# STAT 339
# Hidden Markov Models III

21 April 2017

Bayesian Estimation / Model Averaging

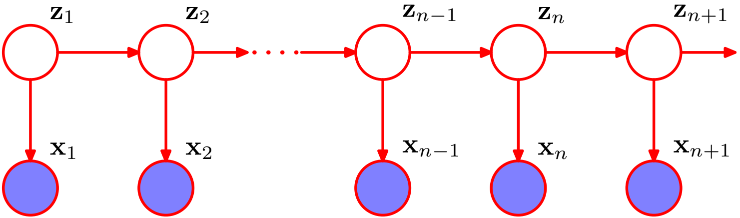
# Outline

# A Generative Model

We can construct a generative model of the joint distribution of the $\mathbf{z}$ and the $\mathbf{x}$

$$p(\mathbf{z}, \mathbf{x}) = \prod_{n=1}^{N} p(z_n \mid z_{n-1}) p(x_n \mid z_n)$$

This corresponds to the graphical model below



# Outline

# Inference in HMMs

Given full specification of the component distributions (transition and emission probabilities), we might want to

1. Find the marginal distribution of a particular state $p(z_{n'})$ or observation $p(x_{n'})$ (e.g., predict the future or recover the past) Forward-Backward Algorithm
2. Evaluate marginal likelihood $p(\mathbf{x})$ of some data (e.g., for model comparison) Forward Algorithm.
3. Find the most likely hidden sequence given data: $\operatorname{argmax}_{\mathbf{z}} p(\mathbf{z} \mid \mathbf{x})$ Viterbi Algorithm (we are skipping)
4. Get samples from $p(\mathbf{z} \mid \mathbf{x})$ today

# Learning HMMs

n If we don't know the transition and emission probabilities, we might want to

1. Find MLE transition matrix and emission parameters

$$\underset{\mathbf{A},\boldsymbol{\theta}}{\operatorname{argmax}} \prod_{n=1}^{N} p(z_n \mid z_{n-1}, \mathbf{A}) p(x_n \mid z_n, \boldsymbol{\theta})$$

   where the element $\mathbf{A}_{k,k'}$ encodes $p(z_n = k', \mid z_{n-1} = k)$, and $\boldsymbol{\theta}$ is a set of parameters of the "emission distributions" for each state. EM Algorithm

2. Do some model averaging using a posterior distribution over $\mathbf{A}$ and $\boldsymbol{\theta}$; e.g., by getting samples

$$\mathbf{A}^{(s)}, \boldsymbol{\theta}^{(s)} \sim p(\mathbf{A}, \boldsymbol{\theta} \mid \mathbf{x})$$

   MCMC (today)

# Outline

# Outline

# Summary: Forward-Backward Algorithm

We have defined the following shorthand:

$\mathbf{A}$ : transition matrix: $a_{kk'} := p(z_n = k' \mid z_{n-1} = k)$

$\mathbf{B}^*$ : "observed" likelihood matrix: $b^*_{nk} := p(x_n \mid z_n = k)$

$\mathbf{m}_n$ : "cumulative" prior / "forward" message:

$$m_{nk} := p(z_n = k, x_{1:n})$$

$\mathbf{r}_n$ : "residual" likelihood / "backward" message:

$$r_{nk} := p(x_{n+1:N} \mid z_n = k)$$

We have also derived the following recursion formulas:

$$\mathbf{m}_n = \mathbf{A}^\top \mathbf{m}_{n-1} \odot \mathbf{b}^*_n, \qquad m_{1k} = p(z_1 = k)p(x_1 \mid z_1 = k)$$
$$\mathbf{r}_n = \mathbf{A} \cdot (\mathbf{b}^*_{n+1} \odot \mathbf{r}_{n+1}), \quad \mathbf{r}_N = \mathbf{1}$$

# Summary: Forward-Backward Algorithm

We have defined the following shorthand:

$\mathbf{A}$ : transition matrix: $a_{kk'} := p(z_n = k' \mid z_{n-1} = k)$

$\mathbf{B}^*$ : "observed" likelihood matrix: $b^*_{nk} := p(x_n \mid z_n = k)$

$\mathbf{m}_n$ : "cumulative" prior / "forward" message:

$$m_{nk} := p(z_n = k, x_{1:n})$$

$\mathbf{r}_n$ : "residual" likelihood / "backward" message:

$$r_{nk} := p(x_{n+1:N} \mid z_n = k)$$

Using these we can compute marginals for any $n$

$$p(z_n \mid x_{1:N}) = \frac{p(z_n, x_{1:n})p(x_{n+1:N} \mid z_n)}{p(x_{1:N})} = \frac{\mathbf{m}_n \odot \mathbf{r}_n}{\mathbf{m}_n^\top \mathbf{r}_n}$$

# Summary: Forward-Backward Algorithm

We have defined the following shorthand:

$\mathbf{A}$ : transition matrix: $a_{kk'} := p(z_n = k' \mid z_{n-1} = k)$

$\mathbf{B}^*$ : "observed" likelihood matrix: $b^*_{nk} := p(x_n \mid z_n = k)$

$\mathbf{m}_n$ : "cumulative" prior / "forward" message:

$$m_{nk} := p(z_n = k, x_{1:n})$$

$\mathbf{r}_n$ : "residual" likelihood / "backward" message:

$$r_{nk} := p(x_{n+1:N} \mid z_n = k)$$

As part of this calculation, we get the overall marginal likelihood of the model for free:

$$p(x_{1:N}) = \sum_k p(z_n = k, x_{1:N}) = \mathbf{m}_N^\top \mathbf{1}$$

# Outline

# Outline

# Maximum Likelihood Estimation

- We can parameterize the model using

$$\pi_{kk'} := p(z_n = k' \mid z_{n-1} = k, \boldsymbol{\pi})$$
$$f(\mathbf{x} \mid \theta_k) = p(\mathbf{x} \mid z = k, \boldsymbol{\theta})$$

- Then we have a likelihood function for $\boldsymbol{\theta}$ and $\boldsymbol{\pi}$ given $\mathbf{z}$ and data, $\mathbf{x}$

$$\begin{aligned} p(\mathbf{z}, \mathbf{x} \mid \boldsymbol{\pi}, \boldsymbol{\theta}) &= \prod_{n=1}^{N} p(z_n \mid z_{n-1}) p(\mathbf{x}_n \mid z_n) \\ &= \prod_{n=1}^{N} \pi_{z_{n-1} z_n} f_{z_n}(\mathbf{x}_n \mid \theta_k) \\ &= \left( \prod_{k=1}^{K} \prod_{k'=1}^{K} \pi_{kk'}^{N_{kk'}} \right) \left( \prod_{k=1}^{K} \prod_{n: z_n = k} f_k(\mathbf{x}_n \mid \theta_k) \right) \end{aligned}$$

where $N_{kk'}$ is the number of transitions from state $k'$ to state $k'$ in $\mathbf{z}$

# Max. Likelihood Estimation

- Then we have a likelihood function for $\boldsymbol{\theta}$ and $\boldsymbol{\pi}$ given $\mathbf{z}$ and data, $\mathbf{x}$

$$\begin{aligned} p(\mathbf{z}, \mathbf{x} \mid \boldsymbol{\pi}, \boldsymbol{\theta}) &= \prod_{n=1}^{N} p(z_n \mid z_{n-1}) p(\mathbf{x}_n \mid z_n) \\ &= \prod_{n=1}^{N} \pi_{z_{n-1} z_n} f_{z_n}(\mathbf{x}_n \mid \theta_k) \\ &= \left( \prod_{k=1}^{K} \prod_{k'=1}^{K} \pi_{kk'}^{N_{kk'}} \right) \left( \prod_{k=1}^{K} \prod_{n: z_n = k} f_k(\mathbf{x}_n \mid \theta_k) \right) \end{aligned}$$

  where $N_{kk'}$ is the number of transitions from state $k'$ to state $k'$ in $\mathbf{z}$
- Factorizes into a piece with only $\boldsymbol{\pi}$, and pieces with only one $\theta_k$ each!
- Except this assumes we have $\mathbf{z}$, which we don't.

