Lecture 6:
Probabilistic graphical models

Nando de Freitas
www.cs.ubc.ca/~nando/340-2009/
September 2009

Outline

Probabilistic graphical models (also known as Bayesian networks) combine probability theory and graph theory to represent large domains of random variables.

We will tackle two tasks: inference and learning.

In inference, we assume we have the conditional probability tables and focus on estimating the probability of a group of variables given the other variables. We will derive the celebrated HMM filter as part of this.

In learning, we compute the conditional probability tables from data.

Let $\mathbf{x}$ denote two random variables $\mathbf{x} = (x_1, x_2)$, each taking 3 possible values. That is, $x_i \in E = \{1, 2, 3\}$. We can represent the marginal, conditional and joint distributions with the following tables:

GIVEN

	1	2	3
$P(x_1) =$	0.2	0.5	0.3
$P(x_2) =$	1	0	0

examples

x

$P(x_2|x_1)$

Given

$P(x_1) =$

.2	.5	.3

$P(x_2|x_1) =$

	x_2			
x_1	.2	.8	0	$\rightarrow \sum = 1$
	.1	.1	.8	$\rightarrow 1$
	0.3	0.4	.3	$\rightarrow 1$

$$\sum_{x_2} P(x_2|x_1) = 1$$

$$P(x_2, x_1) = P(x_1)\, P(x_2|x_1)$$

x_1:

0	1	2
.2	.5	.3

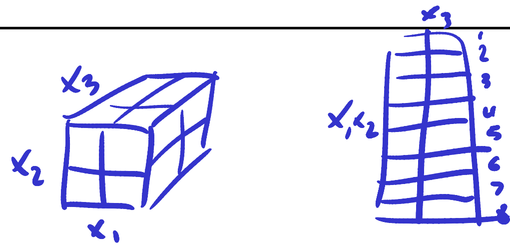
x_2:

x_1	0	1	2
0	.2	.8	0
1	.1	.1	.8
2	.3	.4	.3

$\Rightarrow$

	X_2: 0	1	2	
	.04	.16	0	0
	.05	.05	0.4	1 x_1
	.09	.12	0.09	2

$P(x_2) = ?$



$x_i \in E = \{1, \ldots, r\} \qquad \text{for } i = 1:n$

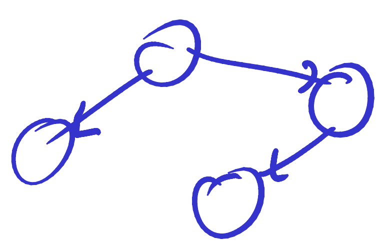
$size(\text{ joint probability table }) = r^n$

Huge!

Directed probabilistic graphs

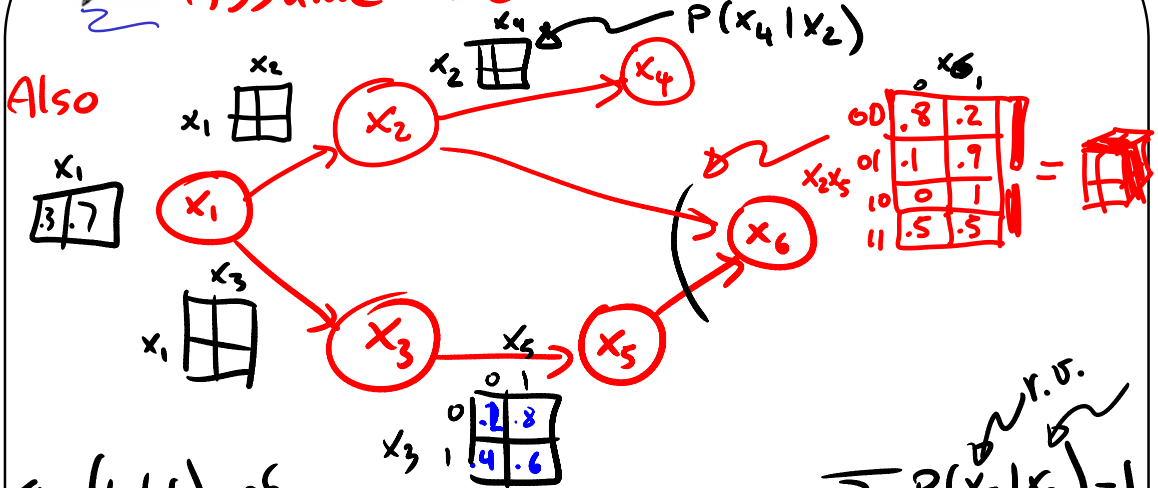
We can exploit conditional independencies and graph theory to replace large tables by a group of smaller tables.

