CSE 250A. Principles of AI

Probabilistic Reasoning and Decision-Making

Lecture 15 – HMMs and Clustering with GMMs

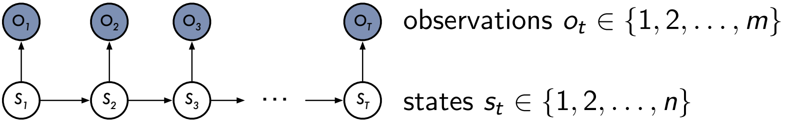
Lawrence Saul
Department of Computer Science and Engineering
University of California, San Diego

Fall 2021

Outline

Hidden Markov models

- **Belief network**



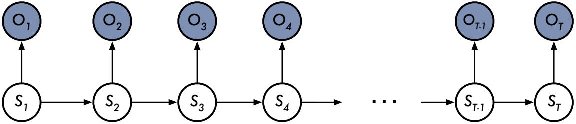
- **Parameters**

$$a_{ij} = P(S_{t+1}=j|S_t=i) \qquad \textbf{transition matrix}$$
$$b_{ik} = P(O_t=k|S_t=i) \qquad \textbf{emission matrix}$$
$$\pi_i = P(S_1=i) \qquad \textbf{initial state distribution}$$

- **Notation**

Sometimes we'll write $b_i(k) = b_{ik}$ to avoid double subscripts.

Key computations in HMMs



Inference

1. How to compute the likelihood $P(o_1, o_2, \ldots, o_T)$? ✓
2. How to compute the most likely hidden states $\text{argmax}_{\vec{s}}\, P(\vec{s}|\vec{o})$? ✓
3. How to update beliefs by computing $P(s_t|o_1, o_2, \ldots, o_t)$? HW 7.4

Learning

How to estimate parameters $\{\pi_i, a_{ij}, b_{ik}\}$ that maximize the log-likelihood of observed sequences? **TODAY**

Forward-backward algorithm

- **Matrices to compute:**

$$
\begin{aligned}
\alpha_{it} &= P(o_1, o_2, \ldots, o_t, S_t = i) \\
\beta_{it} &= P(o_{t+1}, o_{t+2}, \ldots, o_T | S_t = i)
\end{aligned}
$$

- **Forward pass:**

$$
\begin{aligned}
\alpha_{i1} &= \pi_i \, b_i(o_1) \\
\alpha_{j,t+1} &= \sum_{i=1}^{n} \alpha_{it} \, a_{ij} \, b_j(o_{t+1})
\end{aligned}
$$

$$
n \text{ rows} \left\{ \begin{bmatrix} \boxed{\begin{matrix} \alpha_{11} \\ \alpha_{21} \\ \vdots \\ \alpha_{n1} \end{matrix}} & \begin{matrix} \alpha_{12} \\ \alpha_{22} \\ \vdots \\ \alpha_{n2} \end{matrix} & \begin{matrix} \cdots \\ \cdots \\ \vdots \\ \cdots \end{matrix} & \begin{matrix} \alpha_{1,T-1} \\ \alpha_{2,T-1} \\ \vdots \\ \alpha_{n,T-1} \end{matrix} & \begin{matrix} \alpha_{1T} \\ \alpha_{2T} \\ \vdots \\ \alpha_{nT} \end{matrix} \end{bmatrix} \right.
$$

- **Backward pass:**

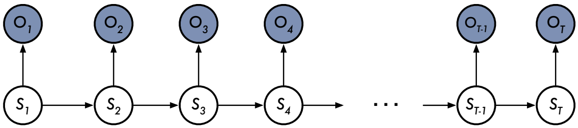
$$
\begin{aligned}
\beta_{iT} &= 1 \text{ for all } i \in \{1, 2, \ldots, n\} \\
\beta_{it} &= \sum_{j=1}^{n} a_{ij} \, b_j(o_{t+1}) \, \beta_{j,t+1}
\end{aligned}
$$

$$
n \text{ rows} \left\{ \begin{bmatrix} \begin{matrix} \beta_{11} \\ \beta_{21} \\ \vdots \\ \beta_{n1} \end{matrix} & \begin{matrix} \beta_{12} \\ \beta_{22} \\ \vdots \\ \beta_{n2} \end{matrix} & \begin{matrix} \cdots \\ \cdots \\ \vdots \\ \cdots \end{matrix} & \begin{matrix} \beta_{1,T-1} \\ \beta_{2,T-1} \\ \vdots \\ \beta_{n,T-1} \end{matrix} & \boxed{\begin{matrix} 1 \\ 1 \\ \vdots \\ 1 \end{matrix}} \end{bmatrix} \right.
$$