# EM Returns!

- Fortunately, if we have a current guess about $\boldsymbol{\pi}$ and $\boldsymbol{\theta}$, then we can compute

$$p(z_n = k \mid \mathbf{x}_{1:N}) \text{ for each } k$$

- Then simply assign each data point to *every* state, with weight

$$q_{nk} := p(z_n = k \mid \mathbf{x}_{1:N})$$

- We can compute these with forward-backward algorithm.

# Quantum transitions

- To estimate $\boldsymbol{\pi}$, need weights on possible transitions from $n-1$ to $n$ (for each $(k, k')$ pair)
- We want these weights to be

$$\xi_{nkk'} := p(z_{n-1} = k, z_n = k' \mid \mathbf{x}_{1:N})$$

- We can write

$$\xi_{nz_{n-1}z_n} = \frac{p(z_{n-1}, \mathbf{x}_{1:n-1})p(z_n \mid z_{n-1})p(x_n \mid z_n)p(x_{n+1:N} \mid z_n)}{p(\mathbf{x}_{1:N})}$$

$$\xi_{nkk'} = \frac{m_{n-1,k}a_{kk'}b^*_{nk'}r_{nk'}}{\mathbf{m}_N^\top \mathbf{1}}$$

# Outline

# Summary: EM for HMMs

We have developed the EM algorithm to do MLE of the HMM transition and emission parameters.

1. E-step: Execute forward-backward to compute the forward and backward messages, $\mathbf{m}_1, \dots, \mathbf{m}_N$ and $\mathbf{r}_N, \dots, \mathbf{r}_1$, , and use them to compute weights

$$\mathbf{q}_n := p(z_n \mid \mathbf{x}_{1:N}) = \frac{\mathbf{m}_n \odot \mathbf{r}_n}{\mathbf{m}_n^\top \mathbf{r}_n}$$

$$\xi_{nkk'} := p(z_{n-1} = k, z_n = k' \mid \mathbf{x}_{1:N}) = \frac{m_{n-1,k} a_{kk'} b^*_{nk'} r_{nk}}{\mathbf{m}_N^\top \mathbf{1}}$$

$$\tilde{N}_{kk'} := \sum_n \xi_{nkk'}$$

2. M-step: Maximize the "quantum" likelihood w.r.t $\boldsymbol{\pi}$ and $\boldsymbol{\theta}$

$$\left( \prod_{k=1}^K \prod_{k'=1}^K \pi_{kk'}^{\tilde{N}_{kk'}} \right) \left( \prod_{k=1}^K \prod_n f_k(\mathbf{x}_n \mid \theta_k)^{q_{nk}} \right)$$

# Outline

# Outline

# Maintaining Uncertainty

- As we've seen, MLE often does poorly unless we have a lot of data
- In particular if $K$ is large compared to $N$, then we have $K^2$ parameters in $\boldsymbol{\pi}$ and some multiple of $K$ in $\boldsymbol{\theta}$ (where the multiple depends on complexity of each $f_k(\mathbf{x} \mid \theta_k)$ distribution)
- May not have too much precision to estimate $\boldsymbol{\pi}$ and $\boldsymbol{\theta}$.
- Also we really only have a local maximum.

# Things we might want to do

- Probabilistically "classify" case $n$ by computing

$$p(z_n \mid \mathbf{x}_{1:N}) = \int p(z_n \mid \mathbf{x}_{1:N}, \boldsymbol{\pi}, \boldsymbol{\theta}) p(\boldsymbol{\pi}, \boldsymbol{\theta} \mid \mathbf{x}_{1:N}) \, d\boldsymbol{\pi} d\boldsymbol{\theta}$$

  i.e., averaging over possible parameters
- Evaluate the "marginal marginal" likelihood

$$p(\mathbf{x}_{1:N}) = \int p(\mathbf{x}_{1:N} \mid \boldsymbol{\pi}, \boldsymbol{\theta}) p(\boldsymbol{\pi}, \boldsymbol{\theta} \mid \mathbf{x}_{1:N}) \, d\boldsymbol{\pi} d\boldsymbol{\theta}$$

  e.g., to compare different models or choices of $K$
- Predict/sample future observations according to

$$p(\mathbf{x}_{N+1:N+M}) = \int p(\mathbf{x}_{N+1:N+M} \mid \boldsymbol{\pi}, \boldsymbol{\theta}) p(\boldsymbol{\pi}, \boldsymbol{\theta} \mid \mathbf{x}_{1:N}) \, d\boldsymbol{\pi} d\boldsymbol{\theta}$$

