})$

we will be able to compute the likelihoods of competing hypotheses about genetic linkage.

Putting the Network to Use

Given network that induces distribution $\Pr(.)$

If $\mathbf{g}$ is evidence about the genotype and $\mathbf{p}$ is evidence about the phenotype, then $\Pr(\mathbf{g}, \mathbf{p})$ represents the likelihood of recombination frequencies included in the network CPTs.

By changing the CPTs for selector variables (which host the recombination frequencies) and recomputing $\Pr(\mathbf{g}, \mathbf{p})$

we will be able to compute the likelihoods of competing hypotheses about genetic linkage.

For a given hypothesis θ_{ij} the score $\log \Pr^{\theta_{ij}}(\mathbf{g}, \mathbf{p}) / \Pr^{.5}(\mathbf{g}, \mathbf{p})$

is typically used to quantify the support for this hypothesis, which is meant to be normalized across different pedigrees.

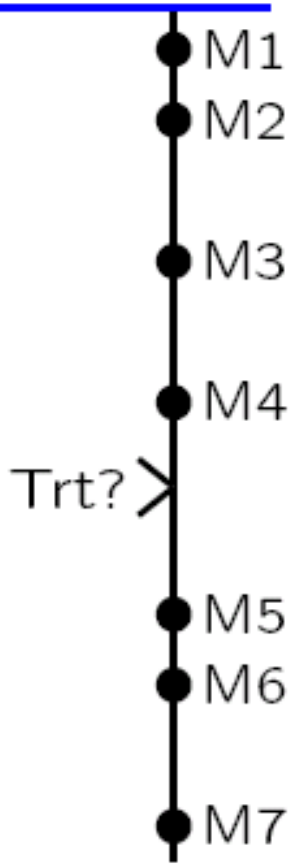
Linkage analysis with pedigree data

GIVEN:

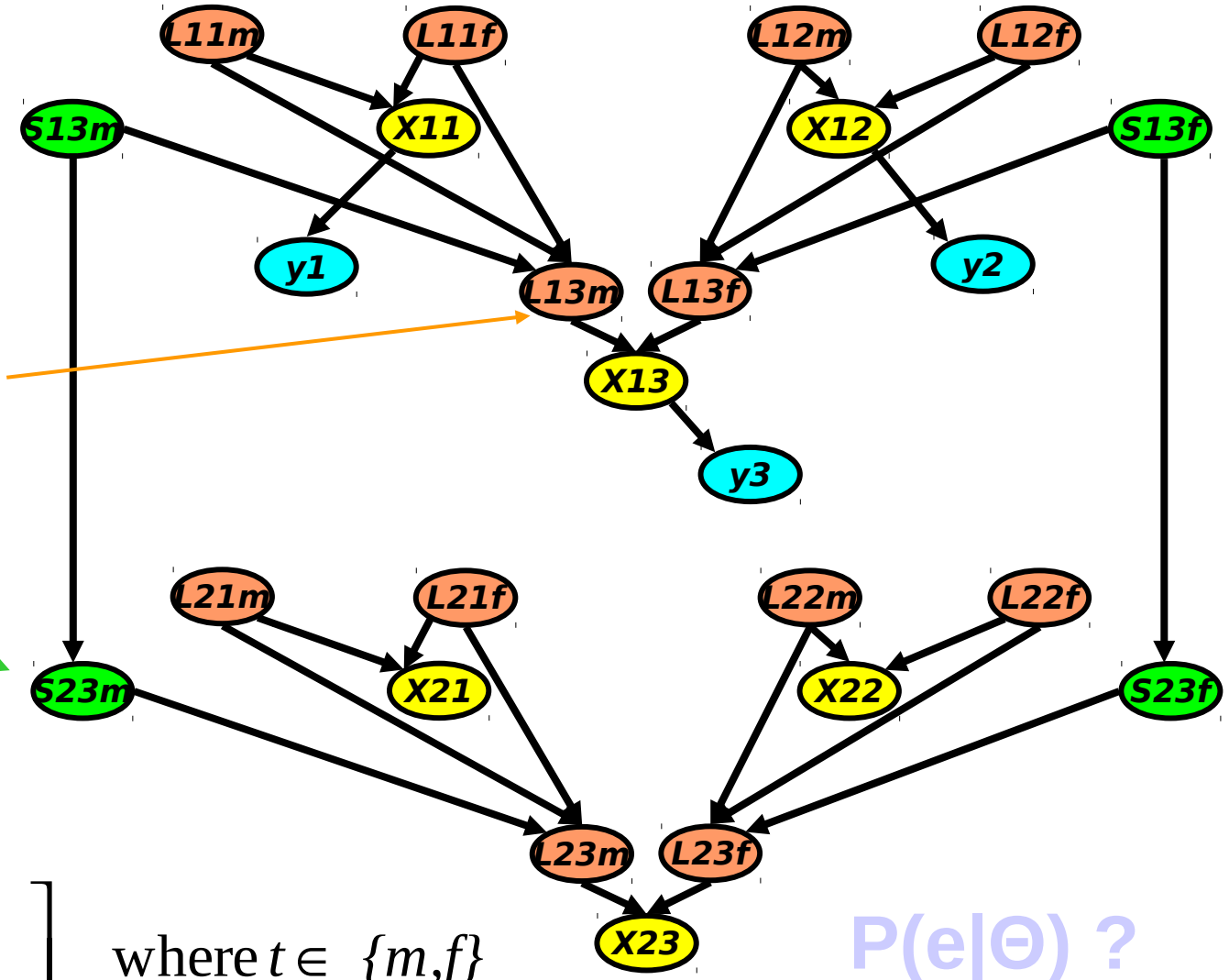
- A set of pedigrees, and some trait of interest.
- A set of DNA markers, with known genetic model (genetic map, and allele frequencies).
- Data on trait(s) and at markers, for some subset of the individuals.

