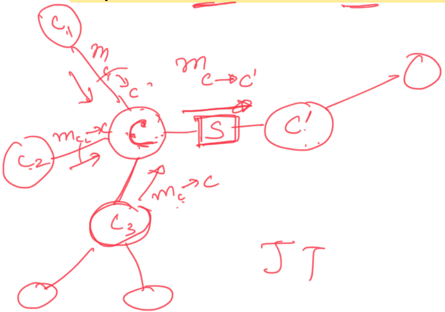
Message passing on junction trees

- Each node c (clique in the JT)
  - sends *message* $m_{c\to c'}(.)$ to each of its neighbors c' once it has messages from every other neighbor $N(c)-\{c'\}$.

$$m_{c\to c'}(\mathbf{x}_s) = \sum_{\mathbf{x}_{c-s}} \psi_c(\mathbf{x}_c) \prod_{d\in N(c)-\{c'\}} m_{d\to c}(\mathbf{x}_{d\cap c})$$

pre-multiplication of all potentials in node c.

Replace "sum" with "max" for MAP queries.



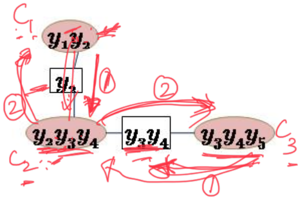
Computing marginal probability of any variable x_i

Marginal probability over all variables in a clique

$$\Pr(\mathbf{x}_c) \propto \psi_c(\mathbf{x}_c) \prod_{d \in N(c)} m_{d \to c}(\mathbf{x}_{d \cap c})$$

1. c = clique in JT containing x_i
2. $\Pr(x_i) = \sum_{\mathbf{x}_{c-x_i}} \Pr(\mathbf{x}_c)$

Example



$$C_2 \quad \psi_{234}(\mathbf{y}_{234}) = \psi_{23}(\mathbf{y}_{23})\psi_{34}(\mathbf{y}_{34})$$

$$C_3 \quad \psi_{345}(\mathbf{y}_{345}) = \psi_{35}(\mathbf{y}_{35})\psi_{45}(\mathbf{y}_{45})$$

$$C_1 \quad \psi_{\cancel{234}\ 12}(\mathbf{y}_{12}) = \psi_{12}(\mathbf{y}_{12})$$

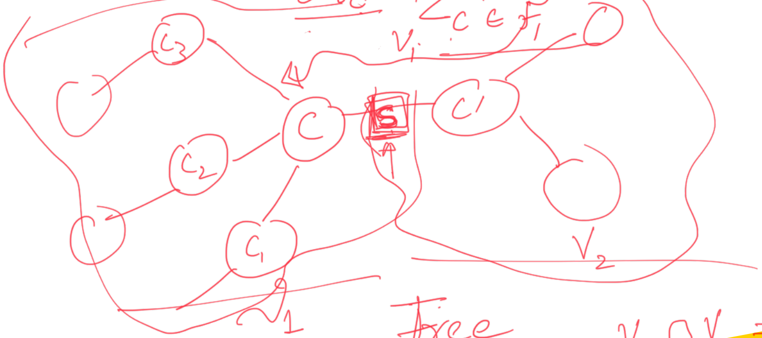
1. Clique "12" sends Message $m_{12\to234}(y_2) = \sum_{y_1} \psi_{12}(\mathbf{y}_{12})$ to its only neighbor.
2. Clique "345" sends Message $m_{345\to234}(\mathbf{y}_{34}) = \sum_{y_5} \psi_{\cancel{234}\ 345}(\mathbf{y}_{345})$ to "234"
3. Clique "234" sends Message $m_{234\to345}(\mathbf{y}_{34}) = \sum_{y_2} \psi_{234}(\mathbf{y}_{234}) m_{12\to234}(y_2)$ to "345"
4. Clique "234" sends Message $m_{234\to12}(y_2) = \sum_{y_4\ y_3} \psi_{234}(\mathbf{y}_{234}) m_{345\to234}(\mathbf{y}_{34})$ to "12"

$$\Pr(y_1) \propto \sum_{y_2} \boxed{\psi_{12}(\mathbf{y}_{12}) m_{234\to12}(y_2)} \quad P(y_1, y_2)$$

Intuition behind message passing algorithm

Message from c to c' denotes the result of VE elimination of potentials on the side of the tree that contains clique c but not c', leaving only the separator variables $s = c \cap c'$

$m_{c \to c'}(s) = \sum \prod_{c \in f_1} \psi_c(x_c)$



Free

$V = s \cup V_1 \cup V_2$

$V_1 \cap V_2 = \phi$

Adding evidence

Given fixed values of a subset of variables $\mathbf{x}_e$ (evidence), find the

1. *Marginal probability queries over a small subset of variables:*
   - Find Pr(Income='High | Degree='PhD')

$$P(x_1|x_e)\quad \cancel{\Pr(x_1)} = \sum_{\underbrace{x_2 \ldots x_m}_{= V - x_e - x_1}} \Pr(x_1 \ldots x_n | \mathbf{x}_e)$$

2. *Most likely labels of remaining variables:* (MAP queries)
   - Find likely temperature at sensors in a room given readings from a subset of them

$$\mathbf{x}^* \overset{x_e}{=} \text{argmax}_{x_1 \ldots x_m} \Pr(x_1 \ldots x_n | \mathbf{x}_e)$$

Easy to add evidence, just change the potential.

