Lecture 13: Graphical Models and Bayesian Inference

Very popular in social sciences, physical sciences and engineering.

Graphical structures . . . A historical perspective from a communication perspective

$$\begin{tikzcd}
\underline{u} \arrow[rr] \arrow[dr] & & s \arrow[dl] & \\
& \underline{v} \arrow[rr] & & \underline{y}
\end{tikzcd}$$

$$P(\underline{u}, s, \underline{v}, \underline{y}) = P(\underline{u})P(s \mid \underline{u})P(\underline{v} \mid s, \underline{u})P(\underline{y} \mid \underline{v})$$

$$p\left(\underline{u}, s, \underline{v}, \underline{y}\right) = p\left(\underline{u}\right) p\left(s \mid \underline{u}\right) p\left(\underline{v} \mid s, \underline{u}\right) p\left(\underline{y} \mid \underline{v}\right)$$

$\underline{u} \sim$ source code; $s =$ state of encoder

$\underline{v} \sim$ encoded signal; $\underline{y} \sim$ Received signal

$$p(\underline{u} \mid \underline{y}) = \frac{p(\underline{u}, \underline{y})}{p(\underline{y})} = \frac{p(\underline{u}, \underline{y})}{\sum_{\underline{u}} p(\underline{u}, \underline{y})}$$

$$p(\underline{u}, \underline{y}) = \sum_{s} \sum_{\underline{v}} p\left(\underline{u}\right) p\left(s \mid \underline{u}\right) p\left(\underline{v} \mid s, \underline{u}\right) p\left(\underline{y} \mid \underline{v}\right)$$

$$= p\left(\underline{u}\right) \underbrace{\sum_{s} p\left(s \mid \underline{u}\right) \underbrace{\sum_{\underline{v}} p\left(\underline{v} \mid s, \underline{u}\right) p\left(\underline{y} \mid \underline{v}\right)}_{\text{do this first..} f(\underline{u}, s, \underline{y})}}_{g(s, \underline{u})} \text{....exploit structure}$$

Sewall Wright (1921) . . . Developed "path analysis" as a means to study statistical relationships in biological data .. Genetics was one of the hottest topics when he entered Harvard in 1912. He worked for USDA on guinea pigs. His theory was that evolution is not gradual, but takes place in sudden bursts.

Why some guinea pigs white and others are colored? What developmental factors in the womb, environmental factors after birth, genetic factors from each individual parent and combined hereditary factors from both parents are responsible for white fur color? He showed that developmental factors were more important than heredity (58% to 42% in randomly bred guinea pigs). In highly inbred family, 3% of the variation is due to genetics and 92% due to developmental factors. These also go under the name of causal diagrams…. Good history in "The Book of Why" by Judea Pearl.

1960's . . . Statisticians use graphs to describe restrictions in log-linear statistical models

Gallagher (1963) . . . Error correcting codes as probabilistic graphs

Viterbi algorithm (Forney, 1973)

Taxonomic hierarchies (Woods, 1975)

Medical diagnosis (Spiegelharter, 1990)

Exact algorithms for computing the joint probability distribution (Lauritzen and Spiegelharter, 1988; Pearl, 1986)

Learning parameters in graph-based log-linear models (Hinton and Sejnowski, 1986)

Bayesian networks (belief networks, causal networks or inference diagrams)

Approximate algorithm based on Monte Carlo methods

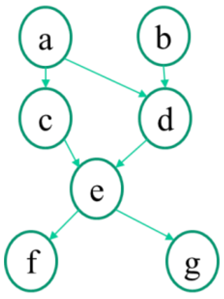
Variational techniques

Also, useful in graph neural networks.

Applications:

- Automatic speech recognition & speaker verification
- Printed and handwritten text parsing
- Face location and identification
- Tracking/separating objects in video
- Search and recommendation (e.g., Google, Amazon)
- Financial prediction, fraud detection (e.g., credit cards)
- Insurance premium prediction, product pricing
- Medical diagnosis/image analysis (e.g., Pneumonia, Pap Smears)
- Game playing (e.g., backgammon)
- Scientific analysis/data visualization (e.g., Galaxy classification)
- Analysis/control of complex systems (e.g. Freeway traffic, industrial
- manufacturing plants, Space Shuttle)
- Troubleshooting and fault correction

