Lecture 15 - max-product, junction tree; HMM

Tuesday, November 1, 2022 2:57 PM

today: • max product alg
• junction tree
• HMM

getting conditionals:

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

indicates values we are conditioning on

keep this fixed during marginalization for each $j \in E$

(formal trick): redefine $\tilde{\psi}_j(x_j) \triangleq \psi_j(x_j) \cdot \delta(x_j, \bar{x}_j)$

Kronecker-delta fct. $\delta(a,b) \triangleq \begin{cases} 1 & \text{when } a=b \\ 0 & \text{o.w.} \end{cases}$

computing $m_{j \to i}(x_i) \leftarrow \sum_{x_j} \tilde{\psi}_j(x_j)\, \text{shuff}(x_j, x_i)$

$= \psi_j(\bar{x}_j)\, \text{shuff}(\bar{x}_j, x_i)$

at the end, result of sum-product will give $p(x_i, \bar{x}_E) = \frac{1}{Z} \psi_i(x_i) \prod_{k \in N(i)} m_{k \to i}(x_i)$

renormalize over x_i to get $p(x_i \mid \bar{x}_E)$

max-product alg.

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

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

↳ ring don't need additive inverses

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

$[a \cdot b + a \cdot c = a \cdot (b+c)]$

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

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

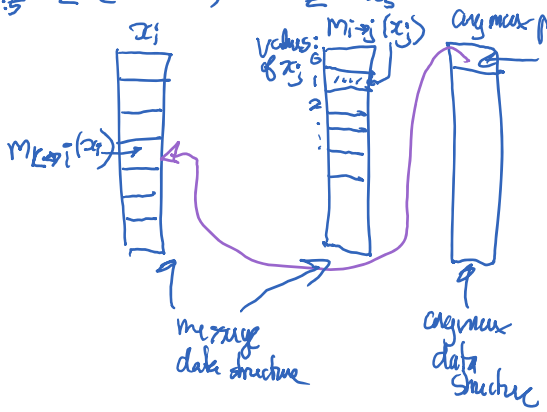
↳ "max-product" $\quad \max_{x_{1:n}} \prod_i f_i(x_i) = \prod_{i=1}^{n} \max_{x_i} f_i(x_i)$

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

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

x_i → for getting argmax store argument of this max as a fct. of x_j

$$\max_{x_{1:5}} \frac{1}{Z} \prod_c \psi_c(x_c) = \frac{1}{Z} \max_{x_5} \left[m_{4\to 5}(x_5)\psi_5(x_5) \right]$$



* to get $\operatorname{argmax}_{x_{1:n}} p(x_{1:n})$ "decoding"
- run max-product alg (only forward messages)
- backtrack the argmax pointers to get full argmax

aka Viterbi algorithm

property of tree UGM

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

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

($\psi_i(x_i)$, $\psi_{ij}(x_i,x_j)$)

* 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"

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

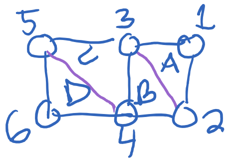
Then if define joint $p(x) = \prod_i f_i(x_i) \prod_{i,j\in E} \frac{f_{ij}(x_i,x_j)}{f_i(x_i)f_j(x_j)}$ (for tree)

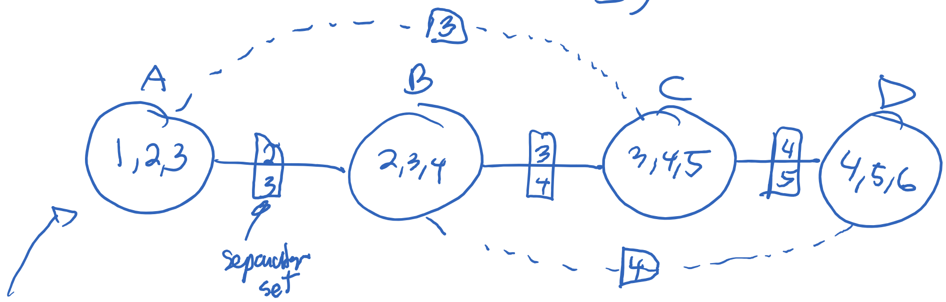
we could store then we get correct marginals i.e. $p(x_i) = f_i(x_i)$ etc...

break: 15h54

junction tree algorithm: generalization of sum-product to a clique tree

junction tree algorithm: generalization of sum-product to a clique tree (with J.T. property)