A **directed graph** is a pair $G = (x, e)$ with nodes $x_{1:n}$ and directed edges $e = \{(x_i, x_j) : i \neq j\}$. The nodes will correspond to r.v.s and the edges to conditional probabilities. We assume that G is acyclic.



$P(x_{1:6}) = P(x_1)P(x_2|x_1)P(x_4|x_2)P(x_3|x_1)P(x_5|x_3)P(x_6|x_5,x_2)$
$= P(x_1,x_2,x_3,x_4,x_5,x_6)$

Assume we have 6 binary r.v.s



size(table) $= 2^6$

size(dag) $= 16+8+2$
$= 24+2$
$= 26$

$P(x_5=1|x_3=1) = 0.6$

$P(x_5=1|x_3=0) = .8$

$\sum_{x_5} P(x_5|x_3) = 1$

$\sum_{x_3} P(x_5|x_3) \neq 1$

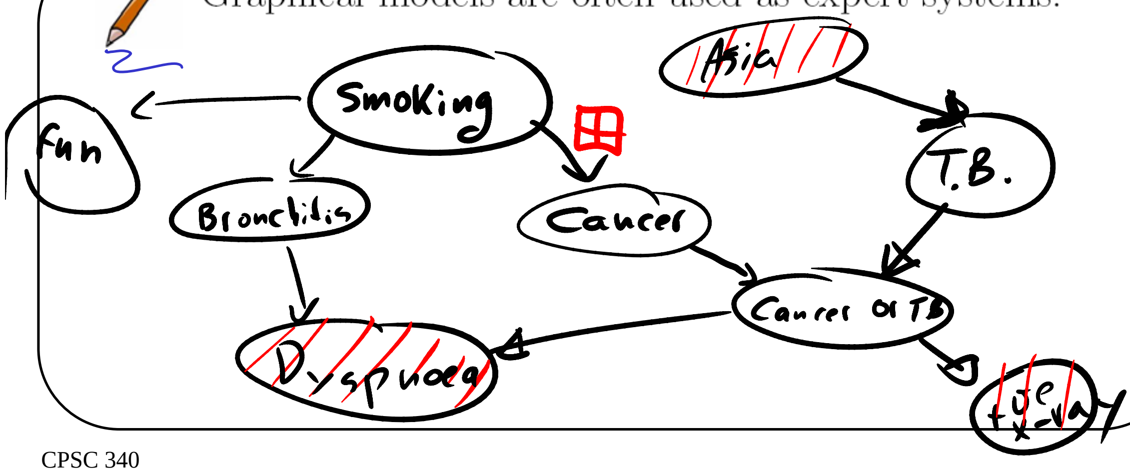
In general: $P(\text{bronchitis}=\text{true} \mid \text{tocxray}=1, \text{Asia}=0, \text{Dysp}=0)$

$$\boxed{p(x_{1:n}) = \prod_{i=1}^{n} p(x_i|parents(x_i))}$$

The size of each table is r^{m_i+1}, where m_i is the number of parents of node x_i.