# Expectations w.r.t. the posterior

- All of these are of the form

$$\mathbb{E}_{p(\boldsymbol{\pi},\boldsymbol{\theta} \mid \mathbf{x}_{1:N})}\{f(\boldsymbol{\pi},\boldsymbol{\theta})\}$$

for different functions of $\boldsymbol{\theta}$ and $\boldsymbol{\pi}$

- We can approximate each of these using

$$\mathbb{E}_{p(\boldsymbol{\pi},\boldsymbol{\theta} \mid \mathbf{x}_{1:N})}\{f(\boldsymbol{\pi},\boldsymbol{\theta})\} \approx \frac{1}{S}\sum_{s=1}^{S} f(\boldsymbol{\pi}^{(s)},\boldsymbol{\theta}^{(s)})$$

if we can draw $\boldsymbol{\pi}^{(s)},\boldsymbol{\theta}^{(s)}$ pairs from the posterior

$$\boldsymbol{\pi}^{(s)},\boldsymbol{\theta}^{(s)} \sim p(\boldsymbol{\pi},\boldsymbol{\theta} \mid \mathbf{x}_{1:N})$$

# Outline

# EM vs. Gibbs Sampling

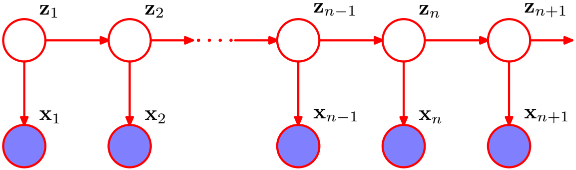
The EM algorithm (in this context) involves, iteratively

1. Computing an expectation over state assignments, $\mathbf{z}$ (using the posterior, conditioned on parameter values, $\boldsymbol{\pi}$ and $\boldsymbol{\theta}$)
2. Arg-Maximizing parameter values $\boldsymbol{\pi}$ and $\boldsymbol{\theta}$ (using the likelihood/posterior conditioned on expected state assignments, $\mathbf{z}$)

Gibbs sampling (in this context) involves, iteratively

1. Sampling state assignments $\mathbf{z}$ (using the posterior, conditioned on parameter values, $\boldsymbol{\pi}$ and $\boldsymbol{\theta}$)
2. Sampling parameter values $\boldsymbol{\pi}$ and $\boldsymbol{\theta}$ (using the posterior, conditioned on state assignments, $\mathbf{z}$)

# Gibbs Steps: Sampling Parameters



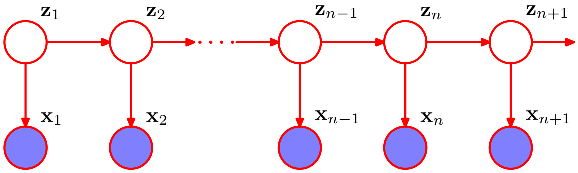

- If we have a current guess for $\mathbf{z}$, conditioning on it renders all the $\mathbf{x}_n$ mutually independent!
- So sampling $\boldsymbol{\theta}$ is completely identical to the (non-dynamic) mixture model, since the conditional likelihood is

$$p(\mathbf{x}_{1:N} \mid \mathbf{z}, \boldsymbol{\pi}, \boldsymbol{\theta}) = \prod_{n=1}^{N} f_{z_n}(\mathbf{x}_n \mid \theta_{z_n})$$

for example if the emission model is Normal,

$$p(\mathbf{x}_{1:N} \mid \mathbf{z}, \boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \prod_{n=1}^{N} \mathcal{N}(\mathbf{x}_n \mid \boldsymbol{\mu}_{z_n}, \boldsymbol{\Sigma}_{z_n})$$

# Gibbs Steps: Sampling Parameters




Provided the $\theta_k$ are independent of each other and of $\boldsymbol{\pi}$ in the prior, they are also independent in the conditional posterior, and we have

$$p(\theta_k \mid \mathbf{z}, \mathbf{x}_{1:N}) \propto p(\theta_k) \prod_{n: z_n = k} f_k(\mathbf{x}_n \mid \theta_k)$$

