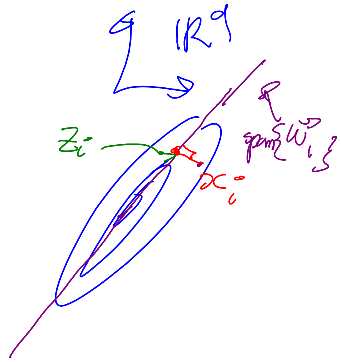
Cours 6 - scribbles

Wednesday, November 4, 2015
9:05



today :
- PCA & factor analysis
- inference : sum product alg.
  HMM

Factor analysis

$\vec{x}_i \in \mathbb{R}^d \rightsquigarrow \vec{z}_i \in \mathbb{R}^K$ $\quad K << d$

(dimensionality reduction)

synthesis view of PCA

get a orthonormal basis $\vec{w}_1, \dots, \vec{w}_K \in \mathbb{R}^d$

$$\underset{d \times K}{W} = \begin{bmatrix} | & & | \\ w_1 & \cdots & w_K \\ | & & | \end{bmatrix}$$

$$\rightarrow W^T W = \underbrace{I_K}_{K \times K \text{ identity}} \qquad \langle \vec{w}_i, \vec{w}_j \rangle = \delta_{ij}$$

orthogonal projection on subspace span$\{\vec{w}_1, \dots, \vec{w}_K\}$

$$P_W = W W^T \qquad (P_W^2 = P_W)$$

$$P_W \vec{x} = \sum_k \vec{w}_k \underbrace{\langle \vec{w}_k, \vec{x} \rangle}_{(\vec{z})_k}$$

$$\underset{\in \mathbb{R}^K}{\vec{z}} = W^T \vec{x}$$

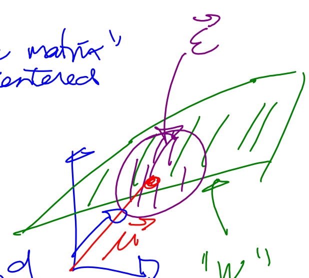
PCA $\quad \min_{\substack{W \in \mathbb{R}^{d \times K} \\ W^T W = I_K}} \sum_i \| \vec{x}_i - \underbrace{W W^T \vec{x}_i}_{W \vec{z}} \|^2$

exercise ...

$$X = \begin{pmatrix} -x_i^T- \\ \vdots \end{pmatrix}$$
$n \times d$
data matrix

"analysis" view $\quad \min \left(cst. - \sum_k \vec{w}_k^T \underbrace{X^T X}_{\text{scaled "empirical covariance matrix" if data is centered}} \vec{w}_k \right)$

generative model (latent variable model)



$\vec{z} \sim N(0, I_K)$ "somewhat uniform"

$$\vec{x} = W\vec{z} + \vec{\mu} + \underbrace{\vec{\varepsilon}}_{\text{noise}} \qquad \vec{\varepsilon} \sim N(0, \Psi) \in \mathbb{R}^d$$

diagonal
(i.e. noise indepen across

$\vec{\varepsilon} \perp\!\!\!\perp \vec{z}$

$$\vec{x}\,|\,\vec{z} \sim N(W\vec{z}+\vec{\mu},\ \Psi)$$

what is $p(\vec{x})$? $\quad E[\vec{x}] = E[\underbrace{E[\vec{x}\,|\,\vec{z}]}_{E[W\vec{z}+\mu]}]$

$$E[\underbrace{W\vec{z}}_{0}+\mu] = \vec{\mu}$$

$$\mathrm{cov}(\vec{x},\vec{x}) = \mathrm{cov}(W\vec{z}+\vec{\mu}+\vec{\varepsilon},\ W\vec{z}+\vec{\mu}+\vec{\varepsilon})$$

all independent $\qquad$ (by independence)

$$= \mathrm{cov}(W\vec{z}, W\vec{z}) + \mathrm{cov}(\vec{\varepsilon},\vec{\varepsilon})$$

$$= W\underbrace{E[\vec{z}\vec{z}^T]}_{I_k}W^T + \Psi$$

$$= WW^T + \Psi$$

$$\vec{x} \sim N(\vec{\mu},\ \underbrace{W}_{d\times k}W^T + \underbrace{\Psi}_{d \text{ degrees of freedom}})$$

