CS383/613 – Machine Learning

Markov Models

Overview

- Markov Systems
- Markov Chains
- Hidden Markov Models

Here we need to

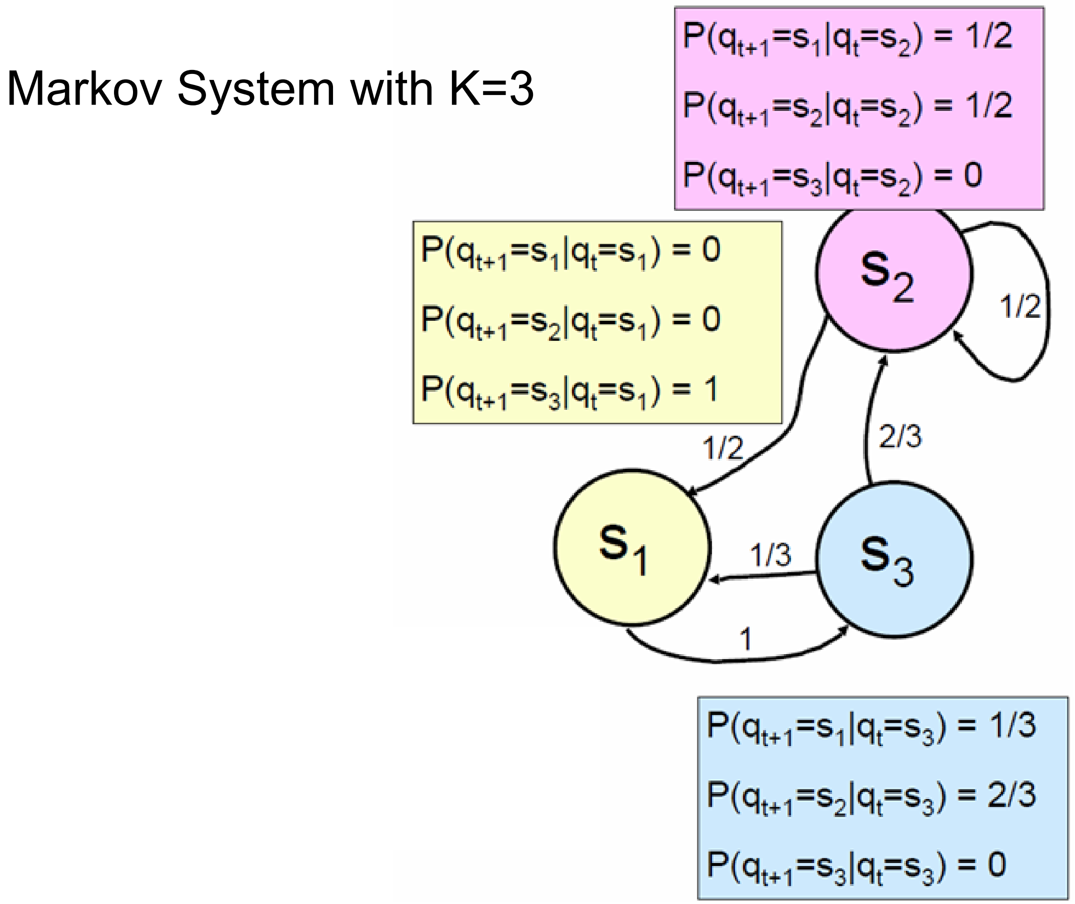
Time-Series Data

- Up until now everything we done is on observations taken at a single moment in time (although we briefly talked about RNNs and LSTMs in the Intro to Deep Learning material).
  - Each of which are temporally independent of one another.
- Some applications look to classify time-series data.
- Examples include:
  - Gesture Recognition
  - Audio classification

A Markov System

- Let a Markov System have:
  - K states, $s = \{s_1, \dots, s_K\}$
  - Discrete time-steps, $t = 1, 2, \dots, T$
- On the t^{th} time-step the system is in exactly one of the available states, call it $q(t) \in \{s_1, \dots, s_K\}$
- Between each time-step, the next state is chosen randomly
  - But based on some distribution, $P\big(q(t+1) = s_j | q(t) = s_i\big)$

A Markov System



Note that this figure uses subscript to denote time, q_t as opposed to parenthesis, $q(t)$.

