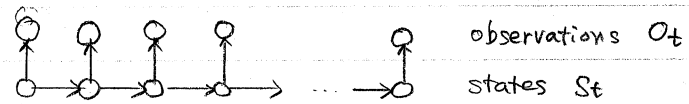
11/4 $\boxed{\text{Review}}$

* Hidden Markov models (HMMs)



* Parameters

$\pi_i = P(S_1 = i)$

$a_{ij} = P(S_{t+1} = j \mid S_t = i)$

$b_{ik} = P(O_t = k \mid S_t = i)$

* Key computations

1) how to compute likelihood $P(O_1, O_2, \cdots, O_T)$?

2) how to compute $\underset{s_1,\cdots,s_T}{\operatorname{argmax}} P(s_1, \cdots s_T \mid O_1, \cdots, O_T)$?

3) how to estimate (learn) $\{\pi_i, a_{ij}, b_{ik}\}$?

* HW problem: belief updating

— recursion for $P(S_t \mid O_1, O_2, \cdots, O_t)$

— important for real-time monitoring.

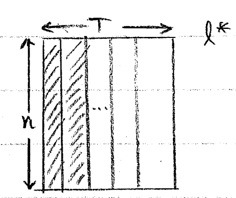
(2) Computing most likely state sequence

$l_{it}^* = \max_{\{S_1,\cdots,S_{t-1}\}} \log P(S_1, S_2, \cdots, S_{t-1}, S_t = i, O_1, \cdots, O_t)$

Recursion: $l_{j\,t+1}^* = \max_i \left[l_{it}^* + \log a_{ij} \right] + \log b_j(O_{t+1})$

How to derive s^* from l^*?

discrete state sequence — real-valued $n \times l$ matrix



* Record most likely state transitions:

$\Phi_{t+1}(j) = \underset{i}{\operatorname{argmax}} \left[l_{it}^* + \log a_{ij} \right]$

What is most likely state at time t given state j at time $t+1$ with observations $O_1, O_2, \cdots, O_t$.

* Compute s^* by backtracking:

$S_T^* = \underset{i}{\operatorname{argmax}} \left[l_{iT}^* \right]$

$S_t^* = \Phi_{t+1}(S_{t+1}^*) \qquad t = T-1, T-2, \cdots, 1$

$S^* = \{S_1^*, S_2^*, \cdots, S_T^*\}$ is known as Viterbi path.

Viterbi algorithm is instance of dynamic programming.

(3) Learning in HMMs.

Given: sequence of observations $\{O_1, O_2, \cdots, O_T\}$ (just one for simplicity)

Goal: estimate $\{\pi_i, a_{ij}, b_{ik}\}$ to maximize $P(O_1, O_2, \cdots, O_T)$

Fixed: number of hidden states n.

* Shared CPTs to estimate: $\pi_i = P(S_1 = i)$

$$a_{ij} = P(S_{t+1} = j \mid S_t = i)$$
$$b_{ik} = P(O_t = k \mid S_t = i)$$

* <u>E-step</u>

Compute $P(S_1 = i \mid O_1, \cdots, O_T)$

$$P(S_t = i, S_{t+1} = j \mid O_1, \cdots, O_T)$$
$$P(S_t = i, O_t = k \mid O_1, \cdots, O_T) = P(S_t = i \mid O_1, \cdots, O_T)\, I(O_t, k)$$

(analogy: pa , x , V)

* Compute posterior probabilities

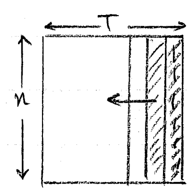
Analogous to $\alpha_{it} = P(O_1, O_2, \cdots, O_t, S_t = i)$ before and up to time t

define $\beta_{it} = P(\underbrace{O_{t+1}}_{\text{Starting at } t+1}, O_{t+2}, \cdots, O_T \mid \underbrace{S_t = i}_{\text{Conditioning on } S_t = i})$ after time t

- Recursion for β_{it}:

(1) base case $\beta_{iT} = P(\cdots \mid S_T = i)$?