to get $\vec{z}$, $\quad p(\vec{z}\,|\,\vec{x}) \qquad \begin{pmatrix}\vec{x}\\ \vec{z}\end{pmatrix} \sim N\left(\begin{pmatrix}\mu_x\\ \mu_z\end{pmatrix}, \begin{pmatrix}\Sigma_{xx} & \Sigma_{zx}\\ \Sigma_{xz} & \Sigma_{zz}\end{pmatrix}\right)$

$$\mu_x = \vec{\mu} \qquad \mu_z = 0 \qquad \Sigma_{xx} = WW^T+\Psi$$

$$\Sigma_{zz} = I_k$$

$$\mathrm{cov}(\vec{x},\vec{z}) = \mathrm{cov}(W\vec{z}+\underbrace{\mu+\vec{\varepsilon}}_{0},\ \vec{z})$$

$$= \mathrm{cov}(W\vec{z},\vec{z}) = W\overbrace{E[zz^T]}^{I_k} = W$$

$$E[\vec{z}\,|\,\vec{x}] = \mu_z + \Sigma_{zx}\Sigma_{xx}^{-1}(\vec{x}-\vec{\mu}_x)$$

$$= 0 + W^T(WW^T+\Psi)^{-1}(\vec{x}-\vec{\mu}_x)$$

$\Psi = \sigma I_d \rightarrow$ probabilistic PCA

$$\lim_{\sigma\to 0} W^T(WW^T+\sigma I)^{-1} = W^{+} \rightarrow \text{pseudo-inverse}$$

$$= W^T \text{ here (exercise)}$$

PCA representation $\qquad W^T(\vec{x}-\vec{\mu}_x)$

PCA as limit PPCA $\sigma \to 0$

to estimate $W, \Psi, \mu \rightarrow$ maximum likelihood

EM $\quad E \to p(\tilde{z} \mid \tilde{x})$

model not identifiable : $\quad WW^T + \Psi$

$$W' = WR \qquad R \text{ } K \times K \text{ rotation matrix} \quad R^T R = R R^T = I_K$$

$$W' W'^T = W \underbrace{R R^T}_{I_K} W^T = W W^T$$

why factor analysis?
- richer noise model Ψ
- easy to plug into bigger models

<u>Inference in graph model</u>

say DGM : $p(x) = \prod_i p(x_i \mid x_{\pi_i})$ $\quad$ $i \,\&\, \pi_i$ put in clique, moralization

or

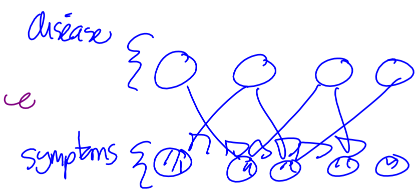
UGM $\quad p(x) = \frac{1}{Z} \prod_{C \in \mathcal{C}} \psi_C(x_C)$

want to compute:

a) marginal $\quad p(x_F) \qquad F \subseteq V$

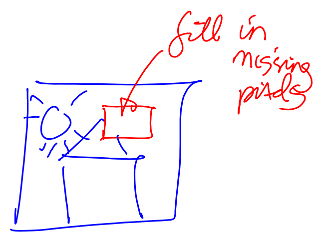
b) marginal conditional $\quad p(x_F \mid x_E)$ $\quad$ evidence

"query" usually 1 or 2 nodes



c) for UGM compute Z

why? • prediction
• fill missing data



d) related : <u>decoding</u> $\quad \arg\max_{x_F} p(x_F \mid x_E)$ $\quad$ F might be big

need inference for estimation (ML) for UGM on discrete data

$$\log p(x \mid \eta) = \sum_{C \in \mathcal{C}} \sum_{y_C} \underbrace{\mathbb{1}\{x_C = y_C\}}_{[T(x)]_{y_C}} \underbrace{\log \psi_C(x_C)}_{\eta(y_C)} - \log Z$$

$$= T(x)^T \eta - A(\eta)$$

$c \in \mathcal{C}$ y^c $\quad [T(x)]_{y_c}$ $\quad \eta(y_c)$

$$= T(x)^T\eta - A(\eta)$$

$$\nabla_\eta[\quad] = T(x) - \underbrace{\nabla_\eta A(\eta)}_{\mathbb{E}_\eta[T(X)]}$$