Often we would use a conjugate prior for $f$, so this yields a distribution with a known form which is easy to sample from (e.g., Normal-Inverse Wishart, or Dirichlet)

# Gibbs Steps: Sampling Parameters

- Sampling $\boldsymbol{\pi}$ is a bit different from the static mixture model, since the mixing weights depend on local context, but this doesn't change much.
- Conditioning on $\mathbf{z}$ we have the counts

$$N_{kk'} = \left|\{n : z_{n-1} = k \text{ and } z_n = k'\}\right|, k, k' = 1, \ldots, K$$

- If we place independent symmetric $\mathrm{Dir}(\alpha\mathbf{1})$ priors on each row of $\boldsymbol{\pi}$ (let $\boldsymbol{\pi}_k$ be the $k$th row), then

$$\boldsymbol{\pi}_k \mid \mathbf{z} \sim \mathrm{Dir}(\alpha + N_{k1}, \ldots, \alpha + N_{kK})$$

independent of all other $k$ and of $\boldsymbol{\theta}$.

# Gibbs Steps: Sampling Hidden States

- The other half of the algorithm is sampling $\mathbf{z}$, conditioned on current states of $\boldsymbol{\pi}$ and $\boldsymbol{\theta}$.
- That is, want to sample from

$$p(\mathbf{z} \mid \boldsymbol{\pi}, \boldsymbol{\theta}, \mathbf{x}_{1:N})$$

- *Evaluating* the joint probability, $p(\mathbf{z}, \mathbf{x} \mid \boldsymbol{\pi}, \boldsymbol{\theta})$ for a particular $\mathbf{z}$ is easy:

$$p(\mathbf{z}, \mathbf{x} \mid \boldsymbol{\pi}, \boldsymbol{\theta}) = \prod_{n=1}^{N} \pi_{z_{n-1} z_n} f_{z_n}(\mathbf{x}_n \mid \theta_{z_n})$$

- But there are $K^N$ possible sequences for $\mathbf{z}$ to take; we don't want to enumerate all of these probabilities.

# Forward Filtering - Backward Sampling

- We can, however, *sample* from this distribution by factoring it using the chain rule (and conditional independence).
- Omitting conditioning on $\boldsymbol{\pi}$ and $\boldsymbol{\theta}$ for easier reading,

$$p(\mathbf{z} \mid \mathbf{x}) = p(z_1 \mid \mathbf{x}_{1:N}) \prod_{n=2}^{N} p(z_n \mid z_{n-1}, \mathbf{x}_{1:N})$$

- However, it turns out it is more efficient to factor the other direction

$$p(\mathbf{z} \mid \mathbf{x}) = p(z_N \mid \mathbf{x}_{1:N}) \prod_{n=N-1}^{1} p(z_n \mid z_{n+1}, \mathbf{x}_{1:N})$$

- Why? Because we can compute $p(z_N \mid \mathbf{x}_{1:N})$ using just the forward algorithm. Computing $p(z_1 \mid \mathbf{x}_{1:N})$ requires full forward *and* backward passes.

# Backward Sampling

1. First step: perform forward message passing to get $\mathbf{m}_N := p(z_N, \mathbf{x}_{1:N})$.

$$\mathbf{m}_n = \mathbf{A}^\top \mathbf{m}_{n-1} \odot \mathbf{b}_n^* \qquad m_{1k} = p(z_1 = k)p(x_1 \mid z_1 = k)$$

2. Normalize $\mathbf{m}_N$ and sample $z_n$ from the distribution.
3. Then, for $n = N-1, \ldots, 1$, sample $z_n$ from

$$\begin{aligned} p(z_n \mid z_{n+1}, \mathbf{x}_{1:N}) &= p(z_n \mid \mathbf{x}_{1:n})p(z_{n+1} \mid z_n) \times C(z_{n+1}, \mathbf{x}_{1:N}) \\ &\propto \mathbf{m}_n \odot \boldsymbol{\pi}_{\cdot, z_{n+1}} \end{aligned}$$

where $\boldsymbol{\pi}_{\cdot, z_{n+1}}$ is the $z_{n+1}$th column of $\boldsymbol{\pi}$ and $C$ is constant in $z_n$ and can be computed by normalizing.