Outline

1. **Review**
2. $\boxed{\textbf{EM algorithm for HMMs}}$
3. **Multivariate Gaussian distributions**
4. **Clustering with GMMs**

Learning in HMMs



Given: one or more sequences of observations $\{o_1, o_2, \ldots, o_T\}$.
For simplicity, we'll assume just one.

Goal: estimate $\{\pi_i, a_{ij}, b_{ik}\}$ to maximize $P(o_1, o_2, \ldots, o_T)$, the likelihood of the observed data.

Assume: the cardinality n of the hidden state space is fixed.
$s_t \in \{1, 2, \ldots, n\}$

EM algorithm for HMMs

- **CPTs to re-estimate:**

$$\begin{aligned}\pi_i &= P(S_1\!=\!i)\\ a_{ij} &= P(S_{t+1}\!=\!j|S_t\!=\!i)\\ b_{ik} &= P(O_t\!=\!k|S_t\!=\!i)\end{aligned}$$

- **E-step in HMMs must compute:**

$$\underbrace{P(S_1\!=\!i|o_1,o_2,\ldots,o_T)}_{\text{special case of below } (t=1)}$$

$$P(S_{t+1}\!=\!j, S_t\!=\!i|o_1,o_2,\ldots,o_T)$$

$$P(O_t\!=\!k, S_t\!=\!i|o_1,o_2,\ldots,o_T) \;=\; I(o_t,k)\,P(S_t\!=\!i|o_1,o_2,\ldots,o_T)$$

How to efficiently compute these posteriors?

Hint: use the α and β matrices.

Computing $P(S_t = i | o_1, \ldots, o_T)$

$$P(S_t = i | o_1, \ldots, o_T) = \frac{P(S_t = i, o_1, o_2, \ldots, o_T)}{P(o_1, o_2, \ldots, o_T)}$$ **product rule**

- **Numerator**

$$\begin{aligned} P(S_t = i, o_1, o_2, \ldots, o_T) & \\ &= P(o_1, \ldots, o_t, S_t = i)\, P(o_{t+1}, \ldots, o_T | S_t = i, o_1, \ldots, o_t) && \text{product rule} \\ &= P(o_1, \ldots, o_t, S_t = i)\, P(o_{t+1}, \ldots, o_T | S_t = i) && \text{conditional independence} \\ &= \alpha_{it}\, \beta_{it} \end{aligned}$$

- **Denominator**

$$\begin{aligned} P(o_1, o_2, \ldots, o_T) &= \sum_k P(S_t = k, o_1, o_2, \ldots, o_T) && \text{marginalization} \\ &= \sum_k \alpha_{kt}\, \beta_{kt} && \text{by above} \end{aligned}$$

Note: this holds for all values of t.

Computing $P(S_t = i, S_{t+1} = j | o_1, \ldots, o_T)$

$$P(S_t = i, S_{t+1} = j | o_1, \ldots, o_T) = \frac{P(S_t = i, S_{t+1} = j, o_1, o_2, \ldots, o_T)}{\underbrace{P(o_1, o_2, \ldots, o_T)}_{\text{already computed}}} \quad \boxed{\textbf{product rule}}$$

- **Numerator**

$$\begin{aligned} & P(S_t = i, S_{t+1} = j, o_1, o_2, \ldots, o_T) \\ = & \Big[P(o_1, \ldots, o_t, S_t = i) \cdot P(S_{t+1} = j | o_1, \ldots, o_t, S_t = i) \cdot \\ & P(o_{t+1} | o_1, \ldots, o_t, S_t = i, S_{t+1} = j) \cdot \\ & P(o_{t+2}, \ldots, o_T | o_1, \ldots, o_{t+1}, S_t = i, S_{t+1} = j) \Big] \quad \boxed{\textbf{product rule}} \\ = & P(o_1, \ldots, o_t, S_t = i) \cdot P(S_{t+1} = j | S_t = i) \cdot \end{aligned}$$

