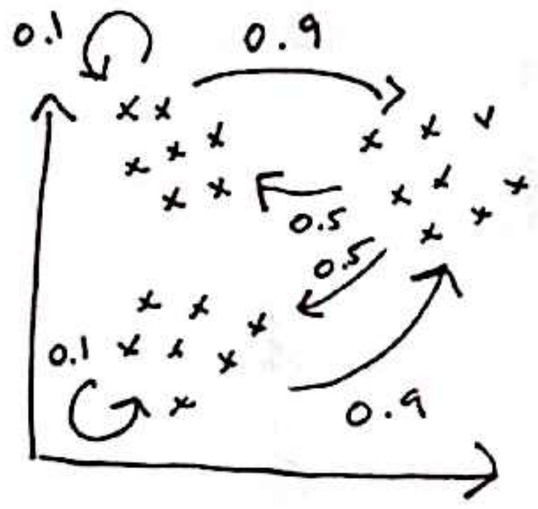
Sequential models

- in many settings data are sequential
  - language data
  - time indexed data
  - DNA
- we will study 2 sequential models — HMM & Kalman filter (further make EM concrete...)
- the HMM is a sequential generalization of the mixture model
- idea — the chosen mixture component at time t depends on the component at time t-1.

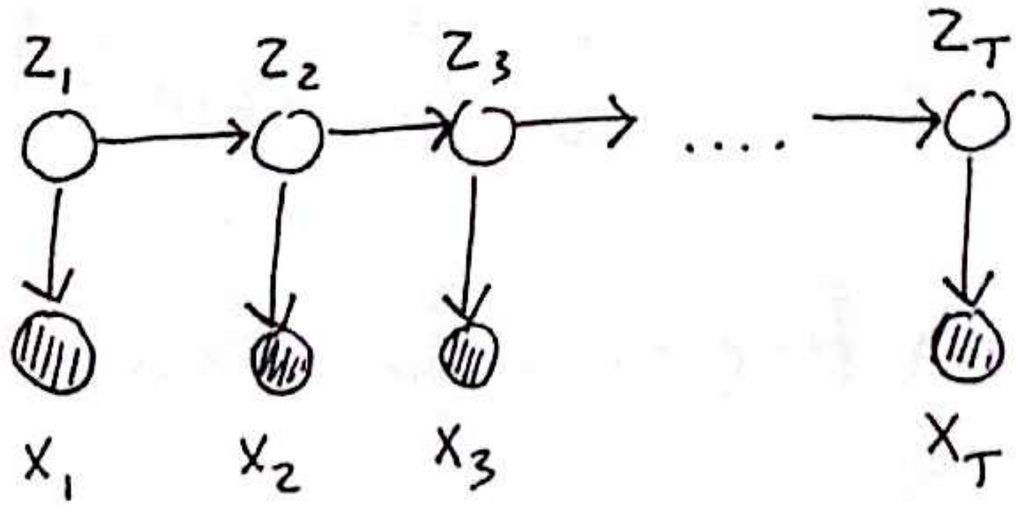
(contrast to a mixture, where the components are assumed $\perp\!\!\!\perp$.)

- 

The HMM models transitions b/w the components.

[show data generated from this picture.]

HMM modeling assumptions



z_t: discrete (K values)

x_t: anything (e.g., $\mathbb{R}^2$)

$$p(z_{1:T}, x_{1:T}) = p(z_1)\, p(x_1 \mid z_1) \prod_{t=2}^{T} p(z_t \mid z_{t-1})\, p(x_t \mid z_t)$$

(or group the x's all together...)

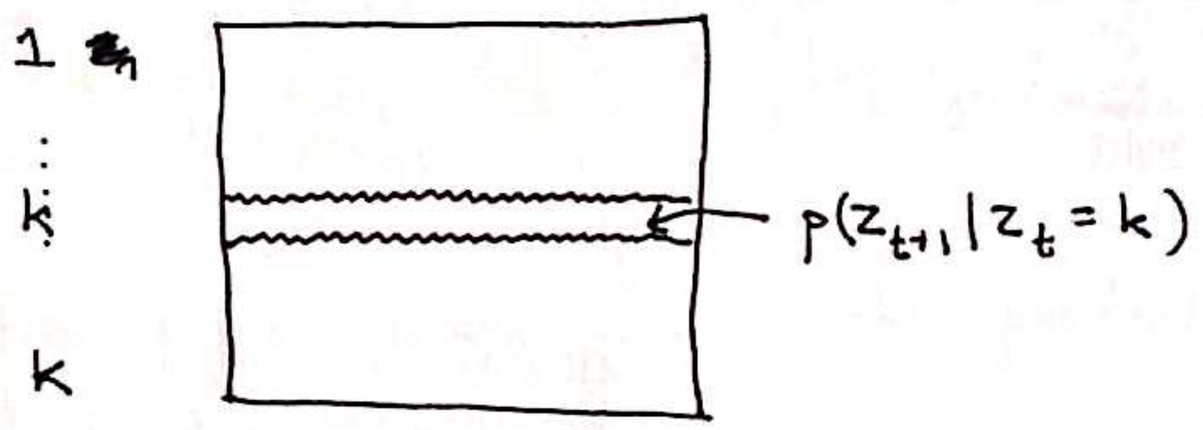
Parameters are —

- Data generating distributions θ_k (also called emission probs)

$p(x_t|z_t) = \prod_{k=1}^{K} z_t^k p(x_t|\theta_k)$ ←

  - As for mixtures, this can be anything.
  - To be concrete, make θ_k parameters to 2D Gaussians (note - MN in genetics)

- Transition matrix A
  - Probability of the next state given the current state.