$$\mathbb{E}[\mathbb{1}\{y_c = x_c\}] = p(y_c)$$

aside: $|C| \leq 2$ → for binary state "Ising model"

k states "pott's model"

graph elimination

consider UGM $\quad p(x) = \frac{1}{Z}\prod_C \psi_C(x_C)$

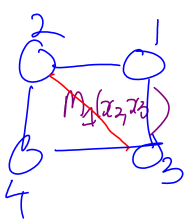
goal: to compute $p(x_F)$

main trick: distributivity $\quad \sum_{x_1, x_2} f(x_1) g(x_2)$

convince yourself $\quad \left(\sum_{x_1} f(x_1)\right)\left(\sum_{x_2} g(x_2)\right)$

$$\underbrace{\sum_{x_{1:n}}}_{\text{huge (exp) sum}} \left(\underbrace{\prod_i f_i(x_i)}_{\text{looks UGM factorization}}\right) = \prod_i \left(\underbrace{\sum_{x_i}}_{\text{small sum}} f_i(x_i)\right)$$

!!



$$p(x_4) = \sum_{x_{1:3}} \frac{1}{Z} \psi_{12}(x_{\boxed{1}}, x_2)\, \psi(x_{\boxed{1}}, x_3)\, \psi(x_2, x_4)\, \psi(x_3, x_4)$$

$$p(x_4) = \frac{1}{Z}\sum_{x_3} \psi(x_3, x_4) \underbrace{\sum_{x_2} \psi(x_2, x_4) \underbrace{\sum_{x_1} \underbrace{\psi(x_1, x_3)\psi(x_1, x_2)}_{\psi_{123}(x_1, x_2, x_3)}}_{m_1(x_2, x_3)}}_{m_2(x_3, x_4)}$$

new interactions (neighbors of 1 were tied together)

$$p(x_4) = \frac{1}{Z} m_3(x_4) \Rightarrow p(x_4) \propto m_3(x_4)$$

$Z = \sum_{x_4} m_3(x_4)$

graph eliminate algo.:

INIT: • choose ordering of elimination s.t. set F is at the end

• put all $\psi_c(x_c)$ on "active list"

UPDATE: repeat in order of elimination

suppose eliminating x_i

1) remove all factors in "active list" that x_i as argument and take product; find all α s.t. $f_\alpha(x_\alpha)$ was factor in list and $i \in \alpha$

let S_i be variables appearing in these factors excluding i

2) sum out x_i of product:

$$m_i(x_{S_i}) = \sum_{x_i} \underbrace{\left(\prod_{\alpha: i \in \alpha} f_\alpha(x_\alpha) \right)}_{\text{function}(x_{S_i \cup \{i\}})}$$

3) add $\boxed{m_i(x_{S_i})}$ on active list

new table to store $2^{|S_i|}$

new big clique

at end you're left with only factors of x_F

→ normalize to get $p(x_F)$

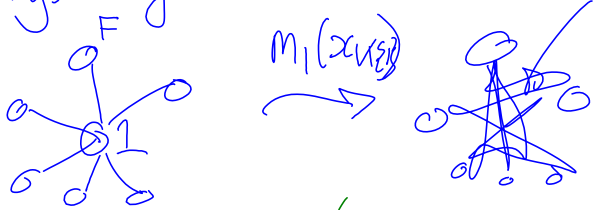
cost of alg.: store factors of size $2^{|S_i|}$

sum over $2^{|S_i|+1}$ values

to compute message $m_i(x_{S_i})$

memory & time is exponential in size of biggest clique formed during elimination

not all orderings are good



convention so that 1 for trees

treewidth of graph $= \min_{\text{over all orderings}}$ (size biggest clique appearing during elimination) -1

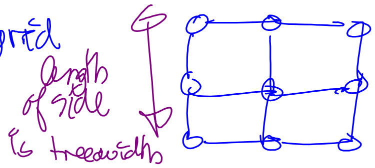
bad news:

• computing treewidth in general graph is NP-hard

(and so also finding optimal order " ")

- inference in general graph model is NP-hard

e.g. grid

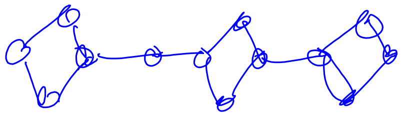


treewidth $\approx \sqrt{|V|}$

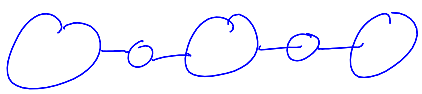
$\rightsquigarrow$ need approximations

<u>good news</u>

- exact inference for trees in linear time in size of tree
- "tree like" graph of small treewidth

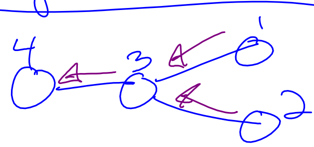


$\nearrow$ junction tree alg.