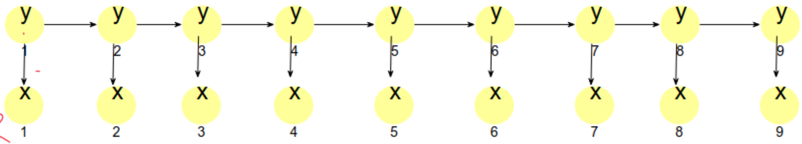
Case study: HMMs for Information Extraction

My review of Fermat's last theorem by S. Singh

$y_i \in \{\text{Author}, \text{Title}, \text{Others}\}$

$m = 3$

t	1	2	3	4	5	6	7	8	9
x	My	review	of	Fermat's	last	theorem	by	S.	Singh
y	**Other**	**Other**	**Other**	**Title**	**Title**	**Title**	**other**	**Author**	**Author**

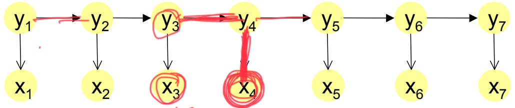


x

$x_i \in$

Inference in HMMs

- Given,
  - Graph



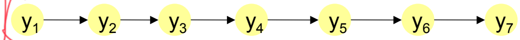
  - Potentials: $\Pr(y_i|y_{i-1}), \Pr(x_i|y_i)$
  - Evidence variables: $\mathbf{x} = x_1 \ldots x_n = o_1 \ldots o_n$.
- Find most likely values of the hidden state variables.

  $\mathbf{y} = y_1 \ldots y_n$

$$\text{argmax}_{\mathbf{y}} \Pr(\mathbf{y}|\vec{\mathbf{x}} = \vec{\mathbf{o}})$$

Re

- Define $\psi_i(y_{i-1}, y_i) = \Pr(y_i|y_{i-1}) \boxed{\Pr(x_i = o_i|y_i)}$
- Reduced graph only a single chain of y nodes.



- Algorithm same as earlier, just replace "Sum" with "Max"

This is the well-known Viterbi algorithm

Numerical Example

$m=2$

$P(y|y') =$

y'	$P(y=0\|y')$	$P(y=1\|y')$
0	0.9	0.1
1	0.2	0.8

$P(y_2|y_1) = P(y_3|y_2) \ldots$

$P(y_i|y_{i-1}) \quad P(y_1)$

$P(x|y) =$

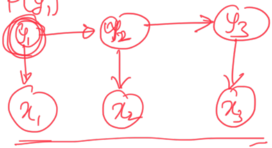
y	$P(x=0\|y)$	$P(x=1\|y)$
0	0.7	0.3
1	0.6	0.4

$P(x_i|y_i) = P(x_1|y_1) = P(x_2|y_2)$

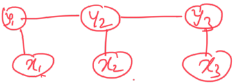
$P(y_1)$

$P(y=1) = 0.5$

Observation $[x_0, x_1, x_2] = [0, 0, 0]$



$\arg\max P(y_1, y_2, y_3 \mid x_1, x_2, x_3 = [0,0,0])$



$$\psi_{12}(y_1, y_2) = \begin{array}{c|cc} & 0 & 1 \\ \hline 0 & 0.9 \times 0.7 \times 0.5 & 0.1 \times 0.7 \times 0.5 \\ 1 & 0.2 \times 0.6 \times 0.5 & 0.8 \times 0.6 \times 0.5 \end{array}$$

y' parent

$[y_1 — y_2 — y_3]$

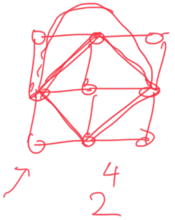
$P(y_2|y_1)\,P(x_1=0|y_1)\,P(y_1)$

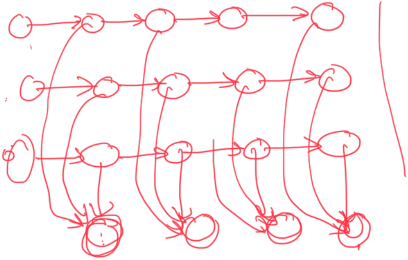
$\psi_{23}(y_2, y_3) = P(y_3|y_2)\,P(x_2=0|y_2)\,P(x_3=0|y_3)$

Doubt -

Why approximate inference

- Exact inference is NP hard. Complexity: $O(m^w)$ doubt- confirm`
  - $w =$ tree width $=$ size of the largest clique in (triangulated) graph-1,
  - $m =$ number of values of each discrete variable in the clique.
- Many real-life graphs produce large cliques on triangulation
  - A $n \times n$ grid has a tree width of n
  - A Kalman filter on K parallel state variables influencing a common observation variable, has a tree width of size $K + 1$

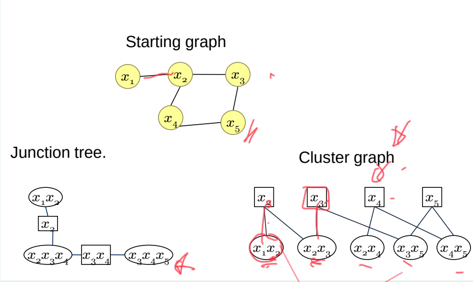




Generalized belief propagation

- Approximate junction tree with a cluster graph where
  1. Nodes = arbitrary clusters, not cliques in triangulated graph. Only ensure all potentials subsumed.
  2. Separator nodes on edges = *subset* of intersecting variables so as to satisfy running intersection property.
- Special case: Factor graphs.

Example cluster graph



Belief propagation in cluster graphs

- Graph can have loops, tree-based two-phase method not applicable.
- Many variants on scheduling order of propagating beliefs.
  - Simple loopy belief propagation [**?**]
  - Tree-reweighted message passing [**?**, **?**]
  - Residual belief probagation [**?**]
- Many have no guarantees of convergence. Specific tree-based orders do [**?**]
- Works well in practice, default method of choice.

Others

- Sampling (to be discussed later)
- Combinatorial algorithms for MAP [?].
- Greedy algorithms: relaxation labeling.
- Variational methods like mean-field and structured mean-field.
- LP and QP based approaches.