$\beta_{iT} = 1$ for all i.



(2) backwards step

$$\beta_{it} = P(O_{t+1}, \cdots, O_T \mid S_t = i)$$
$$= \sum_{j=1}^{n} P(O_{t+1}, \cdots, O_T, S_{t+1} = j \mid S_t = i) \quad \text{marginalization}$$
$$= \sum_{j=1}^{n} P(O_{t+1}, \cdots, O_T \mid S_{t+1} = j, \cancel{S_t = i})\, P(S_{t+1} = j \mid S_t = i) \quad \text{product rule}$$

conditional independence

$$= \sum_{j=1}^{n} P(O_{t+1} \mid S_{t+1} = j)\, P(O_{t+2}, \cdots, O_T \mid S_{t+1} = j, \cancel{O_{t+1}})\, P(S_{t+1} = j \mid S_t = i) \quad \text{product rule}$$
$$= \sum_{j=1}^{n} b_j(O_{t+1})\, \beta_{jt+1}\, a_{ij}$$

Computing α & β matrix $\longleftrightarrow$ "forward-backward" algorithm (a.k.a Baum-Welch)

- Posterior probabilities in E-step:

$$P(S_t = i, S_{t+1} = j \mid O_1, \cdots, O_T) = \frac{P(S_t = i, S_{t+1} = j, O_1, \cdots, O_T)}{P(O_1, \cdots, O_T)}$$

$$= \frac{P(O_1, \cdots, O_t, S_t = i)\, P(S_{t+1} = j \mid S_t = i, \cancel{O_1, \cdots, O_t})\, P(O_{t+1} \mid S_{t+1} = j, \cancel{S_t = i, O_1, \cdots, O_t}) \times P(O_{t+2}, \cdots, O_T \mid S_{t+1} = j, \cancel{S_t = i, O_1, \cdots, O_{t+1}})}{P(O_1, O_2, \cdots, O_T)}$$

$$= \frac{\alpha_{it}\, a_{ij}\, b_j(O_{t+1})\, \beta_{jt+1}}{\sum_i \alpha_{iT}}$$

Other posteriors in E-step

$$P(S_t = i \mid O_1, \cdots, O_T) = \sum_{j=1}^{n} P(S_t = i, S_{t+1} = j \mid O_1, \cdots, O_T)$$

$P(S_1 = i \mid O_1, \cdots, O_T)$ special case with $t = 1$.

* M-step updates

$$\pi_i \longleftarrow P(S_1 = i \mid O_1, \cdots, O_T)$$

$$a_{ij} \longleftarrow \frac{\sum_t P(S_t = i, S_{t+1} = j \mid O_1, \cdots, O_T)}{\sum_t P(S_t = i \mid O_1, \cdots, O_T)}$$

$$b_{ik} \longleftarrow \frac{\sum_t P(S_t = i \mid O_1, \cdots, O_T)\, I(k, O_t)}{\sum_t P(S_t = i \mid O_1, \cdots, O_T)}$$

* Complexity of HMM computations

(i) to compute $P(O_1, O_2, \cdots, O_T)$

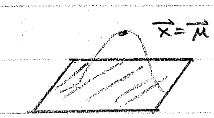
(ii) to decode $s^* = \arg\max_{\vec{s}} P(\vec{s} \mid \vec{o})$

(iii) parameter update of EM

$\Big\}\ O(n^2 T)$ $\quad T$ = sequence length, n = # states.