"clique tree"

Inference on a tree



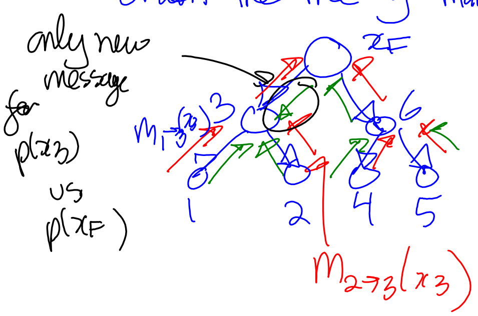
$$m_{1\to 3}(x_3) = \sum_{x_1} \psi(x_1)\psi(x_1,x_3)$$

$$p(x_4) = \frac{\psi_4(x_4)}{Z} \underbrace{\sum_{x_3} \psi_3(x_3)\psi(x_3,x_4) \sum_{x_1} \psi(x_1)\psi(x_1,x_3) \underbrace{\sum_{x_2} \psi(x_2)\psi(x_2,x_3)}_{m_{2\to 3}(x_3)}}_{m_{3\to 4}(x_4)}$$

$$p(x_4) = \frac{\psi_4(x_4)\, m_{3\to 4}(x_4)}{Z}$$

compute $p(x_F)$ where F is singleton

orient the tree by making x_F the root



$$\underset{\text{child}}{m_{i}}{}_{\to \underset{\text{parent}}{j}} = \sum_{x_i} \psi_i(x_i)\psi_{ij}(x_i,x_j) \prod_{k \in ch(i)} m_{k\to i}(x_i)$$

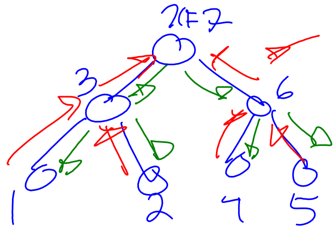
$ch(i)$: children of i

idea: use dynamic programming

idea: use dynamic programming to store messages and efficiently compute $p(x_i)\ \forall i$

$\rightarrow$ <u>sum-product alg.</u>

at end
$$p(x_F) = \frac{1}{Z}\psi_F(x_F)\prod_{k\in C(F)} m_{k\to F}(x_F)$$



"collect phase" $\rightarrow$ needed for $p(x_F)$

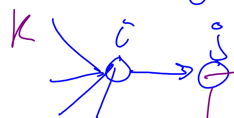
"distribute phase"

$$p(x_3) = m_{1\to 3}(x_3)\, m_{2\to 3}(x_3)\, m_{7\to 3}(x_3)\, \frac{\psi_3(x_3)}{Z}$$

general rule:

want to compute $m_{i\to j}(x_j)$, $m_{j\to i}(x_i)$ for all $\{i,j\}\in E$

rule: i send message to j if it has received all messages from <u>other neighbors</u>



$$\boxed{m_{i\to j}(x_j) = \sum_{x_i} \psi_i(x_i)\,\psi_{ij}(x_i,x_j)\cdot \prod_{k\in N(i)\setminus\{j\}} m_{k\to i}(x_i)}$$

[loopy BP: $m^{new}_{i\to j}(x_j) = \left(m^{old}_{i\to j}(x_j)\right)^{\alpha}\left(\quad\right)^{1-\alpha}$

$0<\alpha<1$ "damping"]

$\Rightarrow$ only <u>leaves</u> can start

<u>parallel schedule</u>:

1) initialize $m_{i\to j}(x_j)$ to uniform distribution

2) at every step (in parallel)