Bayesian networks



- Cliques: (abd), (acd), (cde), (eg)
- $P(a,b,c,d,e,f,g) = P(a)P(b)P(c\,|\,a)P(d\,|\,a,b)P(e\,|\,c,d)P(f\,|\,e)P(g\,|\,e)$

$$= \frac{P(abd)P(acd)P(cde)P(ef)P(eg)}{P(ad)P(cd)P(e)P(e)}$$

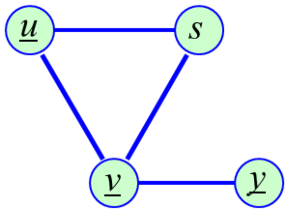
$$= \frac{P(a)P(b)P(d\,|\,a,b)P(c\,|\,a)P(ad)P(e\,|\,cd)P(cd)P(f\,|\,e)P(g\,|\,e)P(e)P(e)}{P(ad)P(cd)P(e)P(e)}$$

$$= P(a)P(b)P(c\,|\,a)P(d\,|\,a,b)P(e\,|\,c,d)P(f\,|\,e)P(g\,|\,e)$$

- Compute the product $P(a,b,d) = P(a)P(b)P(d\,|\,a,b)$
- Sum over b to get $P(a,d) = \sum_b P(a,b,d)$
- Multiply $P(a,d)$ by $P(c\,|\,a)$ to get $P(a,c,d)= P(c\,|\,a)P(a,d)$
- Sum $P(a,c,d)$ over a to get $P(c,d) = \sum_a P(a,c,d)$
- Multiply $P(c,d)$ by $P(e\,|\,c,d)$ to obtain $P(c,d,e) = P(e\,|\,c,d)P(c,d)$
- Sum over c and d to get $P(e)=\sum_c \sum_d P(c,d,e)$
- Multiply $P(e)$ by $P(g\,|\,e)$ to get $P(e,g)$
- Sum $P(e,g)$ over e to get $P(g) = \sum_e P(e,g)$

Variable elimination method. Related to non-serial dynamic programming.
Junction tree algorithm works with cliques and is optimal.
Belief propagation works directly with graph. Optimal for trees and suboptimal for DAGs.

Markov random fields or undirected graphs



"Given its neighbors, each variable is independent of all other variables"

Joint Distribution is given by Hammersley-Clifford Theorem (1971)

$$C_1 = \{\underline{u}, s, \underline{v}\} \qquad C_2 = \{\underline{v}, \underline{y}\}$$

$$P(\mathbf{z}) = \alpha \prod_{j=1}^{NC} \psi_j(C_j) = \alpha \exp\{-\sum_{j=1}^{NC} E_j(C_j)\}$$

$$P(\underline{u}, s, \underline{v}, \underline{y}) = \alpha\, \psi_1(\underline{u}, s, \underline{v})\, \psi_2(\underline{v}, \underline{y})$$

$$\alpha = \frac{1}{P(\underline{v})}$$

Joint distribution = Product of the clique potentials/Product of separator potentials

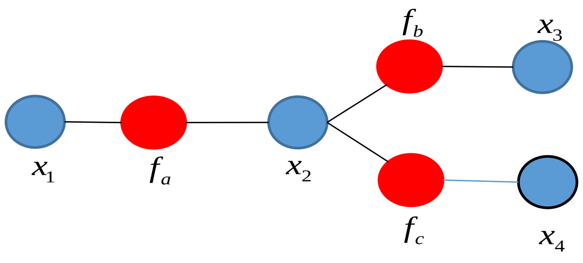
Factor graphs: Show bi-partite graph of the comm problem.

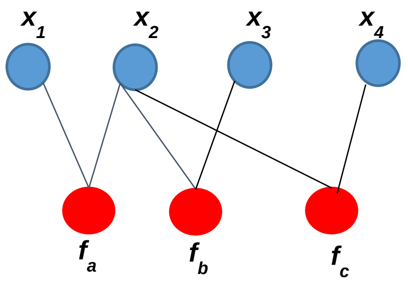
$$p(\underline{u}, s, \underline{v}, \underline{y}) = \underset{factor}{p(\underline{u})}\, \underbrace{p(s\,|\,\underline{u})}_{factor}\, \underbrace{p(\underline{v}\,|\,s, \underline{u})}_{factor}\, \underbrace{p(\underline{y}\,|\,\underline{v})}_{factor}$$

Ideal for belief propagation: Propagate beliefs between variables and factors and factors and variables.

Can still be complex if a factor has lots of variables!

