# More on hidden Markov models

ISTA 410 / INFO 510: Bayesian Modeling and Inference

U. of Arizona School of Information

November 25, 2020

# Outline

Last time:

- Temporal models
- Filtering and smoothing in HMMs

Today:

- Fitting in HMMs and unsupervised learning
- HMMs with other observation distributions

# Temporal and dynamical models

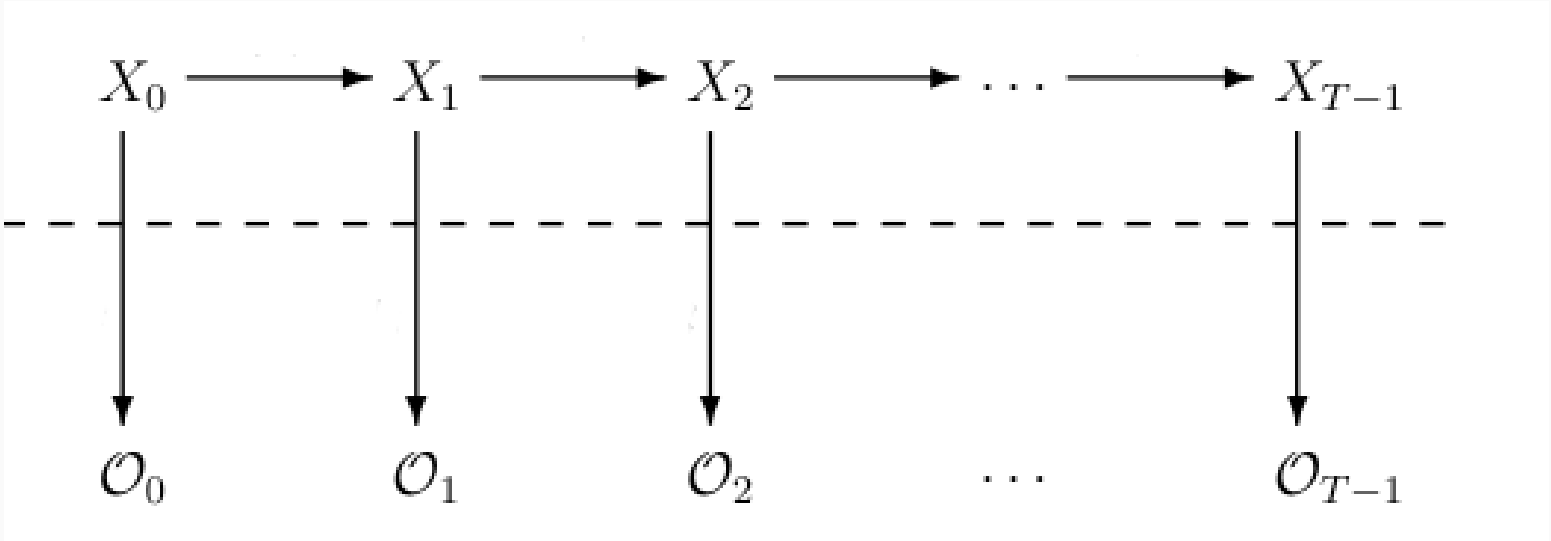
# Temporal and dynamical models

The models we'll look at next model are used to model sequential data, especially time series.

These models are intended to be usable in an on-line fashion – that is, incorporating data in real time as it comes.

# Hidden Markov models

Last time: hidden Markov models




- Latent/hidden system state evolves according to a Markov chain
- Observations based on system state

# Typical inference problems

Typical problems we want to solve, given a sequence of observations $\mathcal{O}$ of time length $T$:

- *Filtering: find the distribution of $X_T$ – that is, the distribution of the current state, accounting for all observations up to now.*
- Prediction: find the distribution of $X_t$ for some $t > T$.
- *Smoothing: find the distribution of $X_t$ for some $1 \leq t < T$. This looks very similar to filtering, but differs in that we can take the observations after time $t$ into account.*
- MAP or best-explanation: find the sequence $(X_i)$ maximizing $P(\mathcal{O}, X)$.
- Fitting: Given a sequence of observations, find the model parameters that maximize $P(\mathcal{O}|\lambda)$.