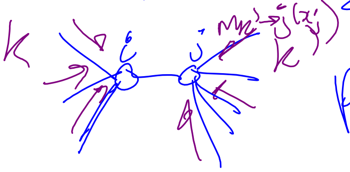
compute $m_{i\to j}(x_j) = \sum \psi_i(x_i)\,\psi_{ij}(x_i,x_j)$

x_i $\quad \prod_{k \in N(j) \setminus \{i\}} m_{k \to i}(x_i)$

→ can prove after $\underbrace{\text{diameter (tree)}}_{}$ steps, (length of longest path)
all messages have converged to correct value

<u>after sum-product</u>

$$p(x_i) = \frac{1}{Z} \psi_i(x_i) \prod_{k \in N(i)} m_{k \to i}(x_i)$$



$$p(x_i, x_j) = \frac{1}{Z} \psi_i(x_i) \psi_j(x_j) \psi_{ij}(x_i, x_j) \left(\prod_{k \in N(i) \setminus \{j\}} m_{k \to i}(x_i) \right) \left(\prod_{k' \in N(j) \setminus \{i\}} m_{k' \to j}(x_j) \right)$$

<u>conditionals:</u>

$$p(x_i \mid \bar{x}_E) \propto p(x_i, \bar{x}_E)$$

just fix $\bar{x}_E$ during sum-product (do not sum them out)

(formal trick) redefine $\tilde{\psi}_j(x_j) = \psi_j(x_j)\, \delta(x_j, \bar{x}_j)$ $\quad j \in E$

Kronecker delta

when have $\sum_{x_j} \tilde{\psi}_j(x_j) \cdot \text{stuff}(x_j, x_i)$
$= \psi_j(\bar{x}_j) \cdot \text{stuff}(\bar{x}_j, x_i)$

$$p(x_i \mid \bar{x}_E) = \frac{p(x_i, \bar{x}_E)}{\sum_{x_i'} p(x_i', \bar{x}_E)}$$

<u>remarks</u>: sum-product a.k.a. belief propagation (BP) or message passing
is exact only on trees
[approximation on graph with cycles] loopy BP

- only property used for sum product is distributivity of $\oplus$ with respect $\odot$

$(\mathbb{R}, \oplus, \odot)$ as a <u>semi-ring</u>

(↳ "don't need inverse to addition)

do sum-product on other semi-rings

$(\mathbb{R}_+, \max, \odot)$ or $(\mathbb{R}, \max, \oplus)$

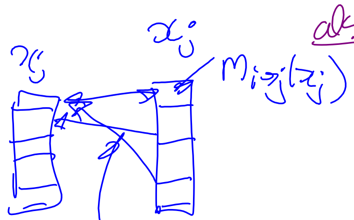
→ "max-product" $\max(a\cdot b, a\cdot c) = a\cdot \max(b,c)$

$\oplus(\quad)$

$$\max_{x_{1:n}} \prod_i f_i(x_i) = \prod_i \max_{x_i} f_i(x_i)$$

$$m_{i\to j}(x_j) = \max_{x_i}\left[\psi_i(x_i)\,\psi_{ij}(x_i,x_j)\prod_{k\in N(i)\setminus\{j\}} m_{k\to i}(x_i)\right]$$

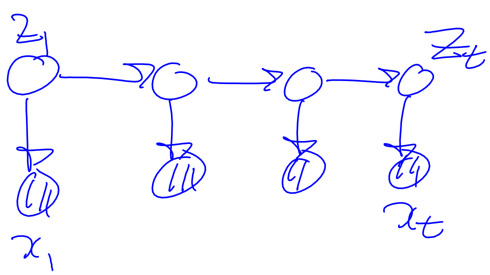
also store arg-max as a function of x_j



storing pointer to $\arg\max_{x_i} f(x_i, \overset{\text{depends on } x_j}{x_j})$

max-product aka viterbi algorithm → decoding $\arg\max_{x_{1:n}} p(x_{1:n})$

HMM: Hidden Markov Model



$z_t \in \{1,\dots,K\}$

} phoneme

} speech signal x_t < cts. / discrete

directed G.M.

$$p(x_{1:T}, z_{1:T}) = p(z_1)\prod_{t=1}^{T}\underbrace{p(x_t|z_t)}_{\text{emission prob.}}\prod_{t=2}^{T}\underbrace{p(z_t|z_{t-1})}_{\text{transition prob.}}$$

