

Lecture 15 - scribbles

Tuesday, October 29, 2019 14:30

- today:
- max product alg.
 - junction tree
 - HMM

max-product alg:

for sum-product, main property used was distributivity of \oplus over \odot

all need $\mathcal{M}(\mathbb{R}, \oplus, \odot)$ is a semi-ring

↳ don't need additive inverse

⊗ can do "sum-product" on other semi-rings

$(\mathbb{R}, \max, +)$ $\max(a+b, a+c) = a + \max(b, c)$

$(\mathbb{R}_+, \max, \cdot)$ $\max(a \cdot b, a \cdot c) = a \cdot \max(b, c)$

↳ "max-product"

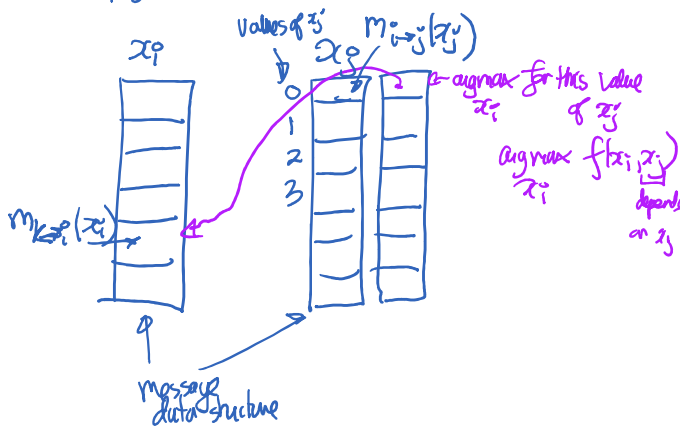
$\max_{x_1:n} \prod_i f_i(x_i) = \prod_i \max_{x_i} f_i(x_i)$

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



↳ for getting argmax, store argument of this max as a fct. of x_j

$\max_{x_1:n} \prod_c \psi_c(x_c) = \prod_c \max_{x_c} m_{2 \rightarrow 1}(x_1)$



↳ to get argmax $f(x_1:n)$ "decoding" $x_1:n$

- run max-product alg (only forward messages)
- backtrack the argmax pointers to get full argmax

aka. viterbi algorithm

property of tree UGM

property of tree UGM

$p \in \mathcal{L}(\text{tree})$
with non-zero marginals

$$\Rightarrow p(x) = \prod_{i \in V} p(x_i) \prod_{(i,j) \in E} \frac{p(x_i, x_j)}{p(x_i)p(x_j)}$$

(proof: simple exercise)

* similar to DGM; for any set of factors $\{f_{ij}(x_i, x_j)\}, \{f_i(x_i)\}$ $f_{ij} \geq 0$
 $f_i \geq 0$

s.t.
"local consistency property"

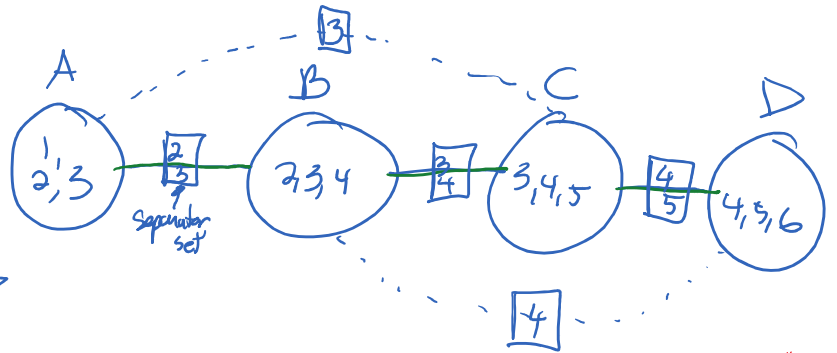
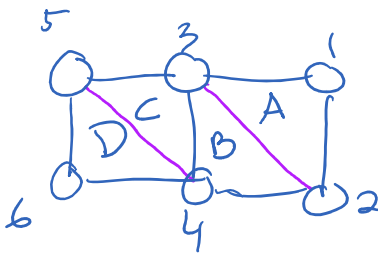
$$\begin{cases} \sum_{x_j} f_{ij}(x_i, x_j) = f_i(x_i) & \forall x_i \\ \sum_{x_i} f_{ij}(x_i, x_j) = f_j(x_j) & \forall x_j \\ \sum_{x_i} f_i(x_i) = 1 \end{cases}$$

then if define joint
(for tree)

$$p(x) = \prod_i f_i(x_i) \prod_{(i,j) \in E} \frac{f_{ij}(x_i, x_j)}{f_i(x_i)f_j(x_j)}$$

then we get correct marginals (e.g. $p(x_i) = f_i(x_i)$) etc...