# Typical inference problems

Typical problems we want to solve, given a sequence of observations $\mathcal{O}$ of time length $T$:

- *Filtering: find the distribution of $X_T$ – that is, the distribution of the current state, accounting for all observations up to now.*
- Prediction: find the distribution of $X_t$ for some $t > T$.
- *Smoothing: find the distribution of $X_t$ for some $1 \leq t < T$. This looks very similar to filtering, but differs in that we can take the observations after time $t$ into account.*
- MAP or best-explanation: find the sequence $(X_i)$ maximizing $P(\mathcal{O}, X)$.
- **Fitting: Given a sequence of observations, find the model parameters that maximize $P(\mathcal{O}|\lambda)$.**

# Hidden Markov models

# Simplest case

In a HMM, the underlying states are governed by a Markov process.

The simplest case, which we'll start with, is a finite state, multinomial HMM:

- Underlying state $X_t$ follows a Markov chain with $N$ states
- Observed values $\mathcal{O}_t$ follow a multinomial distribution conditional on $X_t$

So the model is described by two matrices, $A$ (transition matrix), and $B$ (observation matrix).

To do calculations, we also need to assume a certain distribution $\pi$ on the initial state $X_1$. As a shorthand, I'll use the notation $\lambda = (A, B, \pi)$ to represent a choice of these parameters.

# Naïve filtering

It is clear that we can compute the joint probability of a particular sequence of states:

$$P(X, \mathcal{O}|\lambda) = \pi_{X_1} \prod_{t=1}^{T} A_{X_{t-1},X_t} B_{X_t,\mathcal{O}_t}$$

So, naïvely, we could compute this for all sequences of states, and then

$$P(X_t = x_i) = \sum_{\text{sequences with } X_t = x_i} P(X, \mathcal{O}|\lambda)$$

What's the problem?

# Naïve filtering

It is clear that we can compute the joint probability of a particular sequence of states:

$$P(X, \mathcal{O}|\lambda) = \pi_{X_1} \prod_{t=1}^{T} A_{X_{t-1},X_t} B_{X_t,\mathcal{O}_t}$$

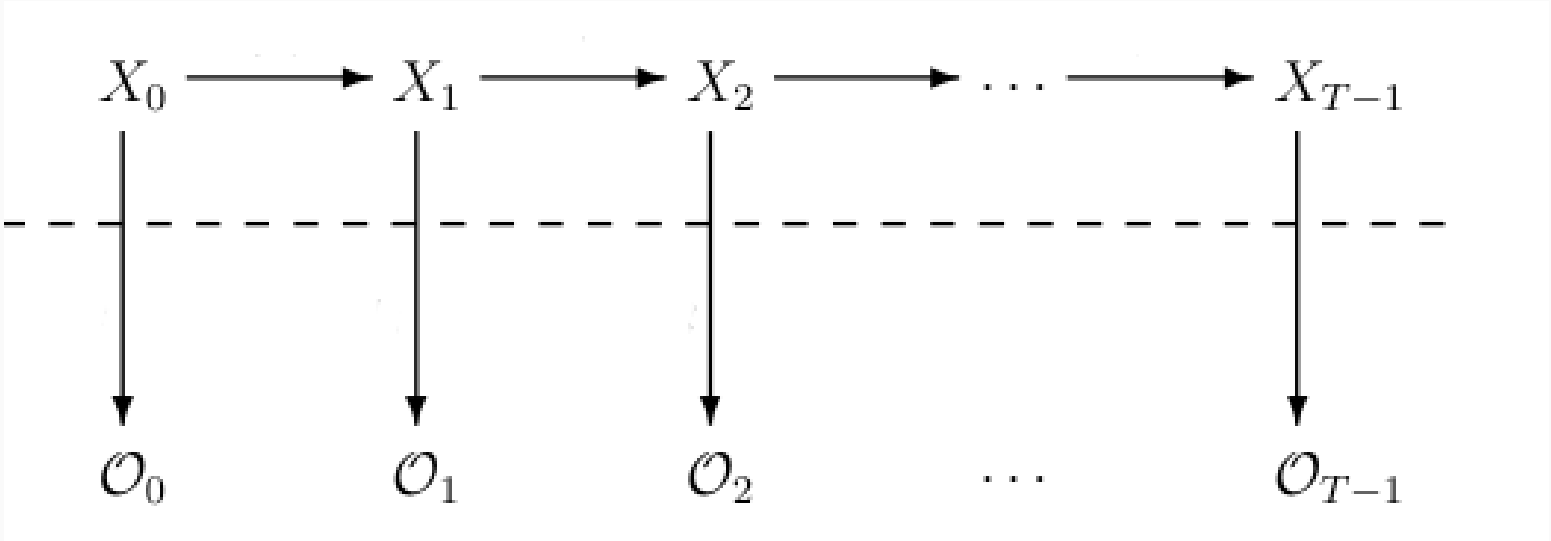
So, naïvely, we could compute this for all sequences of states, and then

$$P(X_t = x_i) = \sum_{\text{sequences with } X_t = x_i} P(X, \mathcal{O}|\lambda)$$