Computing $P(S_t = i, S_{t+1} = j | o_1, \ldots, o_T)$

$$P(S_t = i, S_{t+1} = j | o_1, \ldots, o_T) = \frac{P(S_t = i, S_{t+1} = j, o_1, o_2, \ldots, o_T)}{\underbrace{P(o_1, o_2, \ldots, o_T)}_{\text{already computed}}} \quad \boxed{\text{product rule}}$$

- **Numerator**

$$\begin{aligned} & P(S_t = i, S_{t+1} = j, o_1, o_2, \ldots, o_T) \\ = & \Big[P(o_1, \ldots, o_t, S_t = i) \cdot P(S_{t+1} = j | o_1, \ldots, o_t, S_t = i) \cdot \\ & P(o_{t+1} | o_1, \ldots, o_t, S_t = i, S_{t+1} = j) \cdot \\ & P(o_{t+2}, \ldots, o_T | o_1, \ldots, o_{t+1}, S_t = i, S_{t+1} = j) \Big] \quad \boxed{\text{product rule}} \\ = & P(o_1, \ldots, o_t, S_t = i) \cdot P(S_{t+1} = j | S_t = i) \cdot \\ & P(o_{t+1} | S_{t+1} = j) \cdot \end{aligned}$$

Computing $P(S_t = i, S_{t+1} = j | o_1, \ldots, o_T)$

$$P(S_t = i, S_{t+1} = j | o_1, \ldots, o_T) = \frac{P(S_t = i, S_{t+1} = j, o_1, o_2, \ldots, o_T)}{\underbrace{P(o_1, o_2, \ldots, o_T)}_{\text{already computed}}} \qquad \boxed{\textbf{product rule}}$$

- **Numerator**

$$\begin{aligned} & P(S_t = i, S_{t+1} = j, o_1, o_2, \ldots, o_T) \\ = & \Big[P(o_1, \ldots, o_t, S_t = i) \cdot P(S_{t+1} = j | o_1, \ldots, o_t, S_t = i) \cdot \\ & P(o_{t+1} | o_1, \ldots, o_t, S_t = i, S_{t+1} = j) \cdot \\ & P(o_{t+2}, \ldots, o_T | o_1, \ldots, o_{t+1}, S_t = i, S_{t+1} = j) \Big] \qquad \boxed{\textbf{product rule}} \\ = & P(o_1, \ldots, o_t, S_t = i) \cdot P(S_{t+1} = j | S_t = i) \cdot \\ & P(o_{t+1} | S_{t+1} = j) \cdot P(o_{t+2}, \ldots, o_T | S_{t+1} = j) \qquad \boxed{\textbf{conditional independence}} \\ = & \alpha_{it} \, a_{ij} \, b_j(o_{t+1}) \, \beta_{j,t+1} \end{aligned}$$

Forward-backward algorithm for inference in HMMs

- **Summary of E-step:**

$$\boxed{\begin{aligned} P(S_t=i|o_1,\ldots,o_T) &= \frac{\alpha_{it}\,\beta_{it}}{\sum_j \alpha_{jt}\,\beta_{jt}} \\ P(S_t=i,S_{t+1}=j|o_1,\ldots,o_T) &= \frac{\alpha_{it}\,a_{ij}\,b_j(o_{t+1})\,\beta_{j,t+1}}{\sum_k \alpha_{kt}\,\beta_{kt}} \end{aligned}}$$

- **(Aside) HW 7.2 — use α, β matrices to compute:**

$$P(S_{t+1}=j|S_t=i,o_1,o_2,\ldots,o_T)$$
$$P(S_t=i|S_{t+1}=j,o_1,o_2,\ldots,o_T)$$
$$P(S_{t+1}=j|S_{t-1}=i,o_1,o_2,\ldots,o_T)$$
$$\vdots$$

EM algorithm for HMMs

- **CPTs to re-estimate:**

$$\begin{aligned}\pi_i &= P(S_1 = i)\\ a_{ij} &= P(S_{t+1} = j | S_t = i)\\ b_{ik} &= P(O_t = k | S_t = i)\end{aligned}$$