Efficient inference via message passing:





Corresponding Bipartite Graph

$$P(x_1, x_2, x_3, x_4) = f_a(x_1, x_2)\, f_b(x_2, x_3)\, f_c(x_2, x_4)$$

Suppose want marginal distribution $P(x_2)$. By sum formula or Total Probability Theorem

$$P(x_2) = \sum_{x_1}\sum_{x_3}\sum_{x_4} f_a(x_1, x_2)\, f_b(x_2, x_3)\, f_c(x_2, x_4)$$

$$= \underbrace{\left(\sum_{x_1} f_a(x_1, x_2)\right)}_{\mu_{fa\to x_2}(x_2)} \underbrace{\left(\sum_{x_3} f_b(x_2, x_3)\right)}_{\mu_{fb\to x_2}(x_2)} \underbrace{\left(\sum_{x_4} f_c(x_2, x_4)\right)}_{\mu_{fc\to x_2}(x_2)} = \mu_{f_a\to x_2}(x_2)\mu_{f_b\to x_2}(x_2)\mu_{f_c\to x_2}(x_2)$$

Note: $2K^3$ multiplications and K^3 additions versus 2 multiplications and $3K$ additions for each x_2! In fact, can compute any marginal distribution by exchanging messages from variables to factors and factors to variables.

Leaves $\rightarrow$ Root

$$\mu_{x_3 \to f_b}(x_3) = 1 \forall x_3$$
$$\mu_{f_b \to x_2}(x_2) = \sum_{x_3} f_b(x_2, x_3)\mu_{x_3 \to f_b}(x_3)$$
$$\mu_{x_4 \to f_c}(x_4) = 1 \forall x_4$$
$$\mu_{f_c \to x_2}(x_2) = \sum_{x_4} f_c(x_2, x_4)\mu_{x_4 \to f_c}(x_4)$$
$$\mu_{x_2 \to f_a}(x_2) = \mu_{f_b \to x_2}(x_2)\mu_{f_c \to x_2}(x_2)$$
$$\mu_{f_a \to x_1}(x_1) = \sum_{x_2} f_a(x_1, x_2)\mu_{x_2 \to f_a}(x_2)$$

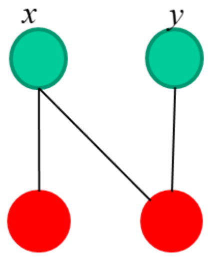
Root $\rightarrow$ Leaves

$$\mu_{x_1 \to f_a}(x_1) = 1 \forall x_1$$
$$\mu_{f_a \to x_2}(x_2) = \sum_{x_1} f_a(x_1, x_2)\mu_{x_1 \to f_a}(x_1)$$
$$\mu_{x_2 \to f_b}(x_2) = \mu_{f_a \to x_2}(x_2)\mu_{f_c \to x_2}(x_2)$$
$$\mu_{x_2 \to f_c}(x_2) = \mu_{f_a \to x_2}(x_2)\mu_{f_b \to x_2}(x_2)$$
$$\mu_{f_b \to x_3}(x_3) = \sum_{x_2} f_b(x_2, x_3)\mu_{x_2 \to f_b}(x_2)$$
$$\mu_{f_c \to x_4}(x_4) = \sum_{x_2} f_c(x_2, x_4)\mu_{x_2 \to f_c}(x_2)$$

Bayes Rule as message passing:

As the clock strikes midnight in the Old Tudor Mansion, a raging storm rattles the shutters and fills the house with the sound of thunder. The dead body of Mr Black lies slumped on the floor of the library, blood still oozing from the fatal wound. Quick to arrive on the scene is the famous sleuth Dr Bayes, who observes that there were only two other people in the Mansion at the time of the murder. So who committed this dastardly crime? was it the mysterious and alluring femme fatale Miss Auburn? Or Was it the fine upstanding pillar of the establishment Major Grey?

Potential weapons: revolver, fire rod, dagger. Searching carefully around the library, Dr Bayes spots a bullet lodged in the book case. “Hmm, interesting”, he says, “I think this could be an important clue”.



$$P(x) = \begin{bmatrix} 0.8 \\ 0.2 \end{bmatrix}; x \in \{1, 2\}$$

$$P(y \mid x) = \begin{bmatrix} 0.05 & 0.65 & 0.30 \\ 0.8 & 0.10 & 0.10 \end{bmatrix}; y \in \{1, 2, 3\}$$

Want $P(y)$ and $P(x \mid y = 1)$

$f_a = P(x)$ $f_b = P(y \mid x)$

Marginal $P(y)$

$\mu_{f_a \to x}(x) = P(x) \Rightarrow \mu_{f_a \to x}(x=1) = 0.8; \mu_{f_a \to x}(x=2) = 0.2$

$\mu_{x \to f_b}(x) = \mu_{f_a \to x}(x)$

$\mu_{f_b \to y}(y) = \sum_x f_b(x,y)\mu_{x \to f_b}(x) \Rightarrow \mu_{f_b \to y}(y=1) = 0.2; \mu_{f_b \to y}(y=2) = 0.54; \mu_{f_b \to y}(y=3) = 0.26$

$P(y) = \mu_{f_b \to y}(y)$

Evidence, $y=1$. What is $P(x \mid y=1)$?

Treat x as the root.