What's the problem? $N^T$ sequences – computationally infeasible for all but short sequences.

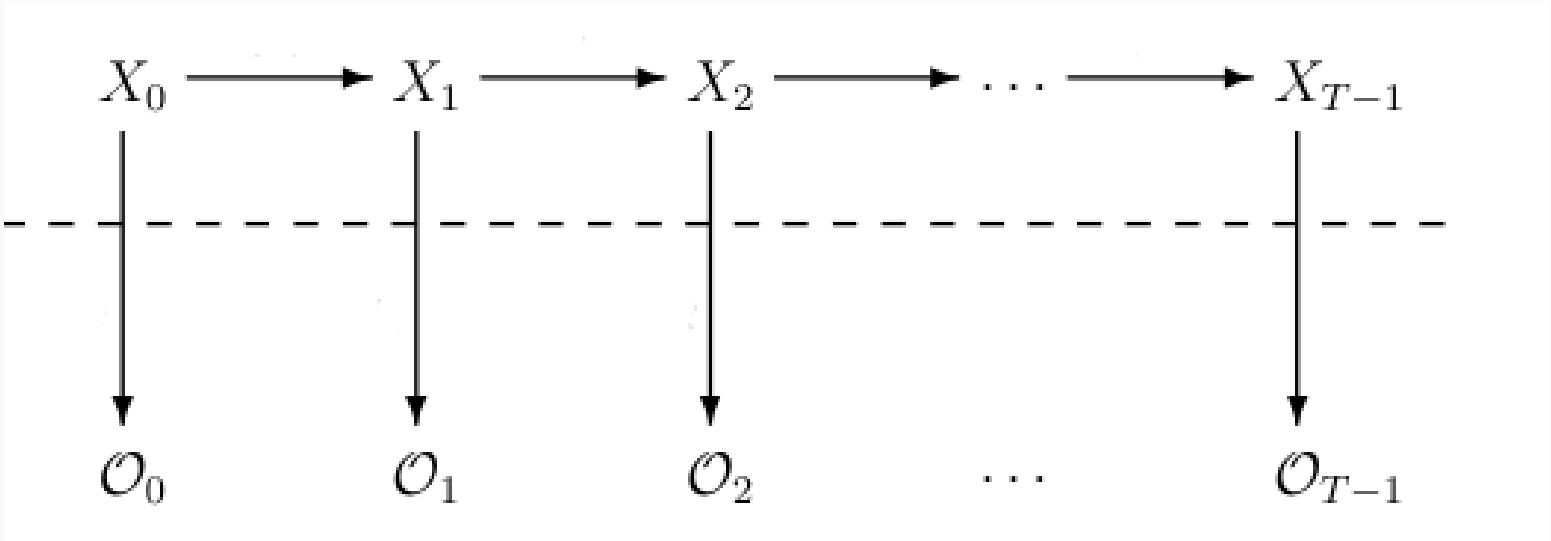
# The forward algorithm

This problem can be solved by the *forward algorithm*, which exploits the Markov property to marginalize on the fly:



# The forward algorithm

This problem can be solved by the *forward algorithm*, which exploits the Markov property to marginalize on the fly:




Let $\alpha_t(x_i) = P(X_t = X_i, \mathcal{O}|\lambda)$; then,

$$\alpha_t(x_i) = B_{x_i,\mathcal{O}_i} \sum_{j=1}^{N} \alpha_{t-1}(x_j) A_{x_j,x_i}$$

# The backward pass

The smoothing problem asks us to calculate $P(X_t = x_i, \mathcal{O})$ for some $t < T$. We could just solve the filtering problem by running the forward algorithm up to time $t$, but we would lose the information from future states.

Solution: do a backward pass too.