- suppose homogeneous ie $p_t(x_t|z_t) = f(x_t|z_t)$ ← does not depend on t

$$p(z_t = i \mid z_{t-1} = j) = A_{ij}$$

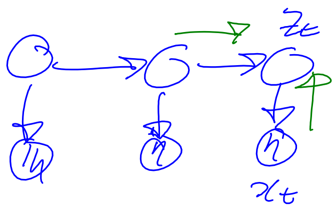
$$p(z_t = i \mid z_{t-1} = j) = A_{ij}$$

A (columns j) — prob. dist over z

need $\sum_i A_{ij} = 1$

tasks:

- prediction: $p(z_t \mid x_{1:t-1})$ "where next?"
- filtering: $p(z_t \mid x_{1:\boxed{t}})$ "where now?"
- smoothing: $p(z_t \mid x_{1:T})$, $T > t$ "where in the past?"



$$p(z_t, \bar{x}_{1:t})$$

$$= \frac{1}{1 = Z} m_{x_t \to z_t}(z_t)\, m_{z_{t-1} \to z_t}(z_t)$$

$$m_{x_t \to z_t}(z_t) = \sum_{x_t} \delta(x_t, \bar{x}_t)\, p(x_t \mid z_t)$$

$$= p(\bar{x}_t \mid z_t)$$

$$m_{z_{t-1} \to z_t}(z_t) = \sum_{z_{t-1}} p(z_t \mid z_{t-1}) \underbrace{m_{x_{t-1} \to z_{t-1}}(z_{t-1})\, m_{z_{t-2} \to z_{t-1}}(z_{t-1})}_{p(z_{t-1}, \bar{x}_{1:t-1})}$$

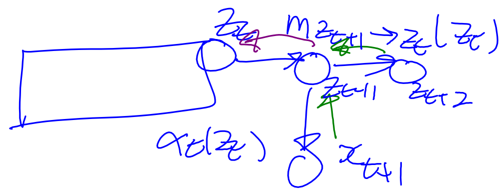
$$\boxed{\alpha_t(z_t) = p(z_t, \bar{x}_{1:t}) = p(\bar{x}_t \mid z_t) \sum_{z_{t-1}} p(z_t \mid z_{t-1})\, \overbrace{\alpha_{t-1}(z_{t-1})}^{\alpha_{t-1}(z_{t-1})}}$$

α-recursion forward recursion

like the "collect phase" in sumproduct

$$p(\bar{x}_{1:t}) = \sum_{z_t} \alpha_t(z_t)$$

for smoothing, also need backward message $\beta_t(z_t)$

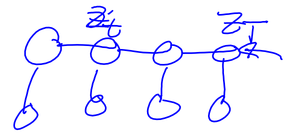


$$m_{z_{t+1}\to z_t}(z_t) = \sum_{z_{t+1}} p(\bar{x}_{t+1}|z_{t+1}) \underbrace{m_{z_{t+2}\to z_{t+1}}(z_{t+1})}_{\beta_{t+1}(z_{t+1})}$$

→ β-recursion
(backward pass)

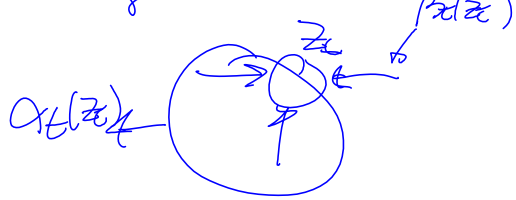
Initialization:

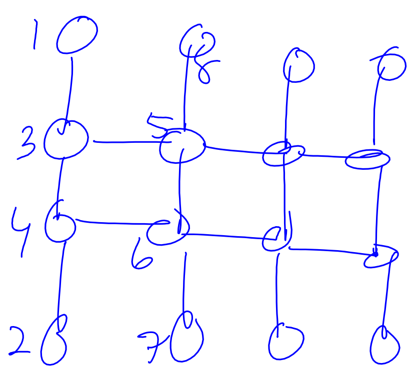
$$\beta_T(z_T) = 1$$

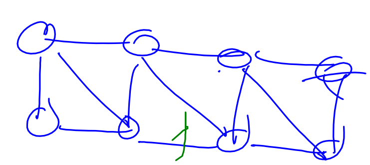


$$\alpha_1(z_1) = p(\bar{x}_1|z_1)\,p(z_1)$$

$$p(z_t, \bar{x}_{1:T}) = \alpha_t(z_t)\,\beta_t(z_t)$$







clique tree
running intersection property