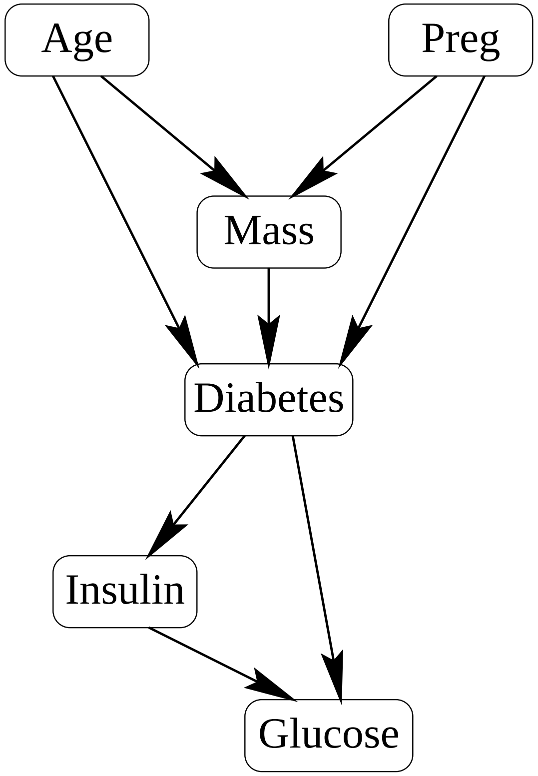
Belief Network Learning

- Known structure, fully observable
- Unknown structure, fully observable
- Known structure, hidden variables
- Unknown structure, hidden variables

Known structure, Fully Observable



Preg	Glucose	Insulin	Mass	Age	Diabetes
5	121	112	26.2	30	0
10	101	180	32.9	63	0
7	137	0	32.0	39	0
12	100	105	30.0	46	0
9	140	0	32.7	45	1
1	102	0	39.5	42	1
2	99	160	36.6	21	0
2	174	120	44.5	24	1
1	111	0	32.8	45	0
5	117	105	39.1	42	0

Learning Process

- **Discretize the Data**

  Glucose $< 100 \Rightarrow 0$

  $100 \leq$ Glucose $< 120 \Rightarrow 1$

  $120 \leq$ Glucose $< 140 \Rightarrow 2$

  $140 \leq$ Glucose $\Rightarrow 3$

- **Count Cases**

  $$P(Mass = 0 | Preg = 1, Age = 2) = \frac{N(Mass = 0, Preg = 1, Age = 2)}{N(Preg = 1, Age = 2)}$$

Unknown structure, Fully Observable

Goal: Find the graph G that maximizes $P(G|D)$.

From Bayes Rule, we know

$$\begin{aligned} \max_G P(G|D) &= \max_G \frac{P(D|G) \cdot P(G)}{P(D)} \\ &= \max_G P(D|G) \cdot P(G) \end{aligned}$$

This is called the MAP (Maximum Aposteriori Probability) hypothesis.

- **Define** $P(G)$
- **Define a search space of graphs** G

  Example: Greedy search in graph space

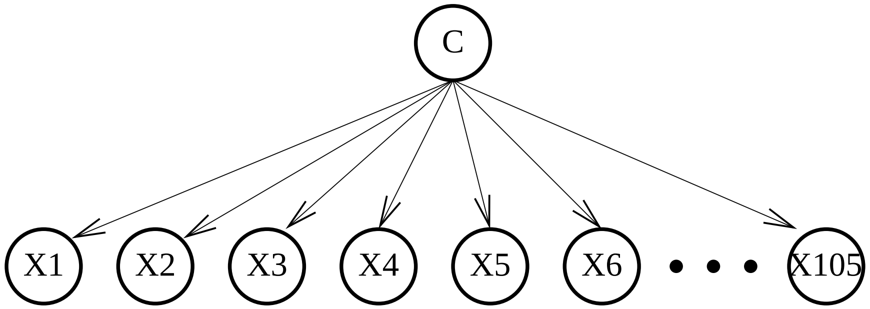
  Start with no edges. Operator adds an edge.