$\mu_{y \to f_b}(y=1) = 1; \mu_{y \to f_b}(y=2) = 0; \mu_{y \to f_b}(y=3) = 0$

$\mu_{f_b \to x}(x) = \sum_{y=1}^{3} f_b(x,y)\mu_{y \to f_b}(y) \Rightarrow \mu_{f_b \to x}(x=1) = 0.05; \mu_{f_b \to x}(x=2) = 0.8$

$\mu_{f_a \to x}(x) = P(x) \Rightarrow \mu_{f_a \to x}(x=1) = 0.8; \mu_{f_a \to x}(x=2) = 0.2$

$$P(x \mid y=1) = \mu_{f_a \to x}(x)\mu_{f_b \to x}(x) \Rightarrow \begin{bmatrix} \mu_{f_a \to x}(1)\mu_{f_b \to x}(1) \\ \mu_{f_a \to x}(2)\mu_{f_b \to x}(2) \end{bmatrix} = \begin{bmatrix} 0.04 \\ 0.16 \end{bmatrix} \Rightarrow Normalize: \begin{bmatrix} 0.20 \\ 0.80 \end{bmatrix}$$

Sum-product Belief propagation algorithm:

Factors $\rightarrow$ Variables Messages (SUM)

Messages sent by a factor node to a variable node involves multiplying all the incoming messages (except variable node x) with the factor and summing over all the variables except x.

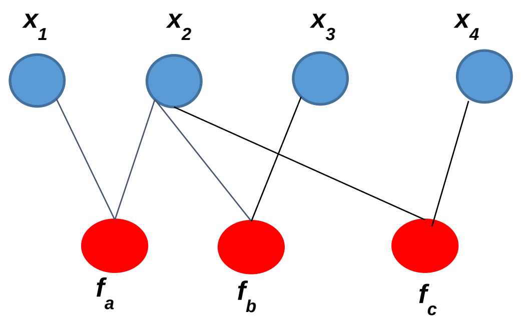
$$\mu_{f_a \to x_1}(x_1) = \sum_{x_2} f_a(x_1, x_2)\mu_{x_2 \to f_a}(x_2); \mu_{f_a \to x_2}(x_2) = \sum_{x_1} f_a(x_1, x_2)\mu_{x_1 \to f_a}(x_1)$$

$$\mu_{f_b \to x_2}(x_2) = \sum_{x_3} f_b(x_2, x_3)\mu_{x_3 \to f_b}(x_3); \mu_{f_b \to x_3}(x_3) = \sum_{x_2} f_b(x_2, x_3)\mu_{x_2 \to f_b}(x_2)$$

$$\mu_{f_c \to x_2}(x_2) = \sum_{x_4} f_c(x_2, x_4)\mu_{x_4 \to f_b}(x_4); \mu_{f_c \to x_4}(x_4) = \sum_{x_2} f_c(x_2, x_4)\mu_{x_2 \to f_c}(x_2)$$

Variables $\rightarrow$ Factors Messages (PRODUCT)

Message sent by a variable node to a factor node is the product of all the incoming messages along all of the other links (factors)



$\mu_{x_1 \to f_a}(x_1) = 1; \mu_{x_3 \to f_b}(x_3) = 1; \mu_{x_4 \to f_c}(x_3) = 1$

$\mu_{x_2 \to f_a}(x_2) = \mu_{f_b \to x_2}(x_2)\mu_{f_c \to x_2}(x_2);$

$\mu_{x_2 \to f_b}(x_2) = \mu_{f_a \to x_2}(x_2)\mu_{f_c \to x_2}(x_2);$

$\mu_{x_2 \to f_c}(x_2) = \mu_{f_a \to x_2}(x_2)\mu_{f_b \to x_2}(x_2)$

Message Passing in general: Messages passed along a link are always a function of the variable it is connected to

Use the node for which you want to compute marginal probability as the root

Factors $\rightarrow$ Variable Messages: SUM

Marginal probability of a variable x

$$p(x) = \prod_{s \in ne(x)} \sum_{\underline{X}_s} F_s(x, \underline{X}_s) = \prod_{s \in ne(x)} \mu_{f_s \to x}(x)$$