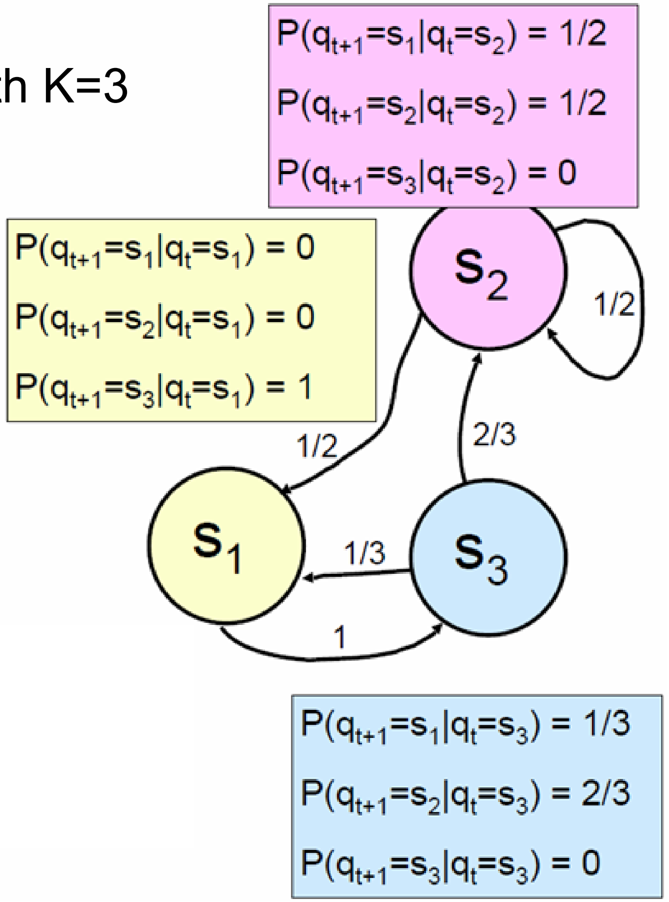
The formulas we'll use will use parenthesis.

A Markov System

- These distributions, $P\left(q(t+1) = s_j | q(t) = s_i\right)$, are typically stored in a *state transition matrix*, A, such that
$$A_{i,j} = P\left(q(t+1) = s_j | q(t) = s_i\right)$$
- Often, we're also given a vector $\boldsymbol{\pi}$ such that π_i is the probability that at time $t = 1$ we are in state i
$$\pi_i = P(q(1) = s_i)$$
- Together we'll say that $\lambda = (s, A, \boldsymbol{\pi})$ defines the Markov system.

A Markov System

Markov System with K=3



$$s = \{s_1, s_2, s_3\}$$

$$\pi = \begin{bmatrix} 1/3 \\ 1/3 \\ 1/3 \end{bmatrix}$$

$$A = \begin{bmatrix} 0 & 0 & 1 \\ \frac{1}{2} & \frac{1}{2} & 0 \\ \frac{1}{3} & \frac{2}{3} & 0 \end{bmatrix}$$

Note that the rows of A sum to one!

Markov Model Evaluation

- A Markov chain is a sequence of states

$$\boldsymbol{q} = \left(q(1), \dots, q(T)\right)$$

- Given a Markov model λ, we can compute the probability of a Markov chain as

$$P(\boldsymbol{q}|\lambda) = \pi_{q(1)} \prod_{t=1}^{T-1} A_{q(t),q(t+1)}$$

- We could also compute this recursively for $t = 1, \dots, T$ as:

$$\alpha(t) = \begin{cases} \pi_{q(t)} & t = 1 \\ A_{q(t-1),q(t)}\alpha(t-1) & otherwise \end{cases}$$

  - And then $P(\boldsymbol{q}|\lambda) = \alpha(T)$
- We call this the **evaluation problem**

Markov Model for Classification

- We could then use this for classification of sequences.
- Given a set of models, $\lambda^{(1)}, \lambda^{(2)}, \ldots$ pertaining to different classes, using Bayes Rule, we can compute:

$$P(y = i|\boldsymbol{q}) \propto P(y = i)P\left(\boldsymbol{q}\middle|\lambda^{(i)}\right)$$

Learning a Markov Model

- How can we learn a Markov Model from observed data?
- Given: some *set* of sequences Q
- Initial state probability vector $\boldsymbol{\pi}$**:**
  - For each state s_k, what percentage of the time did a sequence start at that state?
- State transition matrix A:
  - For each state s_k what percentage of the time did the system transition to state s_j?
- We call this the **learning problem.**
  - Hopefully pretty straight-forward.