QUESTION: Testing and estimation.

- Does any DNA on the chromosome of the markers affect the trait? H_0 : No.
- If so, what is the likely location of this DNA, relative to markers.



Bayesian Network for Recombination



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

P(e|Θ) ?

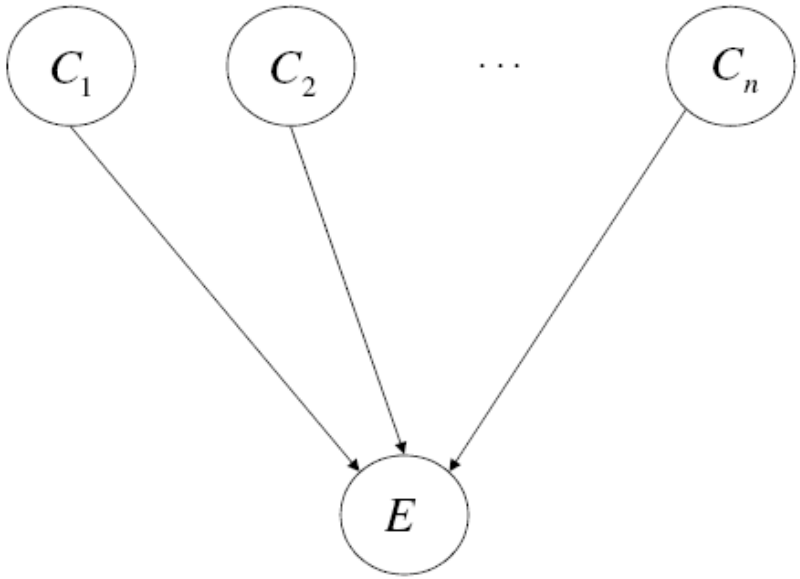
Dealing with Large CPTs

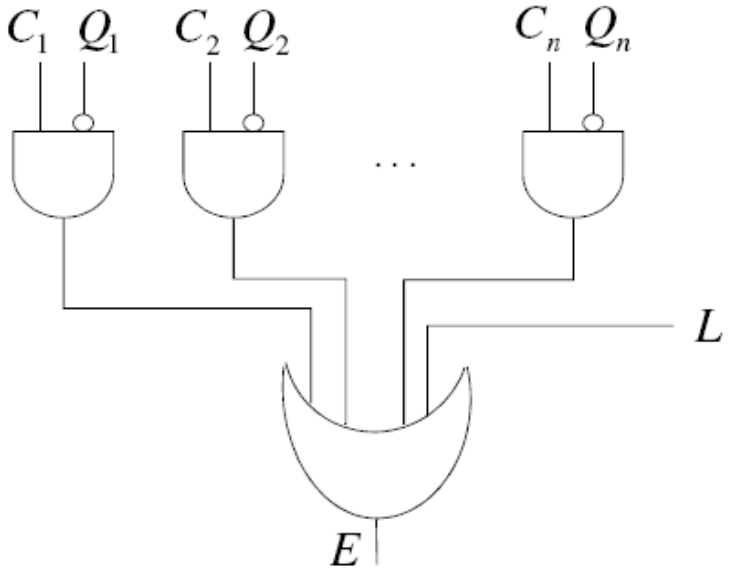
The size of a CPT

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





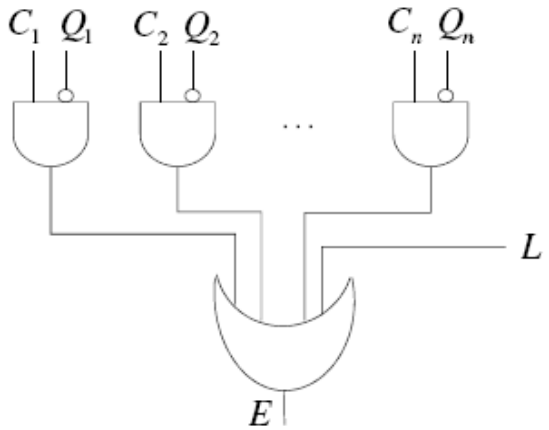
A noisy-or circuit

A micro model

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

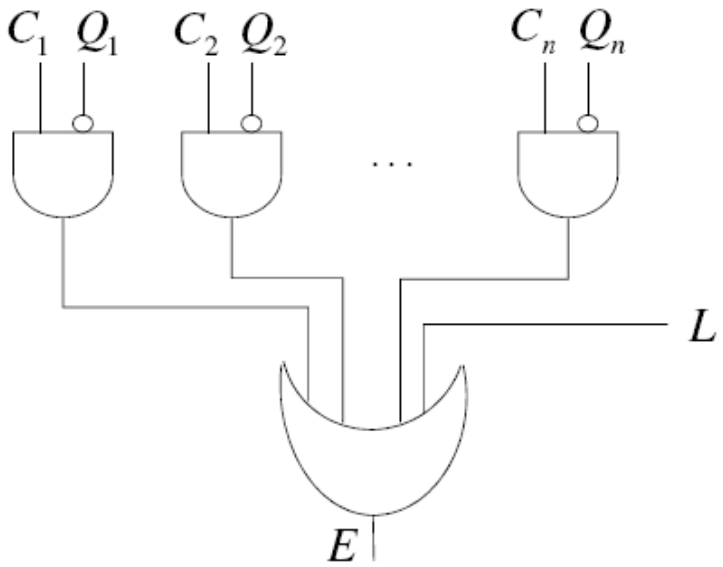
We wish to specify cpt with less parameters

Noisy-or Model



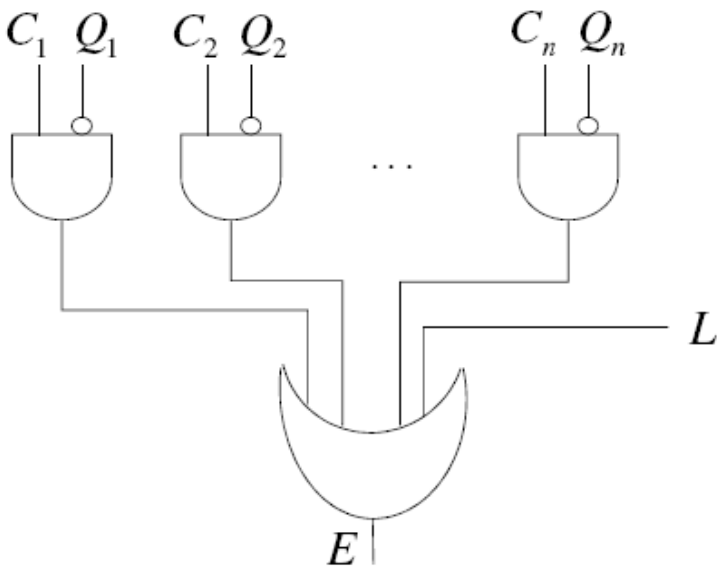
- Cause C_i is capable of establishing effect E, except under some unusual circumstances summarized by suppressor Q_i.
- When suppressor Q_i is active, C_i is no longer able to establish E.
- The leak variable L represents all other causes of E which were not modeled explicitly.
- When none of the causes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

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

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

Noisy-or Model

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

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

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