Recursively,

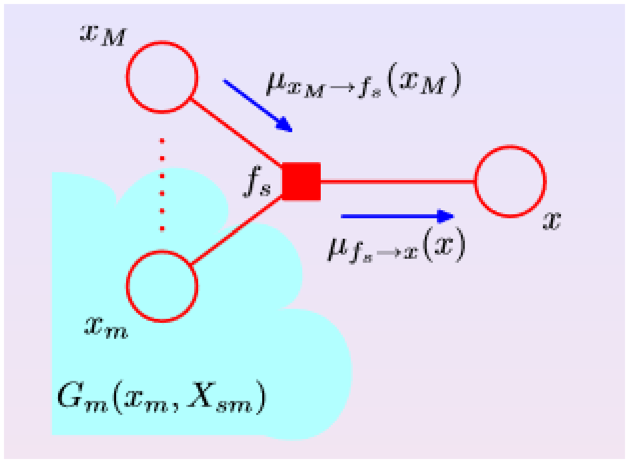
$$F_s(x, \underline{X}_s) = f_s(\underbrace{x, x_1, .., x_M}_{\underline{x}_s}) G_1(x_1, \underline{X}_{s1}) ... G_M(x_M, \underline{X}_{sM})$$

$$\mu_{f_s \to x}(x) = \sum_{x_1} ... \sum_{x_M} f_s(x, x_1, .., x_M) \prod_{m \in ne(f_s) \backslash x} \underbrace{\sum_{\underline{X}_{sm}} G_m(x_m, \underline{X}_{sm})}_{\mu_{x_m \to f_s}(x_m)}$$

$$= \sum_{x_1} ... \sum_{x_M} f_s(x, x_1, .., x_M) \prod_{m \in ne(f_s) \backslash x} \mu_{x_m \to f_s}(x_m)$$

If factor s is a leaf node with only variable x,

set $\mu_{f_s \to x}(x) = f_s(x)$



Variables $\rightarrow$ Factors Messages: Product

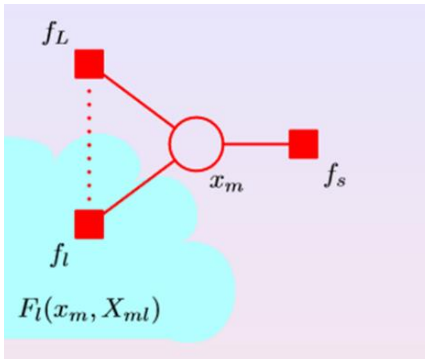
$$\mu_{x_m \to f_s}(x_m) = \sum_{\underline{X}_{sM}} G(x_M, \underline{X}_{sM})$$

$$= \sum_{\underline{X}_{sM}} \prod_{l \in ne(x_m) \backslash f_s} F_l(x_m, \underline{X}_{ml})$$

$$= \prod_{l \in ne(x_m) \backslash f_s} \sum_{\underline{X}_{ml}} F_l(x_m, \underline{X}_{ml})$$

$$= \prod_{l \in ne(x_m) \backslash f_s} \mu_{f_l \to x_m}(x_m)$$

If x_m is a leaf node, $\mu_{x_m \to f_s}(x_m) = 1$



Message sent by a variable node to a factor node is the product of all the incoming messages along all of the other links (factors)

On the other hand, messages sent by a factor node to a variable node involves multiplying all the incoming messages (except variable node x) with the factor and summing over all the variables except x

Computing all marginal probabilities:

- Select an arbitrary node as the root and propagate messages from the leaves to the root as in the sum-product algorithm for a single root node
- Send messages from the root all the way back to the leaves
- Now calculate the marginal probability at each variable and factor node via

$$p(x) = \prod_{s \in ne(x)} \mu_{f_s \to x}(x); \; p(\underline{x}_s) = f_s(\underline{x}_s) \prod_{x_i \in f_s} \mu_{x_i \to f_s}$$

- Can eliminate messages from variable nodes to factors via ... factor-factor version

$$\mu_{f_s \to x}(x) = \sum_{x_1} ... \sum_{x_M} f_s(x, x_1, .., x_M) \prod_{m \in ne(f_s) \backslash x} \mu_{x_m \to f_s}(x_m)$$

$$= \sum_{x_1} ... \sum_{x_M} f_s(x, x_1, .., x_M) \prod_{m \in ne(f_s) \backslash x} \left(\prod_{l \in ne(x_m) \backslash f_s} \mu_{f_l \to x_m}(x_m) \right)$$

MAP problem is called the Max-sum algorithm (Viterbi for Trees)

For DAGs and MRFs, multiple paths may exist. If you use sum-product the usual way, it is called Loopy Belief Propagation and it works OK!