Let $\beta_t(x_i) = P(\mathcal{O}_{t:T}|X_t = x_i)$; that is, the probability of the "remaining" observations from time $t$ to the end, given $X_t = x_i$. Then,

$$\beta_t(x_i) = \sum_{j=1}^{N} A_{x_i,x_j} B_{x_i,\mathcal{O}_t} \beta_{t+1}(x_j)$$

so we can recursively calculate from the end of the sequence, letting $\beta_T(x_j) = 1$ for each $j$.

# The forward-backward algorithm

The forward-backward algorithm solves the smoothing problem for HMMs:

$$P(X_t = x_i|\mathcal{O}, \lambda) = \frac{\alpha_t(x_i)\beta_t(x_i)}{P(\mathcal{O}|\lambda)}$$

Where can we get the normalizing constant?

$$P(\mathcal{O}|\lambda) = \sum_{i=1}^{N} \alpha_T(x_i)$$

# Fitting parameters

The fitting problem gives a new challenge:

- Given a fixed state space $\{0, 1, \ldots, n\}$ and a sequence $\mathcal{O}$ of observations, find the model parameters that best fit the sequence $\mathcal{O}$
- i.e., tune $A$ (transition matrix), $B$ (observation matrix), and $\pi$ (initial state distribution)
- Target: maximize $P(\mathcal{O}|A, B, \pi)$

This is a form of unsupervised learning.

# Baum-Welch algorithm

The Baum-Welch algorithm iteratively improves the fit of the model parameters in a two-step process:

- Do a smoothing step, estimating the probability distributions of the hidden states $X_t$
- Re-adjust the model parameters to better fit this estimated distribution

# Idea behind BW algorithm

Intuitively:

- The smoothing step allows us to estimate the probability that the underlying chain is in each state $x_i$ at time $t$
- We can use this to count the estimated probability of transitions from state $x_i$ to state $x_j$
- We can use this, together with the observation sequence, to estimate the probability of each observation from state $x_i$

# Estimating the transition matrix

Smoothing gives us:

$$\gamma_t(i) = \frac{\alpha_t(i)\beta_t(i)}{P(\mathcal{O}|A, B, \pi)}$$

which estimate the probability that the chain was in state $x_i$ at time $t$. We extend this to:

$$\gamma_t(i, j) = \frac{\alpha_t(i)A_{ij}B_{j,\mathcal{O}_t}, \beta_{t+1}(j)}{P(\mathcal{O}|A, B, \pi)}$$

which estimates the probability that the chain was in state $x_i$ at time $t$ *and* state $x_j$ at time $t + 1$.

Then, we re-estimate the transition probability $A_{ij}$ as:

$$A_{ij} = \frac{\sum_t \gamma_t(i, j)}{\sum_t \gamma_t(i)}$$

# Estimating the rest

Similarly, we can re-estimate the observation probability $B_{ij}$ as

$$B_{ij} = \frac{\sum_{t,\mathcal{O}_t=j} \gamma_t(i)}{\sum_t \gamma_(i)}$$

the expected proportion of the time spent in state $i$ that produces observation $j$.

# Estimating the rest

Similarly, we can re-estimate the observation probability $B_{ij}$ as

$$B_{ij} = \frac{\sum_{t,\mathcal{O}_t=j} \gamma_t(i)}{\sum_t \gamma_(i)}$$

the expected proportion of the time spent in state $i$ that produces observation $j$.

The estimate of the initial state vector is just:

# Simple applications

A couple of simple applications that illustrate the usefulness of HMM fitting:

- Pattern recognition in text orthography
- Simple cipher analysis

# Text analysis example

Imagine you're an alien with no knowledge of human language, but you gain access to a sample of English text, and you would like to extract some information about the relationships between characters.

Simplifying assumptions:

- No cases – everything is lowercase
- No digits or punctuation; only characters are letters and spaces

# Text analysis example

Idea: different characters play different roles in the written language;

Approach: fit a hidden Markov model with $k$ different states to a large sample of text, and see if any patterns can be seen.

Let's take a look at the results for $k = 2$.

# HMMs with other observation distributions

# Gaussian HMM

The most common alternative distribution for HMMs is the Gaussian (normal) distribution. In this model:

- $X_t$ still evolves according to a Markov chain with transition matrix $A$
- $\mathcal{O}_t \sim \mathrm{MVNormal}(\mu_{X_t}, \Sigma_{X_t})$
- Result: the observation distributions are Gaussian mixtures

# Filtering and smoothing in Gaussian HMM

What has to change for our filtering and smoothing algorithms?