# Outline

# Summary: Gibbs Sampler for HMM

Goal: Get samples $\{\mathbf{z}^{(s)}, \boldsymbol{\pi}^{(s)}, \boldsymbol{\theta}^{(s)}\}, s = 1, \ldots, S$, where each comes from

$$p(\mathbf{z}, \boldsymbol{\pi}, \boldsymbol{\theta} \mid \mathbf{x}_{1:N})$$

# Summary: Gibbs Sampler for HMM

Algorithm (assuming independent conjugate priors on $\boldsymbol{\pi}, \boldsymbol{\theta}$)

1. Initialize something (e.g., $\mathbf{z}$ via a static clustering approach such as $k$-means)
2. While not tired (or for $s = 1, \ldots, S$)
   (a) Sample $\boldsymbol{\pi}_k \mid \mathbf{z} \sim \text{Dir}(\alpha + N_{k1}, \ldots, \alpha + N_{kK})$
   (b) Sample $\theta_k \mid \mathbf{z}, \mathbf{x}_{1:N}$ by computing hyperparameter updates using $\{\mathbf{x}_n : z_n = k\}$.

$$p(\theta_k \mid \mathbf{z}, \mathbf{x}_{1:N}) \propto p(\theta_k) \prod_{n:z_n=k} f_k(\mathbf{x}_n \mid \theta_k)$$

   (c) Fixing $\boldsymbol{\pi}$ and $\boldsymbol{\theta}$, sample $\mathbf{z}$ by
      (i) Iteratively computing each $\mathbf{m}_n$ using the forward algorithm: $\mathbf{m}_n = \mathbf{A}^\mathsf{T}\mathbf{m}_{n-1} \odot \mathbf{m}_n^*$
      (ii) Iteratively sampling $z_n$ in reverse order according to

$$p(z_n \mid z_{n+1}, \mathbf{x}_{1:N}) \propto \mathbf{m}_n \odot \boldsymbol{\pi}_{\cdot, z_{n+1}}$$

# Outline

# Using the Samples

Having drawn

$$\mathbf{z}^{(s)}, \boldsymbol{\pi}^{(s)}, \boldsymbol{\theta}^{(s)} \sim p(\mathbf{z}, \boldsymbol{\pi}, \boldsymbol{\theta} \mid \mathbf{x}_{1:N}), s = 1, \ldots, S$$

we can now approximate

$$\mathbb{E}_{p(\mathbf{z}, \boldsymbol{\pi}, \boldsymbol{\theta} \mid \mathbf{x}_{1:N})} \{f(\mathbf{z}, \boldsymbol{\pi}, \boldsymbol{\theta})\} \approx \frac{1}{S} \sum_{s=1}^{S} f(\boldsymbol{\pi}^{(s)}, \boldsymbol{\theta}^{(s)})$$

for any $f$.

# Things we might want to do

- Probabilistically "classify" case $n$ by computing

$$p(z_n \mid \mathbf{x}_{1:N}) = \mathbb{E}_{p(\mathbf{z},\boldsymbol{\pi},\boldsymbol{\theta} \mid \mathbf{x}_{1:N})}\left\{p(z_n \mid \mathbf{x}_{1:N}, \boldsymbol{\pi}, \boldsymbol{\theta})\right\}$$

  i.e., averaging over possible parameters
- Evaluate the "marginal marginal" likelihood

$$p(\mathbf{x}_{1:N}) = \mathbb{E}_{p(\mathbf{z},\boldsymbol{\pi},\boldsymbol{\theta} \mid \mathbf{x}_{1:N})}\left\{p(\mathbf{x}_{1:N} \mid \boldsymbol{\pi}, \boldsymbol{\theta})\right\}$$

  e.g., to compare different models or choices of $K$
- Predict/sample future observations according to

$$p(\mathbf{x}_{N+1:N+M}) = \mathbb{E}_{p(\mathbf{z},\boldsymbol{\pi},\boldsymbol{\theta} \mid \mathbf{x}_{1:N})}\left\{p(\mathbf{x}_{N+1:N+M} \mid \boldsymbol{\pi}, \boldsymbol{\theta})\right\}$$