- **M-step updates:**

$$\begin{aligned}\pi_i &\leftarrow P(S_1 = i | o_1, o_2, \ldots, o_T)\\ a_{ij} &\leftarrow \frac{\sum_t P(S_{t+1} = j, S_t = i | o_1, o_2, \ldots, o_T)}{\sum_t P(S_t = i | o_1, o_2, \ldots, o_T)}\\ b_{ik} &\leftarrow \frac{\sum_t I(o_t, k)\, P(S_t = i | o_1, o_2, \ldots, o_T)}{\sum_t P(S_t = i | o_1, o_2, \ldots, o_T)}\end{aligned}$$

(for one sequence of observations)

Time complexity of HMM computations

T	length of observation sequence $(o_1, o_2, \ldots, o_T)$
n	cardinality of state space $s_t \in \{1, 2, \ldots, n\}$
m	cardinality of observation space $o_t \in \{1, 2, \ldots, m\}$

- **All of the following computations are $O(n^2 T)$:**

  (a) computing the likelihood $P(o_1, o_2, \ldots, o_T)$

  (b) decoding $\text{argmax}_{s_1,\ldots,s_T} P(s_1, \ldots, s_T | o_1, \ldots, o_T)$

  (c) re-estimating $\{\pi_i, a_{ij}, b_{ik}\}$ by **one update of EM**

  (d) updating beliefs $P(S_t = i | o_1, \ldots, o_t)$ **for T steps** $\boxed{\textbf{HW 7}}$

Outline

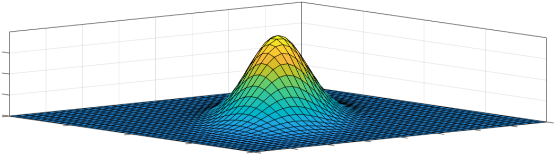
1. **Review**
2. **EM algorithm for HMMs**
3. **Multivariate Gaussian distributions**
4. **Clustering with GMMs**

Multivariate Gaussian distribution

- **Probability density function (PDF) over $\Re^d$**

$$P(\mathbf{x}) = \frac{1}{\sqrt{(2\pi)^d \det(\boldsymbol{\Sigma})}} \exp\left\{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})\right\}$$

- **PDF in $d = 2$**



- **Parameters**

$$\textbf{mean} \quad \boldsymbol{\mu} = \mathrm{E}[\mathbf{x}] = \int_{\Re^d} P(\mathbf{x})\,\mathbf{x}$$

$$\textbf{covariance} \quad \boldsymbol{\Sigma} = \mathrm{E}[(\mathbf{x}-\boldsymbol{\mu})(\mathbf{x}-\boldsymbol{\mu})^\top] = \int_{\Re^d} P(\mathbf{x})\,(\mathbf{x}-\boldsymbol{\mu})(\mathbf{x}-\boldsymbol{\mu})^\top$$

Definitions

- We say that $\mathbf{x}$ is **normally distributed** if

$$P(\mathbf{x}) = \frac{1}{\sqrt{(2\pi)^d \det(\boldsymbol{\Sigma})}} \exp\left\{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})\right\}$$

  for some $\boldsymbol{\mu} \in \Re^d$ and some **positive-definite** $\boldsymbol{\Sigma} \in \Re^{d\times d}$.

- $\boldsymbol{\Sigma}$ is **positive-definite** if $\boldsymbol{\Sigma} = \boldsymbol{\Sigma}^\top$ and $\mathbf{x}^\top \boldsymbol{\Sigma} \mathbf{x} > 0$ for all $\mathbf{x} \neq 0$. This is necessary for the above PDF to be normalizable.

- If $\mathbf{x}$ is normally distributed, we commonly write

$$\mathbf{x} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$$

  to indicate that $\mathbf{x}$ has mean $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$.

Mathematical properties

Normality is preserved by many important operations:

1. **Linear transformations**

   Suppose $\mathbf{x} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$. Then for any linear transformation $\mathbf{A}$, we have $\mathbf{A}\mathbf{x} \sim \mathcal{N}(\mathbf{A}\boldsymbol{\mu}, \mathbf{A}\boldsymbol{\Sigma}\mathbf{A}^\top)$.

2. **Linear combinations**

   Suppose $\mathbf{x}_i \sim \mathcal{N}(\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)$ are independent random variables in $\Re^d$. Then any linear combination $\sum_i \alpha_i \mathbf{x}_i$ is normally distributed in $\Re^d$.

3. **Marginalization and conditionalization**

   Suppose $\mathbf{x} = (x_1, x_2, \ldots, x_d)$, and $P(\mathbf{x})$ is multivariate Gaussian.