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

↳ if $j \in C_1 \cap C_2$, then $j \in C_j \; \forall C_j$ along path C_1 to C_2

to build a J.T.: on a Δ-graph

- use maximum weight spanning tree alg. on clique graph (with size of separator sets as weight on edges) ⟹ has running intersection property

∃ a J.T. ⟺ triangulated / decomposable graph ⟵ running graphEliminate

⊛ when have a J.T., one can show

$$p(x_V) = \frac{\prod_{C} p(x_C)}{\prod_{S} p(x_S)}$$

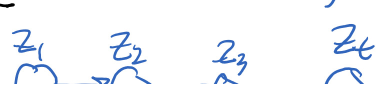
S: separator sets of J.T.

J.T. alg.: reconstruct the above formulation

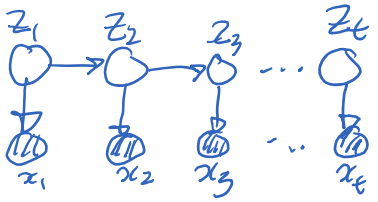
by starting with $p(x_V) = \frac{1}{Z} \frac{\prod_C \psi_C(x_C)}{\prod_S \varphi_S(x_S)}$ where $\varphi_S(x_S) = 1$ at initialization

do message passing on J.T. to update ψ_C^{new}, φ_S^{new} —at the end→ $p(x_C)$, $p(x_S)$

HMM (hidden Markov model)



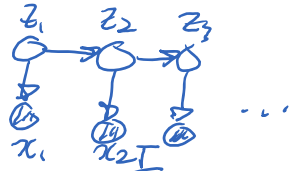
$z_t \in \{1, \dots, K\}$ discrete ($x_t | z_t \sim$ Gaussian)



$z_t \in \{1,\dots,k\}$ discrete (later $z_t \sim$ Gaussian → kalman filter)

x_t ← cts. eg speech signal
discrete eg. DNA sequence

HMM → generalization of mixture model

GMM  add dependence on time 

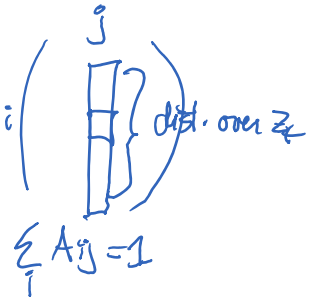
$$\text{DGM}: p(x_{1:T}, z_{1:T}) = p(z_1)\prod_{t=1}^{T} p(x_t|z_t)\prod_{t=2}^{T} p(z_t|z_{t-1})$$

emission prob. transition prob.

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

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

("stochastic matrix")



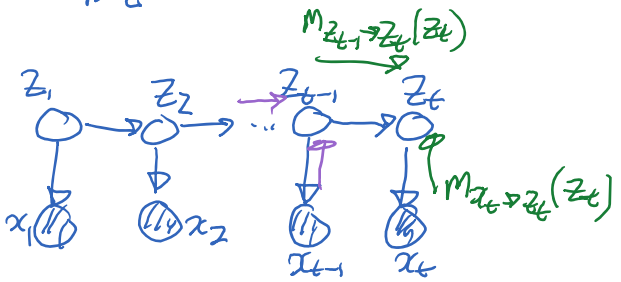
$\sum_i A_{ij} = 1$

inference tasks:

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

α-recursion:

Let's run sum-product alg. to derive recursion to compute $p(z_t \mid x_{1:t})$



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

$$\underbrace{p(z_t, \bar{x}_{1:t})}_{\alpha_t(z_t)} = \frac{1}{Z} 1 \cdot m_{z_{t-1}\to z_t}(z_t) \cdot m_{x_t \to z_t}(z_t) \quad (\text{now } Z=1)$$

$$m_{x_t\to z_t}(z_t) = \sum_{x_t} p(x_t|z_t)\,\underbrace{\delta(x_t, \bar{x}_t)}_{\text{conditioning trick}} = p(\bar{x}_t|z_t)$$

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

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

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

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

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

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

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

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

$$\alpha_t = O_t \odot (A\alpha_{t-1})$$

Hadamard

space complexity: $O(K)$ extra storage

time complexity $O(tK^2)$

$\tilde{\alpha}_t(z_t) \triangleq p(z_t|\bar{x}_{1:t})$ "filtering distribution"

$= \alpha_t(z_t)$

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

"evidence prob."