- Recall that z_t is an indicator vector
- Thus,

$$p(z_t|z_{t-1}) = \prod_{k=1}^{K}\prod_{j=1}^{K} a_{jk}^{z_{t-1}^j z_t^k}$$

- Probability of the first state (to start things off) π

$$p(z_1) = \prod_{k=1}^{K} \pi_k^{z_1^k}$$

- Show a state transition diagram and emphasize it is not a G.M.
- Applications
  - speech recognition
  - handwriting recognition
  - DNA analysis
  - others...
  - part of speech tagging

Fitting an HMM

EM algorithm.

① Write down the expected complete log likelihood

$$\mathbb{E}\left[\log p(z_{1:T}, x_{1:T})\right] = \mathbb{E}\left[\log p(z_1) \prod_{t=2}^{T} p(z_t|z_{t-1}) \prod_{t=1}^{T} p(x_t|z_t)\right]$$

$$= \mathbb{E}\left[\log \prod_{k=1}^{K} \pi_k^{z_1^k} \prod_{t=2}^{T} \prod_{j=1}^{K} \prod_{k=1}^{K} a_{jk}^{z_{t-1}^j z_t^k} \prod_{t=1}^{T} \prod_{k=1}^{K} z_t^k \, p(x_t|\theta_k)\right]$$

$$= \sum_{k=1}^{K} \mathbb{E}[z_1^k] \underbrace{\log \pi_k}_{\gamma_k} + \sum_{t=2}^{T} \sum_{j=1}^{K} \sum_{k=1}^{K} \mathbb{E}\left[z_{t-1}^j z_t^k\right] \log a_{jk} + \sum_{t=1}^{T} \mathbb{E}[z_t^k] \log p(x_t|\theta_k)$$

All expectations are taken w/r/t the posterior. $p(z_{1:T} \mid x_{1:T})$

And, since z_t is an indicator vector —

$$\triangleq \gamma(z_t^k)$$

$$\mathbb{E}_x[z_1^k] = p(z_1 = k \mid x_{1:T}), \quad \mathbb{E}_x[z_t^k] = p(z_t = k \mid x_{1:T})$$

$$\mathbb{E}_x[z_{t-1}^j z_t^k] = p(z_{t-1} = j, z_t = k \mid x_{1:T}) \quad \triangleq \xi(z_{t-1}^j, z_t^k)$$

② M-step Note as usual that the log complete likelihood decomposes.

$$\pi_k = \mathbb{E}[z_1^k] \Big/ \sum_j \mathbb{E}[z_1^j]$$ (normalized expected counts)

$$a_{jk} = \sum_{t=2}^{T} \mathbb{E}\left[z_{t-1}^j z_t^k\right] \Big/ \sum_{t=2}^{T} \sum_{\ell=1}^{K} \mathbb{E}\left[z_{t-1}^j z_t^\ell\right]$$

note: this stays at j.

θ_k: weighted maximum likelihood, as for mixture models.

Gaussian —

$$\mu_k = \sum_{t=2}^{T} \mathbb{E}[z_t^k]\, x_t \Big/ \sum_{t=2}^{T} \mathbb{E}[z_t^k]$$

$$\Sigma_k = \sum_{t=2}^{T} \mathbb{E}[z_t^k]\, (x_t - \mu_k)(x_t - \mu_k)^T \Big/ \sum_{t=2}^{T} \mathbb{E}[z_t^k]$$

Discrete (e.g., biology) —

$$p(x_t \mid \theta_k) = \prod_{i=1}^{V} \theta_{ki}^{\,x_t^i}$$

$$\theta_{ki} = \sum_{t=2}^{T} \mathbb{E}[z_t^k]\, x_t^i \Big/ \sum_{t=2}^{T} \mathbb{E}[z_t^k]$$

note: this is how to do a mixture of $\mathbb{R}^d$ Gaussians

Others ... Poisson, etc.

<u>E-step</u> — also used for <u>prediction</u>.