   Then so are all the marginal distributions $\{P(x_1), P(x_1, x_2), \ldots\}$ and conditional distributions $\{P(x_1|x_2), P(x_1|x_2, x_3), \ldots\}$.

Maximum likelihood (ML) estimation

- **Learning from data**

  Let $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_T\}$ be *i.i.d.* examples in $\Re^d$.
  Assume $\mathbf{x}$ is normally distributed: how to estimate $(\boldsymbol{\mu}, \boldsymbol{\Sigma})$?

- **ML estimates**

  Choose $(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ to maximize the log-likelihood $\mathcal{L} = \sum_{t=1}^{T} \log P(\mathbf{x}_t)$.
  Setting $\frac{\partial \mathcal{L}}{\partial \mu_i} = \frac{\partial \mathcal{L}}{\partial \Sigma_{ij}} = 0$, we find:

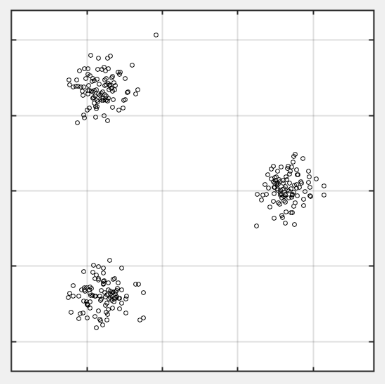
$$\boldsymbol{\mu}_{\mathrm{ML}} = \frac{1}{T}\sum_{t=1}^{T} \mathbf{x}_t \qquad \boxed{\textbf{sample mean}}$$

$$\boldsymbol{\Sigma}_{\mathrm{ML}} = \frac{1}{T}\sum_{t=1}^{T} (\mathbf{x}_t - \boldsymbol{\mu}_{\mathrm{ML}})(\mathbf{x}_t - \boldsymbol{\mu}_{\mathrm{ML}})^{\top} \qquad \boxed{\textbf{sample covariance matrix}}$$

Outline

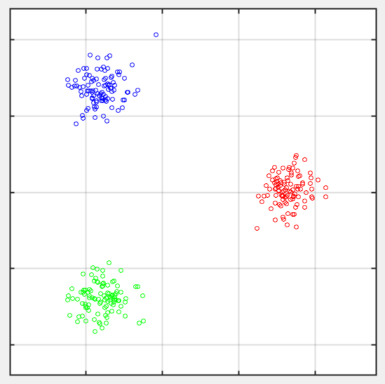
1. Review
2. EM algorithm for HMMs
3. Multivariate Gaussian distributions
4. Clustering with GMMs

Clustering



How to cluster these points in the xy-plane?

Clustering

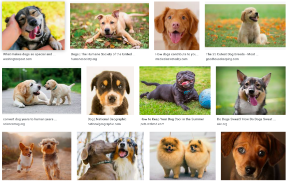


Visually it's obvious.

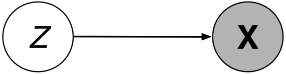
But what if the points are not in the xy-plane?

Clustering — more generally

What if the points are documents, represented as vectors of word counts?



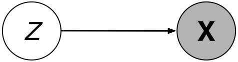
What if the points are images, represented as vectors of pixels?



More generally:

how to infer labels $z \in \{1, 2, \ldots, k\}$
from inputs $\mathbf{x} \in \mathbb{R}^d$
without any labeled examples?

Gaussian mixture model (GMM)



- **Random variables**

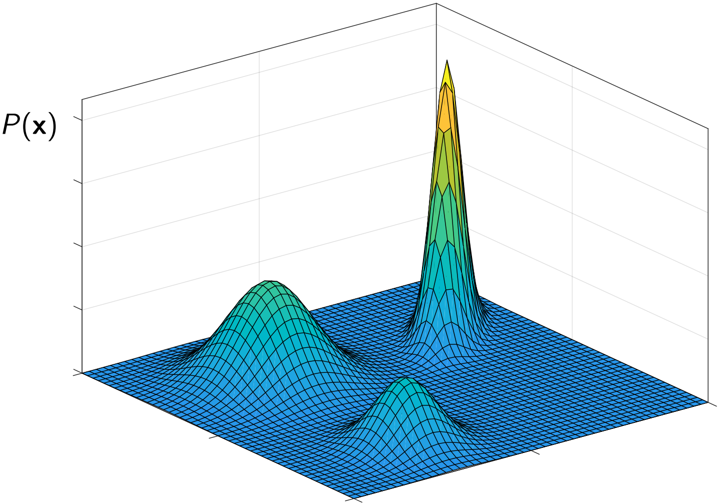
$\mathbf{x} \in \Re^d$	real-valued vector	(observed)
$z \in \{1, 2, \ldots, k\}$	cluster label	(hidden)