Variational Inference and Mean Field Method for Denoising images: Ising Model

- Observed noisy image described by an array of binary pixel values

  $y_i \in \{-1,+1\}, i = 1,2,..,p$ $\boxed{\text{Variation: } p(\,y_i\,|x_i)\ \textit{Gaussian}}$

- Original (hidden) image has binary pixel values

  $x_i \in \{-1,+1\}, i = 1,2,..,p$

- Joint distribution p(x, y) is the Boltzmann distribution

  $p(\underline{x},\underline{y}) = \alpha \exp\{-E(\underline{x},\underline{y})\}$

$$E(\underline{x},\underline{y}) = \underbrace{\sum_{i=1}^{p} h_i x_i}_{\substack{\text{to bias towards}\\ -1\ or\ +1}} \underbrace{-\sum_{i=1}^{p}\sum_{j\in n_i} \beta_{ij} x_i x_j}_{\substack{\text{want energy to be small}\\ \text{when } x_i \text{ and } x_j \text{ have same sign}}} \underbrace{-\sum_{i=1}^{p} \eta_i x_i y_i}_{\substack{\text{want energy to be small}\\ \text{when } x_i \text{ and } y_i \text{ have same sign}}}$$

  Negative E is like Lyapunov function

MAP estimate via mean field (variational approximation)

$$q(\underline{x}) = \prod_{i=1}^{p} q_i(x_i,\mu_i);\ \mu_i = \text{mean value of pixel } i$$

$$\log q_i(x_i) = E_{-q_i}\{\log(p(\underline{x},\underline{y})\} + constant$$

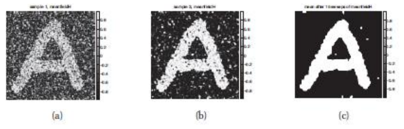
$$\log q_i(x_i) = E_{q_{-i}}[\log p(\underline{x},\underline{y})] = x_i\left(\sum_{j\in n_i}\beta_{ij}\mu_j - h_i + \eta_i y_i\right) + constant$$

$$\therefore q_i(x_i) \propto \exp(x_i\left[\left(\sum_{j\in n_i}\beta_{ij}\mu_j - h_i\right) + \eta_i y_i\right])$$

$$\Rightarrow q_i(x_i = 1) \propto \exp(a_i); q_i(x_i = -1) \propto \exp(-a_i); a_i = \left(\sum_{j\in n_i}\beta_{ij}\mu_j - h_i\right) + \eta_i y_i$$

$$\mu_i = q_i(x_i = 1) - q_i(x_i = -1) = \frac{e^{a_i} - e^{-a_i}}{e^{a_i} + e^{-a_i}} = \tanh(a_i)$$

$\mu_i^{t+1} = \lambda\mu_i^t + (1-\lambda)\tanh(a_i^t)$ I t is usually good to low pass filter the updates (λ≈0.5)



Belief propagation, Variational Inference and Active Inference:

Can we just iterate on quantities of interest? To see how, let us consider HMMs and do our forward variable in a slightly different way.

$$
\begin{aligned}
\gamma_t(x_t) = P(x_t \mid y_1, y_2, ..., y_t, y_{t+1}, ..., y_T) &= \frac{P(x_t, y_1, y_2, ..., y_t, y_{t+1}, ..., y_T)}{P(y_1, y_2, ..., y_t, y_{t+1}, ..., y_T)} \\
&= \frac{P(y_1, y_2, ..., y_{t-1}, x_t) P(y_t, y_{t+1}, ..., y_T \mid x_t, y_1, y_2, ..., y_{t-1})}{P(y_1, y_2, ..., y_t, y_{t+1}, ..., y_T)} \\
&= \frac{P(y_1, y_2, ..., y_{t-1}, x_t) P(y_t \mid x_t) P(y_{t+1}, ..., y_T \mid x_t)}{P(y_1, y_2, ..., y_t, y_{t+1}, ..., y_T)} \\
&= \frac{\tilde{\alpha}_t(x_t) P(y_t \mid x_t) \beta_t(x_t)}{\sum_{x_t} \tilde{\alpha}_t(x_t) \beta_t(x_t)}
\end{aligned}
$$

$\tilde{\alpha}_t(x_t) =$ joint probability of observing all of the data upto time $t-1$ and the value of $x_t = \sum_{x_{t-1}} P(x_t \mid x_{t-1}) P(y_{t-1} \mid x_{t-1}) \tilde{\alpha}_{t-1}(x_{t-1})$

$\beta_t(x_t) =$ conditional probability of all future data from time $(t+1)$ to T given the value of x_t

Belief Propagation:

$$\gamma_t(x_t) \propto \tilde{\alpha}_t(x_t) P(y_t \mid x_t) \beta_t(x_t) = \tilde{\alpha}_t(x_t) b_y(x_t) \beta_t(x_t)$$

$$\tilde{\alpha}_t(x_t) = \sum_{x_{t-1}} P(x_t \mid x_{t-1}) P(y_{t-1} \mid x_{t-1}) \tilde{\alpha}_{t-1}(x_{t-1}) = \sum_{x_{t-1}} P(x_t \mid x_{t-1}) b_y(x_{t-1}) \tilde{\alpha}_{t-1}(x_{t-1})$$

$$\Rightarrow \tilde{\alpha}_t(x_t) \propto \exp[\ln \gamma_t(x_t) - \ln b_y(x_t) - \ln \beta_t(x_t)]$$

$$\gamma_t(x_t) \propto \exp\{\ln b_y(x_t) + \ln E_{\tilde{\alpha}_{t-1}(x_{t-1}) b_y(x_{t-1})}[P(x_t \mid x_{t-1})] + \ln E_{\beta_{t+1}(x_{t+1}) b_y(x_{t+1})}[P(x_{t+1} \mid x_t)]\}$$