This is the interesting part

$$\mathbb{E}[z_t \mid x_{1:T}] = p(z_t \mid x_{1:T})$$

$$= p(z_t, x_{1:T}) \,/\, p(x_{1:T})$$

$$= p(x_{1:t}, z_t)\, p(x_{t+1:T} \mid z_t) \,/\, p(x_{1:T})$$

We will compute these terms recursively —

$$\alpha(z_t) \triangleq p(x_{1:t}, z_t) \qquad [K\text{-vector}]$$

$$\beta(z_t) \triangleq p(x_{t+1:T} \mid z_t) \qquad [K\text{-vector}]$$

$$= \alpha(z_t)\, \beta(z_t) \,/\, p(x_{1:T})$$

- Also lets us compute:

$$\mathbb{E}[z_{t-1} z_t \mid x_{1:T}] = p(z_{t-1}, z_t \mid x_{1:T})$$
$$= p(x_{1:T}, z_{t-1}, z_t) \,/\, p(x_{1:T})$$
$$= p(x_{1:t}, z_{t-1})\, p(z_t \mid z_{t-1})\, p(x_t \mid z_t)\, p(x_{t+1:T} \mid z_t) \,/\, p(x_{1:T})$$
$$= \alpha(z_{t-1})\, p(z_t \mid z_{t-1})\, p(x_t \mid z_t)\, \beta(z_t) \,/\, p(x_{1:T})$$

(Note – recursively applying this gives us any subsequence. But, we only need pairs for EM.)

- The likelihood is easy from these quantities, from ⊛

$$\sum_{z_t} \alpha(z_t)\, \beta(z_t) / p(x_{1:T}) = 1 \;\rightarrow\; p(x_{1:T}) = \sum_{z_t} \alpha(z_t)\, \beta(z_t)$$

- We are left to compute α, β: Recursive algorithm.
  Sometimes called forward-backward, α-β, is an instance of propagation.

$$\alpha(z_1) = p(z_1, x_1)$$
$$= \pi_{z_1}\, p(x_1 \mid z_1) \quad \text{slight abuse of notation.}$$

$$\alpha(z_{t+1}) = p(x_{1:t+1}, z_{t+1})$$
$$= p(x_{1:t+1} \mid z_{t+1})\, p(z_{t+1})$$
$$= p(x_{1:t} \mid z_{t+1})\, p(x_{t+1} \mid z_{t+1})\, p(z_{t+1}) \qquad [x_{t+1} \perp\!\!\!\perp x_{1:t} \mid z_{t+1}]$$
$$= \sum_{z_t} p(x_{1:t}, z_{t+1})\, p(x_{t+1} \mid z_{t+1})$$
$$= \sum_{z_t} p(x_{1:t}, z_t, z_{t+1})\, p(x_{t+1} \mid z_{t+1})$$
$$= \sum_{z_t} p(x_{1:t}, z_{t+1} \mid z_t)\, p(z_t)\, p(x_{t+1} \mid z_{t+1})$$

$$= \sum_{z_t} p(x_{1:t} | z_t)\, p(z_{t+1} | z_t)\, p(z_t)\, p(x_{t+1} | z_{t+1}) \qquad [z_{t+1} \perp\!\!\!\perp x_{1:t} \mid z_t]$$

$$= \sum_{z_t} p(x_{1:t}, z_t)\, p(z_{t+1} | z_t)\, p(x_{t+1} | z_{t+1})$$

$\alpha(z_t)$ $\quad$ $a_{z_t, z_{t+1}}$ $\quad$ emission prob.

Complexity for each step is $O(K^2)$:
for each value of z_{t+1}, we sum over K elements.
Complexity for the recursion is $O(T K^2)$

Turn to the β recursion, i.e., the backward step.

$$\beta(z_t) = p(x_{t+1:T} | z_t)$$

$$= \sum_{z_{t+1}} p(x_{t+1:T}, z_{t+1} | z_t)$$

$$= \sum_{z_{t+1}} p(x_{t+1:T} | z_{t+1}, z_t)\, p(z_{t+1} | z_t) \qquad [\text{chain rule}]$$

$$= \sum_{z_{t+1}} p(x_{t+2:T} | z_{t+1})\, p(x_{t+1} | z_{t+1})\, p(z_{t+1} | z_t)$$

$\beta(z_{t+1})$ $\quad$ emission $\quad$ $a_{z_t, z_{t+1}}$ $\quad$ $\begin{bmatrix} x_{t+1} \perp\!\!\!\perp x_{t+2:T} \mid z_{t+1} \\ x_{t+1:T} \perp\!\!\!\perp z_t \mid z_{t+1} \end{bmatrix}$

Setting $\beta(z_T) \triangleq 1$ makes the recursion work.
(note $\beta(z_T)$ is meaningless because x_{T+1} does not exist.)

or $\beta(z_{T-1}) = p(x_T | z_{T-1})$

$$= \sum_{z_T} p(x_T | z_T)\, p(z_T | z_{T-1})$$

Can also compute the predictive dist —

$$p(x_{T+1} \mid x_{1:T}) = \sum_{z_{T+1}} p(x_{T+1} \mid z_{T+1})\, p(z_{T+1} \mid x_{1:T})$$

Many variants of HMMs —

[redacted]

Hierarchical HMMs — state transitions at diff. levels

Generalized HMMs — random # of emissions per time point

Shadower — HMM respecting a phylogenetic tree

Factorial HMMs — $\binom{1}{0}$ Factorial states.

"Infinite HMMs" — K is unbounded