Given

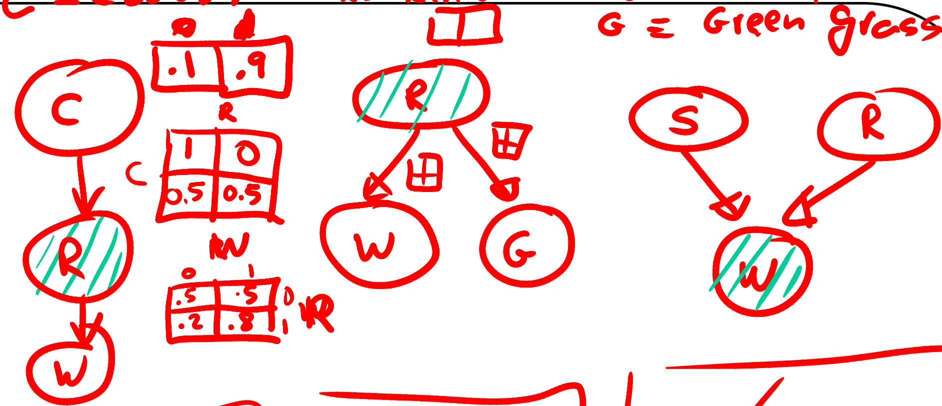
Graphical models are often used as expert systems:



2

Conditional independence

C ≡ CLOUDY R ≡ RAIN S ≡ Shower W ≡ WET G ≡ Green grass

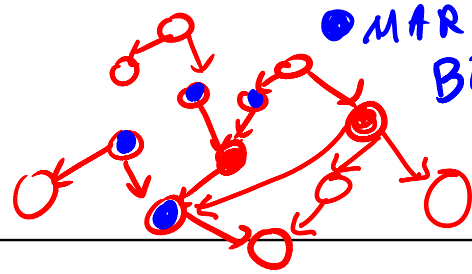


$\boxed{W \perp\!\!\!\perp C \mid R}$ $\boxed{W \perp\!\!\!\perp G \mid R}$ $\boxed{S \not\perp\!\!\!\perp R \mid W}$

independent

black guy only depends on blue guys

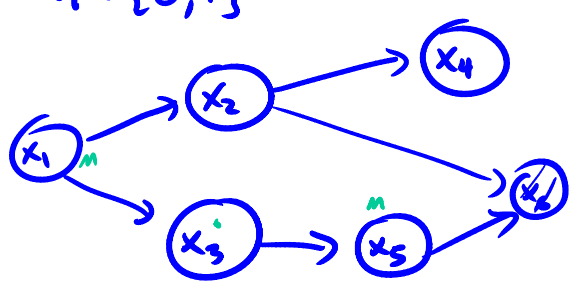
MARKOV BLANKET



ab+ac
a(b+c)

Efficient inference in DAGs

$x_i \in \{0,1\}$



$P(x_1 | x_6=1) = ?$

$$P(x_1 | x_6 = 1) = \frac{P(x_1, x_6=1)}{P(x_6=1)}$$

$$P(x_1 | x_6 = 1) = \frac{\sum_{x_2} \sum_{x_3} \sum_{x_4} \sum_{x_5} P(x_1 x_2 x_3 x_4 x_5 x_6 = 1)}{\sum_{x_{1:5}} P(x_{1:5}, x_6 = 1)}$$

Efficient inference in DAGs

$ab + ac$
$\downarrow$
$a(b+c)$

$$P(x_1, x_6=1) = \sum_{x_{2:5}} P(x_1)\, \underline{P(x_2|x_1)}\, \underline{P(x_3|x_2)}\, P(x_4|x_2)\, P(x_5|x_3)\, P(x_6=1|x_2, x_5)$$

$$= P(x_1) \sum_{x_2} P(x_2|x_1) \left\{ \sum_{x_3} P(x_3|x_2) \left[\sum_{x_4} P(x_4|x_2) \underbrace{\left(\sum_{x_5} P(x_5|x_3)\, P(x_6=1|x_2 x_5) \right)}_{\phi(x_3, x_2)} \right] \right\}$$