# Filtering and smoothing in Gaussian HMM

What has to change for our filtering and smoothing algorithms?

- Only change: $P(\mathcal{O}_t = j | X_t = x_i)$ is no longer given by a matrix entry $B_{ij}$
- Instead, we have $p(\mathcal{O}_t = \mathrm{y} | X_t = x_i) = \mathrm{MVNormal}(\mu_i, \Sigma_i)$ for a certain mean vector $\mu_i$, covariance matrix $\Sigma$

# Expectation-maximization algorithms

# EM algorithms

The Baum-Welch algorithm we saw before is an example of a much wider class of algorithms called *expectation-maximization* algorithms.

These are applicable when the observed data depends on hidden/latent state variables as well as model parameters. Roughly, the idea is:

- Expectation step: compute the distribution of hidden state variables, given current model parameters
- Maximization step: compute the model parameters that maximize (log) likelihood given the state parameters from the expectation step

Repeat until done – score model by total log-likelihood of the data.

Section 13.4-13.6 in BDA has another presentation of EM algorithms in a different context.

# EM algorithms

Formally:

- $\theta$: model parameters
- X: hidden variables
- Y: observations
- $L(\theta|X, Y)$: likelihood function

1. E-step: compute $Q(\theta|\hat{\theta}) = E_{X|Y,\hat{\theta}}[\log L(\theta|X, Y)]$
2. M-step: compute $\theta^{\text{new}} = \arg\max_{\theta} Q(\theta|\hat{\theta})$

# BW algorithm as EM

Recall the Baum-Welch algorithm has two steps:

- Perform smoothing to estimate the distribution of each $X_t$, given current transition/observation matrix values
- Update parameter values by counting transitions/observations given distributions of $X_t$

Although we don't explicitly calculate expectations of log-likelihoods, the smoothing step is an E step and the update step is an M step.

# EM for Gaussian HMM

To fit the Gaussian HMM, we only need to make the following modifications to the M step:

- Replace $B_{ij}$ with $\boldsymbol{\mu}_i, \Sigma_i$
- Replace the update of $B_{ij}$ with a maximum-likelihood estimate for a Gaussian, weighted by the estimated state probabilities (from smoothing):

$$\boldsymbol{\mu}_i^{\text{new}} = \frac{\sum_t P(X_t = i)\boldsymbol{y}_t}{\sum_t P(X_t = i)}$$

$$\Sigma_i^{\text{new}} = \frac{\sum_t P(X_t = i)(\boldsymbol{y}_t - \boldsymbol{\mu}_i^{\text{new}})(\boldsymbol{y}_t - \boldsymbol{\mu}_i^{\text{new}})^T}{\sum_t P(X_t = i)}$$

where $y_i$ are the observations.

# A few comments

# Missing data

Suppose we have an incomplete sequence of observations:

$$(\mathcal{O}_t) = (\mathcal{O}_1, \mathcal{O}_2, \ldots, \mathcal{O}_T)$$

where some $\mathcal{O}_t$ are unobserved (NA).

We can still perform forward and backward algorithms for filtering/smoothing; however, steps where $\mathcal{O}_t = NA$ only involve transition probabilities, no observation.

Result: estimated distribution of hidden states relaxes toward the stationary distribution of the MC

# Continuous time

Continuous-time Markov chains do exist, so we could build a HMM on top of one of those.

- Applications:
- How a CT-MC works:
  - Each state $x_i$ has an associated *holding time* – an exponential random variable
  - Chain stays in current state for the holding time and then undergoes a transition according to a
- Challenge: transition times are unobserved, and may not correspond to the observation times

# Continuous time

Reduction to discrete HMM:

- The continuous time chain can be expressed in terms of a *transition rate matrix* $Q$
- Each entry $q_{ij}$ gives the rate parameter for an exponential random variable; transitions from state $i$ are determined by the minimum of the exponential random variables
- Can reduce to a discrete-time Markov chain with transition matrix dependent on the time interval between two observations: $P(t) = \exp(Qt)$

Details: Liu et al., "Efficient Learning of Continuous-Time Hidden Markov Models for Disease Progression" (2015)