- **Conditional probability tables (CPTs)**

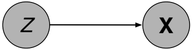
$P(z=i)$	fraction of data in i^{th} cluster
$P(\mathbf{x}\|z=i)$	multivariate Gaussian $\mathbf{x} \sim \mathcal{N}(\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)$

Each cluster has its own mean and covariance matrix!

GMM visualized ($d = 2$, $k = 3$)



Learning from complete data — EASY



data $\{(\mathbf{x}_t, z_t)\}_{t=1}^{T}$

- **Shorthand**

  Let $T_i = \sum_{t=1}^{T} I(z_t, i)$ count the number of points in the i^{th} cluster.

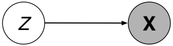
- **ML estimates**

$$P(Z=i) = \frac{1}{T}\sum_{t=1}^{T} I(z_t, i) = \frac{T_i}{T}$$

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

$$\boldsymbol{\Sigma}_i = \frac{1}{T_i}\sum_{t=1}^{T} I(z_t, i)\, (\mathbf{x}_t - \boldsymbol{\mu}_i)(\mathbf{x}_t - \boldsymbol{\mu}_i)^{\top}$$

Learning from incomplete data — EM



data $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_T\}$

- **E-step: compute posterior probabilities**

$$\begin{aligned} P(Z\!=\!i|\mathbf{x}_t) &= \frac{P(\mathbf{x}_t|Z\!=\!i)\,P(Z\!=\!i)}{P(\mathbf{x}_t)} && \boxed{\textbf{Bayes rule}} \\ &= \frac{P(\mathbf{x}_t|Z\!=\!i)\,P(Z\!=\!i)}{\sum_j P(\mathbf{x}_t|Z\!=\!j)\,P(Z\!=\!j)} && \boxed{\textbf{normalization}} \end{aligned}$$

- **M-step: update model parameters**

$$\begin{aligned} P(Z\!=\!i) &\leftarrow \frac{1}{T}\sum_t P(Z\!=\!i|\mathbf{x}_t) \\ \boldsymbol{\mu}_i &\leftarrow ? \\ \boldsymbol{\Sigma}_i &\leftarrow ? \end{aligned}$$

We will not rigorously derive the latter updates.
But they are exactly what you'd expect by analogy ...

EM updates for GMMs

- **ML estimates for complete data**

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

$$\boldsymbol{\Sigma}_i = \frac{\sum_{t=1}^{T} I(z_t, i)\, (\mathbf{x}_t - \boldsymbol{\mu}_i)(\mathbf{x}_t - \boldsymbol{\mu}_i)^\top}{\sum_{t=1}^{T} I(z_t, i)}$$

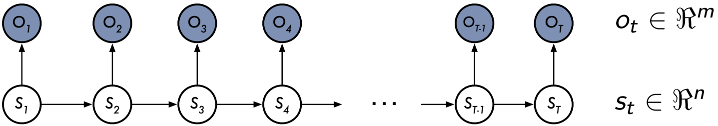
- **EM updates for incomplete data**

$$\boldsymbol{\mu}_i \leftarrow \frac{\sum_{t=1}^{T} P(Z=i|\mathbf{x}_t)\, \mathbf{x}_t}{\sum_{t=1}^{T} P(Z=i|\mathbf{x}_t)}$$

$$\boldsymbol{\Sigma}_i \leftarrow \frac{\sum_{t=1}^{T} P(Z=i|\mathbf{x}_t)\, (\mathbf{x}_t - \boldsymbol{\mu}_i)(\mathbf{x}_t - \boldsymbol{\mu}_i)^\top}{\sum_{t=1}^{T} P(Z=i|\mathbf{x}_t)}$$

Next lecture ...

- **Linear dynamical systems**



- **Reinforcement learning — introduction**