$$= P(x_1) \sum_{x_2} P(x_2|x_1) \left\{ \sum_{x_3} P(x_3|x_2)\, \phi(x_3, x_2) \underbrace{\left[\sum_{x_4} P(x_4|x_2) \right]}_{1} \right\}$$

$$= P(x_1) \sum_{x_2} P(x_2|x_1) \left[\sum_{x_3} P(x_3|x_2)\, \phi(x_3, x_2) \right]$$

$$= P(x_1) \sum P(x_2|x_1)\, \psi(x_1, x_2) = P(x_1)\, \Omega(x_1)$$

Efficient inference in DAGs

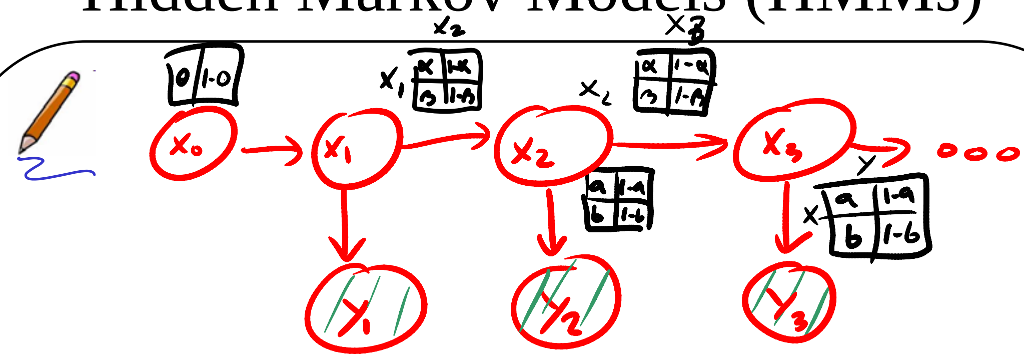
$$P(x_1 | x_6=1) = \frac{P(x_1)\, \Omega(x_1)}{\sum_{x_1} P(x_1)\, \Omega(x_1)}$$

Junction tree algorithm

The idea of replacing sums of products ($ac+ab$) by products of sums ($a(b+c)$) is at the heart of most inference algorithms. For exact inference, in Gaussian and discrete networks of reasonable size, we use the **junction tree algorithm**. This algorithm involves two steps:

1. Converting the directed graph to an undirected graph called the junction tree.
2. Running belief propagation. That is, replace sums of products by products of sums.

Hidden Markov Models (HMMs)



What is $P(x_2 \mid y_1, y_2)$?

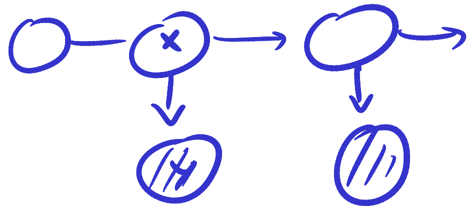
Given: $P(x_t \mid x_{t-1}) = \begin{array}{c|cc} & 0 & 1 \\ \hline 0 & \alpha & 1-\alpha \\ 1 & \beta & 1-\beta \end{array}$

dynamic model

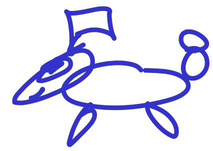
$P(y_t \mid x_t) = \begin{array}{|c|c|} \hline a & 1-a \\ \hline b & 1-b \\ \hline \end{array}$

Observation model

Hidden Markov Models (HMMs)