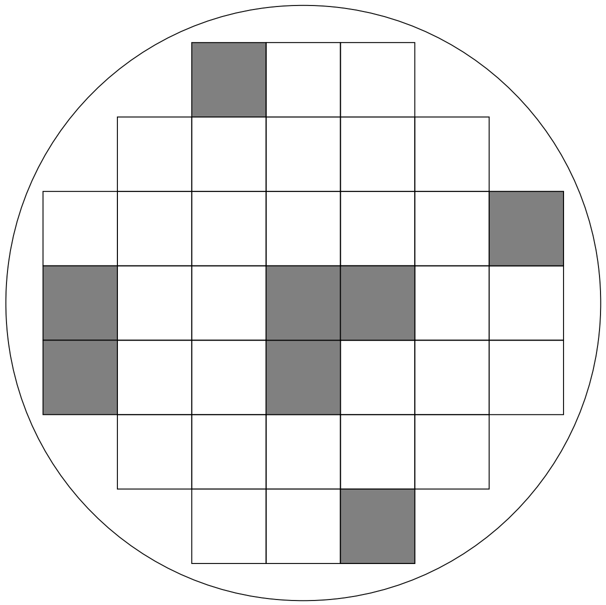
  Add edge that most increases $P(D|G) \cdot P(G)$

  This requires computing all of the conditional probability tables in G using the data D.

Known structure, Hidden Variables

Example: VLSI Wafer Test Results





Learning with Hidden Variables

- **Choose the number of values of each hidden variable**
- **Randomly initialize the conditional probability tables**
- **Gradient Descent Search in** $\frac{\partial \ln P(D|G)}{\partial w_i}$ **for each weight** w_i

$$\frac{\partial \ln P(D|G)}{\partial w_i} = \frac{\partial \ln \Pi_j P(D_j|G)}{\partial w_i} = \sum_j \frac{\partial \ln P(D_j|G)}{\partial w_i} = \sum_j \frac{1}{P(D_j)} \frac{\partial P(D_j)}{\partial w_i}$$

Suppose we are interested in a particular CPT entry $w_i = P(X = x|U = u)$, where U are the parent variables of some node X. By the chain rule:

$$P(D_j) = \sum_{X,U} P(D_j, X, U) = \sum_{X,U} P(D_j|X,U)P(X|U)P(U)$$

Hence,

$$\frac{1}{P(D_j)} \frac{\partial P(D_j)}{\partial w_i} = \frac{1}{P(D_j)} \frac{\partial [\sum_{X,U} P(D_j|X,U)P(X|U)P(U)]}{\partial w_i}$$

Only one term in this sum will involve w_i:

$$P(D_j|X = x, U = u)P(X = x|U = u)P(U = u) = P(D_j|X = x, U = x)w_i P(U = u)$$

The derivative is therefore $P(D_j|X = x, U = u)P(u)$

Gradient Descent Computation (2)

$$\frac{1}{P(D_j)} \frac{\partial P(D_j)}{\partial w_i} = \frac{P(D_j|X = x, U = u)P(U = u)}{P(D_j)}$$

Now apply Bayes' Rule to $P(D_j|X = x, U = u)$, and obtain

$$\begin{aligned}
\frac{1}{P(D_j)} \frac{\partial P(D_j)}{\partial w_i} &= \frac{P(X = x, U = u|D_j)P(D_j)P(U = u)}{P(X = x, U = u)P(D_j)} \\
&= \frac{P(X = x, U = u|D_j)P(U = u)}{P(X = x|U = u)P(U = u)} \\
&= \frac{P(X = x, U = u|D_j)}{P(X = x|U = u)} \\
&= \frac{P(X = x, U = u|D_j)}{w_i}
\end{aligned}$$

The numerator can be computed by standard belief net inference algorithms (e.g., SPI).

Wafer Test Example (2)

Gradient for $w = P(X_i = x|C = c)$:

$$\frac{P(x_i = x, C = c|D_j)}{w} = \frac{P(C = c|D_j)P(X_i = x|C = c, D_j)}{w}$$

The second term in the numerator is either 0 or 1 depending on whether D_j contains $X_i = x$.

The first term in the numerator can be computed as follows:

$$P(C = c|D_j) = \frac{P(D_j|C = c)P(C = c)}{P(D_j)} = \alpha P(C = c) \prod_i P(X_i = x_i|C = c)$$

Gradient for $w = P(C = c)$:

$$\frac{P(C = c|D_j)}{w}$$

This can be computed in exactly the same way.

Wafer Test Example (3)

Graduate Student Tony Fountain learned a chip failure model using a hidden variable C with four values. The learned model can then be used to decide which chips to test in order to maximize the profit on a product (by minimizing the amount of testing needed to decide which chips to "package").

The result was a (predicted) increase in profit of 3.6% for one product manufactured by HP Corvallis.

Unknown Structure, Hidden Variables

No effective algorithms for this case yet.