Multivariate Gaussian distributions

* Random variable $\vec{x} \in \mathbb{R}^n$ (real-valued vector)



* Probability density function (PDF)

$$P(\vec{x}) = \frac{1}{(2\pi)^{n/2} \sqrt{\det(\Sigma)}} \exp\left\{ -\frac{1}{2} (\vec{x} - \vec{\mu})^T \Sigma^{-1} (\vec{x} - \vec{\mu}) \right\}.$$

* Parameters

$$\vec{\mu} = E[\vec{x}] = \int P(\vec{x})\, \vec{x}\, d^n\vec{x}$$

$$\Sigma_{ij} = E[(\vec{x} - \vec{\mu})_i (\vec{x} - \vec{\mu})_j] = \int P(\vec{x})(x_i - \mu_i)(x_j - \mu_j)\, d^n x.$$

Shorthand : $P(\vec{x}) = N(\vec{x}; \vec{\mu}, \Sigma)$

* Mathematical properties :

(i) if $P(\vec{x})$ and $P(\vec{y})$ are gaussian PDFs over $\vec{x}, \vec{y} \in \mathbb{R}^n$,

then $P(\alpha\vec{x} + \beta\vec{y})$ is also gaussian.

α, β are linear scalar coefficients.

(ii) if $P(\vec{x})$ is gaussian over $\vec{x} \in (X_1, X_2, \cdots, X_n) \in \mathbb{R}^n$,

then all marginals $\{P(X_i), P(X_i, X_j), \cdots\}$ and all conditionals $\{P(X_i \mid X_j), P(X_i, X_j \mid X_k), \cdots\}$

are also gaussian.

* Maximum likelihood estimation

Given i.i.d data $\{\vec{x}_t\}_{t=1}^T$ where $\vec{x}_t \in \mathbb{R}^n$, how to choose $\vec{\mu}, \Sigma$

to maximize $P(\text{data}) = \prod_{t=1}^{T} N(\vec{x}_t; \vec{\mu}, \Sigma)$

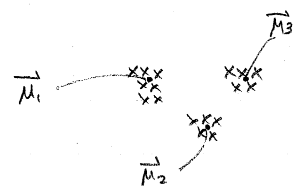
– log-likelihood

$$\mathcal{L} = \log P(\text{data}) = \sum_t \log N(\vec{x}_t; \vec{\mu}, \Sigma)$$

To maximize : $\frac{\partial \mathcal{L}}{\partial \vec{\mu}} = 0$, $\frac{\partial \mathcal{L}}{\partial \Sigma_{ij}} = 0$.

$\vec{\mu} = \frac{1}{T} \sum_{t=1}^{T} \vec{X}_t$ sample mean

$\Sigma_{ij} = \frac{1}{T} \sum_t (\vec{X}_t - \vec{\mu})_i (\vec{X}_t - \vec{\mu})_j$ sample covariance.



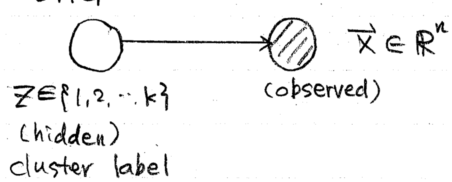
Ex: $n=2$ dimensions
$k=3$ clusters.

$\boxed{\text{Clustering}}$

* Inputs $\{\vec{x}_1, \vec{x}_2, \cdots, \vec{x}_T\}$ with $\vec{x}_t \in \mathbb{R}^n$
* Goal : partition inputs into k clusters.

$\boxed{\text{Gaussian mixture model}}$

* DAG



* CPTs

$P(Z=i)$ fraction of data in cluster i

$P(\vec{X} \mid Z=i) = N(\vec{X} ; \vec{\mu}_i, \Sigma_i)$ cluster-dependent means and covariance matrices.

* Aside : ML estimation for complete data $\{(\vec{x}_t, z_t)\}_{t=1}^{T}$

Let $T_i = \sum_{t=1}^{T} I(z_t, i)$ count of label i.

$P(z_i) = \frac{T_i}{T}$

$\vec{\mu}_i = \frac{1}{T_i} \sum_{t=1}^{T} \vec{x}_t \, I(z_t, i)$

$\Sigma^i_{\alpha\beta} = \frac{1}{T_i} \sum_{t=1}^{T} (\vec{x}_t - \vec{\mu}_i)_\alpha (\vec{x}_t - \vec{\mu}_i)_\beta \, I(z_t, i)$