ABC



$x \in \{\text{sad}, \text{happy}\}$

$y \in \{\text{watch TV}, \text{sleeping}, \text{crying}, \text{socializing}\}$

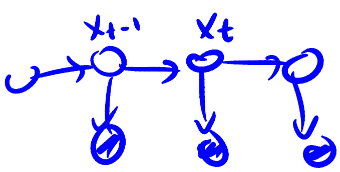
$P(x_t \mid x_{t-1}) =$

	s	h
s	0.8	0.2
h	0.09	0.91

	w	s	c	soc
S	0.3	0	0.1	0.6
H	0.3	0.7	0	0

$P(y_t \mid x_t)$

Hidden Markov Models (HMMs)



$x_t \perp\!\!\!\perp y_{t-1} \mid x_{t-1}$

$P(x_t \mid y_{1:t})$ is what we want

① $P(x_t \mid y_{1:t-1}) = \sum_{x_{t-1}} P(x_t, x_{t-1} \mid y_{1:t-1})$

$= \sum_{x_{t-1}} P(x_t \mid x_{t-1}) P(x_{t-1} \mid y_{1:t-1})$

Hidden Markov Models (HMMs)

$$P(X_t \mid y_{1:t-1}) = \sum_{X_{t-1}} P(X_t \mid X_{t-1})\, P(X_{t-1} \mid y_{1:t-1})$$

0 1

$$P(A \mid BC) = \frac{P(B \mid AC)\, P(A \mid C)}{P(B \mid C)}$$

$$P(X_t \mid y_{1:t}) = P(\overset{A}{X_t} \mid \overset{B}{y_t}\, \overset{C}{y_{1:t-1}})$$

$$= \frac{P(\overset{B}{y_t} \mid \overset{A}{X_t}, \overset{C}{y_{1:t-1}})\, P(\overset{A}{X_t} \mid \overset{C}{y_{1:t-1}})}{P(\overset{B}{y_t} \mid \overset{C}{y_{1:t-1}})}$$

$$= \frac{P(y_t \mid X_t)\, P(X_t \mid y_{1:t-1})}{P(y_t \mid y_{1:t-1})}$$

MRFs

In undirected graphs, the nodes still represent the random variables, but the edges represent compatibility functions.

Continuous graphical models

Parameter learning in DAGs

Let the DAG be

And assume we have collected the data:

c	r	**g**
0	0	0
0	0	0
1	0	1
1	1	1
1	1	1

Parameter learning in DAGs

The conditional probabilities are:

$$
\begin{aligned}
p(c|\gamma) &\propto \\
p(r|\alpha_1, c=0) &\propto \\
p(r|\alpha_2, c=1) &\propto \\
p(g|\beta_1, c=0) &\propto \\
p(g|\beta_2, c=1) &\propto
\end{aligned}
$$

Parameter learning in DAGs

and hence, the ML estimates are:

$$
\begin{aligned}
\gamma &= \\
\alpha_1 &= \\
\alpha_2 &= \\
\beta_1 &= \\
\beta_2 &=
\end{aligned}
$$

Now we can carry out inference to answer queries like $p(g|r=1)$.

$$p(g=0|r=1) =$$

Likelihood-based model selection

How about using another model to represent the same data?

$$\begin{aligned} p(c|\gamma) &\propto \\ p(r|\alpha_1, c=0) &\propto \\ p(r|\alpha_2, c=1) &\propto \\ p(g) &\propto \end{aligned}$$

$$\begin{aligned} \gamma &= \\ \alpha_1 &= \\ \alpha_2 &= \\ \beta &= \end{aligned}$$

Cross-validation for model selection

⋆ Let the test data point be $x_{test} = (1, 1, 1)$ and the two DAGs be denoted M_1 and M_2. Then

$$p(x_{test}|\theta_1, M_1) =$$

$$p(x_{test}|\theta_2, M_2) =$$

Bayesian model choice

The current approach has a few short-comings:

- There is no mechanism for incorporating *a priori* knowledge.
- The model selection strategy is very dependent on the parameter estimates. If we have few data points, the parameter estimates can be misleading.
- Model selection requires extra data (the test dataset).

The Bayesian learning paradigm helps surmount these difficulties.