Junction tree algorithm: generalization of sum-product to a clique tree
(with JT property)



above is a clique tree with the "running intersection property" ← "junction tree"

to build a JT: • use max weight spanning tree alg. on clique graph (with size of separator sets as weight on the edges)

JJT \Leftrightarrow triangulated graph / decomposable graph \Leftrightarrow running graph eliminate

when have J.T., one can show

$$p(x_r) = \frac{\prod_C p(x_c)}{\prod_S p(x_s)}$$

separator set in the J.T.

J.T. tree dg.: reconstruct the above formulation

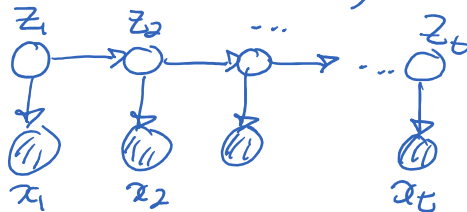
by starting with $p(x_c) = \frac{\prod_S \psi_S(x_c)}{\prod_S \psi_S(x_s)}$ where $\psi_S(x_c) = 1$ at leaf.

do message passing on J.T. to update ψ_C^{new} at the end

$p(x_c)$
 $p(x_s)$

15h35

HMM (hidden Markov model)

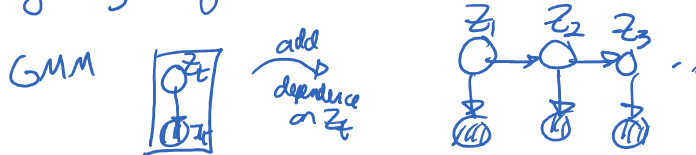


$z_t \in \{1, \dots, k\}$ discrete

(later $z_t \sim \text{Gaussian}$)
 \rightarrow Kalman filter

x_t — cts, e.g. speech signal
discrete e.g. DNA sequence

HMM \rightarrow generalization of mixture model



$$p(x_{1:T}, z_{1:T}) = p(z_1) \prod_{t=1}^T p(x_t | z_t) \prod_{t=2}^T p(z_t | z_{t-1})$$

emission probs transition probs

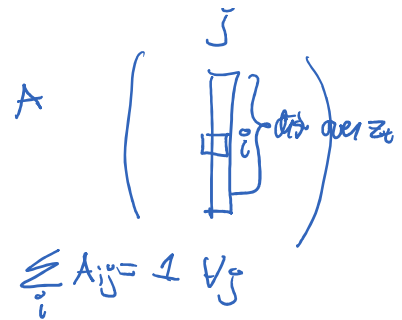
often, emission probs & trans. probs are homogeneous in time (i.e. do not depend on t)

$$p_t(x_t | z_t) = f(x_t | z_t)$$

A $\int \int \int$

$$p_t(x_t | z_t) = f(x_t | z_t)$$

$$p_t(z_t = i | z_{t-1} = j) = A_{ij}$$



"stochastic matrix" $\sum_i A_{ij} = 1 \forall j$

inference tasks:

prediction $p(z_t | x_{1:t-1})$ "where next?"

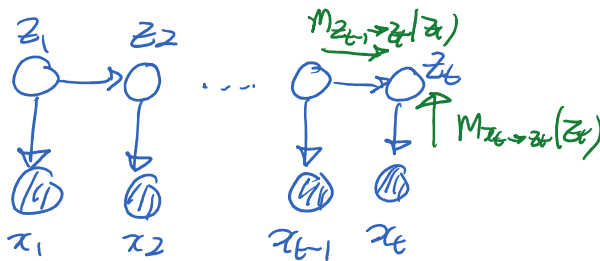
filtering $p(z_t | x_{1:t})$ "where now?"

smoothing $p(z_t | x_{1:T})$ "where in the past?"