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

Noisy-or Model

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

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

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

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

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

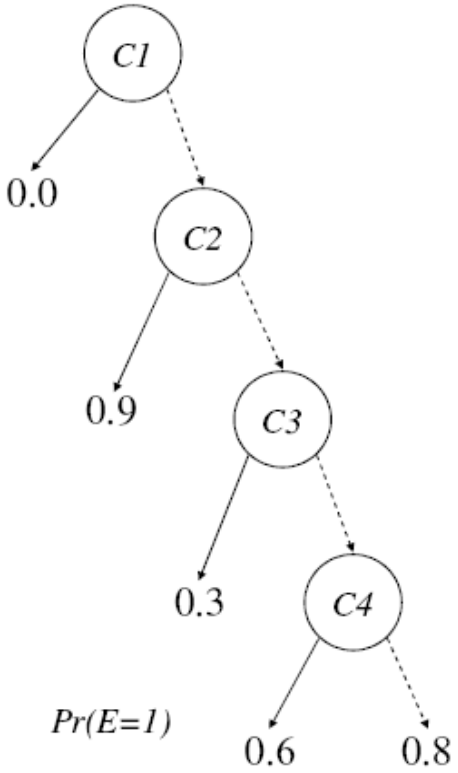
Sore throat (S) has three causes: cold (C), flu (F), tonsillitis (T).

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	$1-(1-.02)$

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.

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.

Deterministic CPTs

A deterministic, or functional CPT

is one in which every probability is either 0 or 1

A deterministic CPT for variable E with values $e_1, \ldots, e_m$

can be represented by a set of propositional sentences of the form:

$$\Gamma_i \iff E = e_i,$$

where we have one rule for each value e_i of E, and the premises Γ_i are mutually exclusive and exhaustive.

The CPT for variable E is then given by

$$\theta_{e_i|\alpha} = \begin{cases} 1, & \text{if parent instantiation } \alpha \text{ is consistent with } \Gamma_i; \\ 0, & \text{otherwise.} \end{cases}$$

Deterministic CPTs

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

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