Gradient Method:

$$\dot{\underline{v}} = \varepsilon_t = \ln[P^T \underline{\tilde{\alpha}}_{t-1} \odot B^T \underline{y}_t] + \ln[P \beta_{t+1}(x_{t+1}) \odot B^T \underline{y}_t] + \ln(B^T \underline{y}_t) - \ln \underline{\gamma}_t$$

$$\underline{\gamma}_t = soft\max(\underline{v})$$

$$\underline{\tilde{\alpha}}_t = soft\max(\ln \underline{\gamma}_t - \ln(B^T \underline{y}_t) - \ln \underline{\beta}_t)$$

$$\underline{\beta}_t = soft\max(\ln \underline{\gamma}_t - \ln(B^T \underline{y}_t) - \ln \underline{\tilde{\alpha}}_t)$$

Marginal / Active Inference:

$$\dot{\underline{v}} = \varepsilon_t = \frac{1}{2}[\ln[P^T \underline{\gamma}_{t-1}] + \ln[\tilde{P} \underline{\gamma}_{t+1}] + \ln(B^T \underline{y}_t) - \ln \underline{\gamma}_t$$

$\tilde{P} = P$ with columns normalized to 1

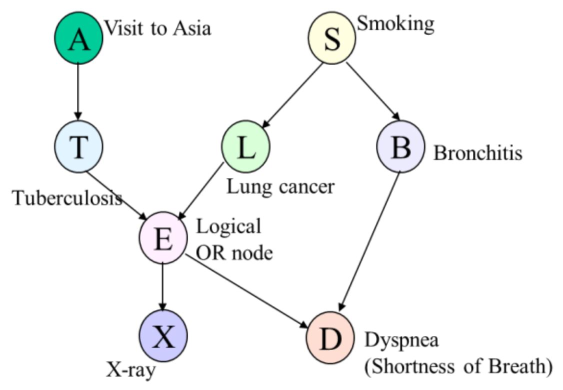
$$\underline{\gamma}_t = soft\max(\underline{v})$$

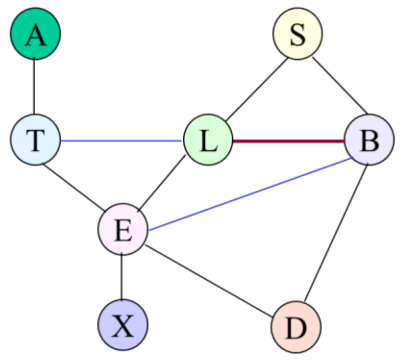
Variational:

$$\dot{\underline{v}} = \varepsilon_t = \ln[P^T \underline{\gamma}_{t-1}] + \ln[P \underline{\gamma}_{t+1}] + \ln(B^T \underline{y}_t) - \ln \underline{\gamma}_t$$

$$\underline{\gamma}_t = soft\max(\underline{v})$$

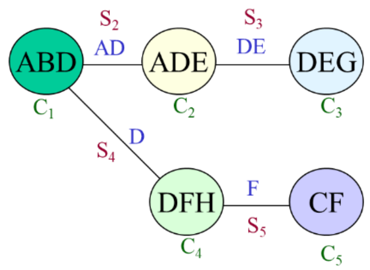
General DAGs:





Tringulated Graph

Cliques and Junction tree:



Key property: running intersection property:If a variable x is contained in two cliques, then it is contained in every clique on the path connecting the two cliques.

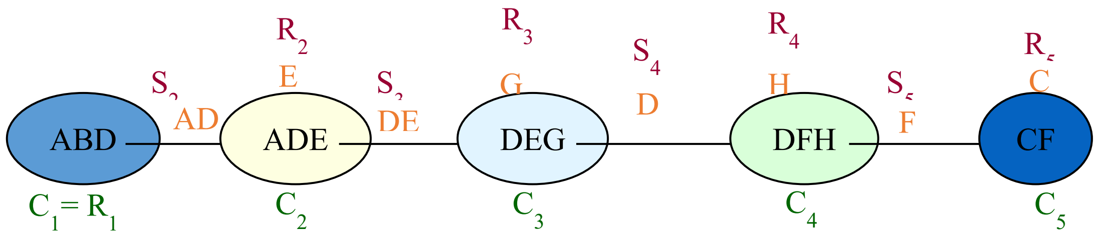
$$P(ABCDEFGH) = P(C_1)\prod_{i=2}^{5} P(C_i \mid S_i)$$

$$= P(C_1)\prod_{i=2}^{5} P(R_i \mid S_i) = \prod_{i=1}^{5} P(R_i \mid S_i); S_1 = \phi$$

$$= P(ABD)P(E \mid AD)P(G \mid DE)P(HF \mid D)P(C \mid F)$$

$$= \prod_{i=1}^{5} \psi(C_i)$$

Junction tree is not unique:



Inference on junction trees: Sum-product algorithm on junction trees. Treat cliques as factors and separators as variables

Leaf to the Root:

Operationalization in words

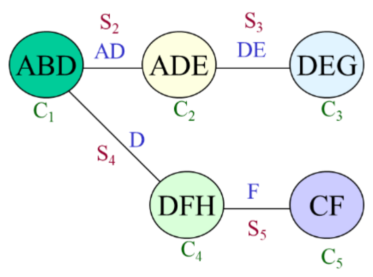
If have a clique C with incoming arcs (absorbing messages) from C_1, C_2,….,Cn with separators S1, S2,….,Sn and sending message to clique C0 with separator S

Inward Pass (Do until reaching the root). Each clique sums over its R variables and sends a message to its separator the “lambda” message. The separator passes on that message to the next clique. The clique multiplies all the separator messages and its own potential.

1. $\mathrm{Re}\,ceive\ \lambda_{C_j}(S_j)=\sum_{R_j}\psi(C_j); R_j=C_j\setminus S_j; j=1,2,\ldots,N$
2. $\psi(C)=\psi(C)\prod_{j=1}^{n}\lambda_{C_j}(S_j)$
3. $Generate\ message\ \lambda_C(S)=\sum_{C\setminus S}\psi(C)\Rightarrow\psi(C_0)=\psi(C_0)\lambda_C(S)$

Root to the Leaf: Recall trees have a single path from the root to any node. Root clique normalizes its probabilities. Each clique sums over its R variables and sends a message to its separator the “pi” message. The clique multiplies the potential with separator message and normalizes.

1. $send\ \pi_{C_0}(S)=\sum_{C_0\setminus S}P(C_0)\ to\ C$
2. $Compute\ P(C)=\psi(C)\pi_{C_0}(S)$
3. $send\ \pi_C(S_j)=\sum_{C\setminus S_j}P(C)\ to\ C_j\ ; j=1,2,..,N$
4. $Compute\ P(C_j)=\psi(C_j)\pi_C(S_j); j=1,2,..,N$



Leaf to root:

1. $\lambda_{CF}(F)=\sum_C\psi(CF);\lambda_{DEG}(DE)=\sum_G\psi(DEG)$
2. $\psi(DFH)=\psi(DFH)\lambda_{CF}(F);\psi(ADE)=\psi(ADE)\lambda_{DEG}(DE)$
3. $Generate\ message\ \lambda_{DFH}(D)=\sum_{DF}\psi(DFH)\ ;\ \lambda_{ADE}(AD)=\sum_E\psi(ADE)$
4. $\psi(ABD)=\ \psi(ABD)\lambda_{DFH}(D)\lambda_{ADE}(AD)$

Root to Leaf:

1. $\textit{Normalize}\ \Psi(ABD)$ to obtain $P(ABD)$
2. Send $\pi_{ABD}(AD)=\sum_B P(ABD)$ to ADE and $\pi_{ABD}(D)=\sum_A \pi_{ABD}(AD)$ to DFH
3. $\textit{Compute}\ \Psi(ADE)=\psi(ADE)\pi_{ABD}(AD)$ and normalize to obtain $P(ADE)$;
   $\textit{Compute}\ \Psi(DFH)=\psi(DFH)\pi_{ABD}(D)$ and normalize to obtain $P(DFH)$.
4. $\textit{send}\ \pi_{DFH}(F)=\sum_{DH} P(DFH)\ \textit{to}\ CF$ and $\pi_{ADE}(DE)=\sum_A P(ADE)$ to DEG
5. $\textit{Compute}\ \Psi(CF)=\psi(CF)\pi_{DFH}(F)$ and normalize to obtain $P(CF)$;
   $\textit{Compute}\ \Psi(DEG)=\psi(DEG)\pi_{ADE}(DE)$ and normalize to obtain $P(ADE)$.
6. Compute marginal probabilities by the smallest clique that has the variable
   Compute joint by multiplying clique probabilities and normalizing or by multiplying clique probabilities and dividing it by the product of corresponding separator variables.

Evidence is easy to incorporate.