$T > t$

α -recursion:

Let's run sum product here to derive recursions to compute prob.



to compute $p(z_t, \bar{x}_{1:t}) \propto p(z_t | \bar{x}_{1:t})$

$$p(z_t, \bar{x}_{1:t}) = \frac{1}{Z} 1 \cdot m_{z_{t-1} \rightarrow z_t}(z_t) M_{z_t \rightarrow z_t}(z_t) \quad (\text{here } z=1)$$

$\alpha_t(z_t)$

$$M_{z_t \rightarrow z_t}(z_t) = \sum_{x_t} p(x_t | z_t) \delta(x_t, \bar{x}_t) = p(\bar{x}_t | z_t)$$

$$m_{z_{t-1} \rightarrow z_t}(z_t) = \sum_{z_{t-1}} p(z_t | z_{t-1}) \underbrace{m_{z_{t-2} \rightarrow z_{t-1}}(z_{t-1}) M_{z_{t-1} \rightarrow z_{t-1}}(z_{t-1})}_{p(z_{t-1}, \bar{x}_{1:t-1})}$$

$$\alpha_{t-1}(z_{t-1})$$

$$\alpha_t(z_t) = \underbrace{p(\bar{x}_t | z_t)}_{\text{vector}(z_t)} \sum_{z_{t-1}} \underbrace{p(z_t | z_{t-1})}_{\text{matrix}} \underbrace{\alpha_{t-1}(z_{t-1})}_{\text{vector}}$$

α -recursion a.k.a. "forward recursion" like the "collect phase" in sum-product alg

with z_t as the rest

$$\text{let } \alpha_t(z_t) \triangleq p(\bar{x}_t | z_t)$$

$$\alpha_t = \alpha_{t-1} \underset{\text{Hadamard product}}{\odot} (A \alpha_{t-1})$$

initialization:

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

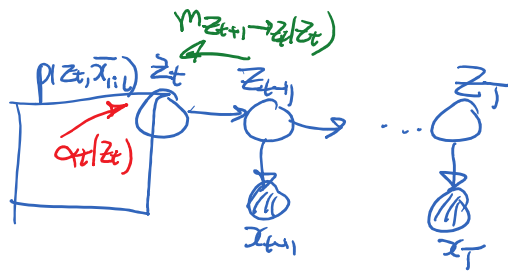
$$\tilde{\alpha}_t \triangleq p(z_t | x_{1:t}) \text{ "filtering distribution"}$$

time complexity : $O(tk^2)$

space complexity : $O(k)$ extra storage

$$\sum_{z_t} \underbrace{p(z_t) \tilde{\alpha}_t(z_t)}_{\alpha_t(z_t)} = p(\bar{x}_{1:t}) \text{ "evidence probability"}$$

β -recursion (smoothing)



$$p(z_t, \bar{x}_{1:T}) = \prod_{i=1}^T \alpha_i(z_i) \underbrace{m_{z_{t+1} \rightarrow z_t}(z_t)}_{\triangleq \beta_t(z_t)}$$

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

$$\beta_t(z_t) = \sum_{z_{t+1}} p(z_{t+1} | z_t) p(\bar{x}_{t+1} | z_{t+1}) \beta_{t+1}(z_{t+1})$$

β -recursion (aka backward recursion)

turns out that $\beta_t(z_t) \triangleq p(\bar{x}_{t+1:T} | z_t)$

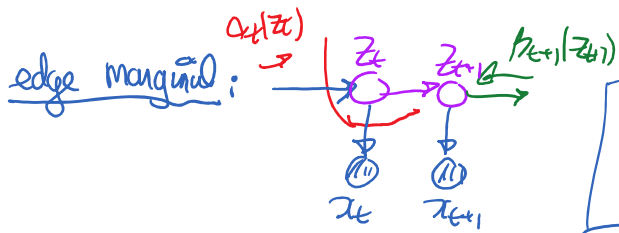
why? \downarrow smoothing prob

$$p(z_t, \bar{x}) = p(\bar{x} | z_t) p(z_t)$$

$$\stackrel{\text{c.t.d.}}{=} \underbrace{p(\bar{x}_{t+1:T} | z_t)}_{\beta_t(z_t)} \underbrace{p(\bar{x}_t | z_t)}_{\alpha_t(z_t)}$$

initialization : $\beta_t(z_T) = 1 \quad \forall z_T$

also $m_{z_{t+1} \rightarrow z_t}(z_t) \rightarrow \alpha_t(z_t) \quad p(z_{t+1} | z_t)$



$$p(z_t, z_{t+1}, \bar{x}_{1:T}) = \alpha_t(z_t) \beta_{t+1}(z_{t+1}) \cdot p(z_{t+1} | z_t) \cdot p(\bar{x}_{t+1} | z_{t+1})$$