Example: Learning a Markov Model

- Given:
  - Three states
  - Observed sequences $Q = \{[s_1, s_1, s_2, s_3, s_1], [s_3, s_2, s_1, s_1]\}$
- What is the initial state probabilities?
  - $\pi_1 = \frac{1}{2}, \pi_2 = 0, \pi_3 = \frac{1}{2}$
- What are the state transitions?

$$A_{1,1} = \frac{2}{3}, A_{1,2} = \frac{1}{3}, A_{1,3} = 0$$
$$A_{2,1} = \frac{1}{2}, A_{2,2} = 0, A_{2,3} = \frac{1}{2}$$
$$A_{3,1} = \frac{1}{2}, A_{3,2} = \frac{1}{2}, A_{3,3} = 0$$

$$\boxed{s = \{s_1, s_2, s_3\}, \pi = \begin{bmatrix} 1/2 \\ 0 \\ 1/2 \end{bmatrix}, A = \begin{bmatrix} \frac{2}{3} & \frac{1}{3} & 0 \\ \frac{1}{2} & 0 & \frac{1}{2} \\ \frac{1}{2} & \frac{1}{2} & 0 \end{bmatrix}}$$

Example: Evaluating a Markov Chain

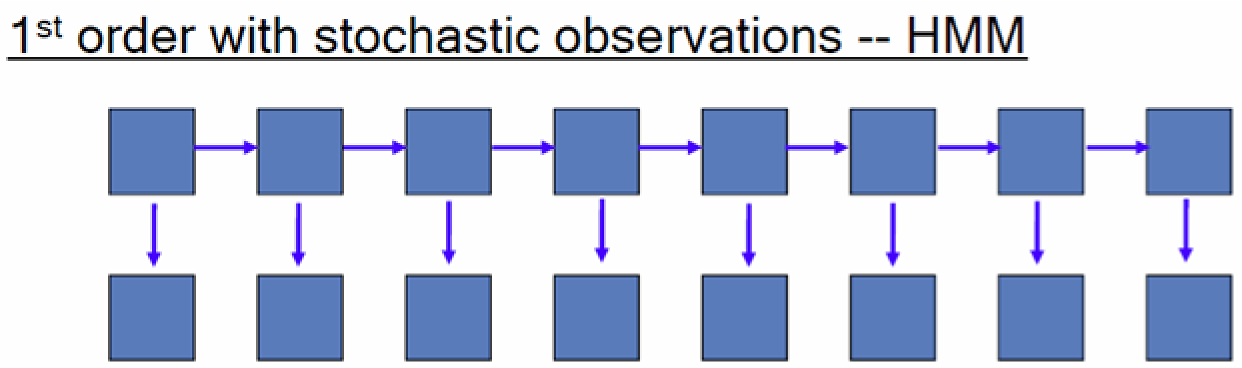
$$s = \{s_1, s_2, s_3\}, \pi = \begin{bmatrix} 1/2 \\ 0 \\ 1/2 \end{bmatrix} A = \begin{bmatrix} 1 & 0 & 0 \\ \frac{1}{2} & 0 & \frac{1}{2} \\ \frac{1}{2} & \frac{1}{2} & 0 \end{bmatrix}$$

- What would be the probability of observing the sequence $\boldsymbol{q} = [s_3, s_2, s_3]$?

$$P(\boldsymbol{q}|\lambda) = \pi_3 \cdot A_{3,2} \cdot A_{2,3} = \left(\frac{1}{2}\right)\left(\frac{1}{2}\right)\left(\frac{1}{2}\right) = \frac{1}{8}$$

Hidden Markov Models

- Often, we can't observe directly the states
- Instead, we observe some other information related to the states
- This is the idea of a *hidden* Markov Model (HMM).



HMM Example: 3 Coins

HTHTHTHHTHTTHTTTTHTTTHTTTTTHHHHTHHTHHHH

- Assume there are 3 coins:
  - One biased towards heads
  - One biased towards tails
  - One non-biased
- Someone tosses one coin repeatedly, then switches to another, etc..
- You observe the sequence of outputs/results (though not which coin was used)
- Can you find the most likely explanation as to which coin he used at each moment in time?

HMM: Definition

- Hidden Markov Model
  - Double stochastic process
  - There is an underlying stochastic process that is not observable (hidden) but can only be observed through another set of stochastic processes that produce the sequence of observed symbols
- Stochastic process #1: Probability of any given coin being used.
- Stochastic process #2: Probability of the current coin generated a head or tail.
- The observations are the outcomes of the tosses
- The biased coins are the hidden states

HMM Notation

- We have a lot of the same stuff as with regular Markov models/chains:
  - States $s = \{s_1, \dots, s_K\}$
  - The state transition matrix, A
  - The initial state probability vector $\boldsymbol{\pi}$
- However, an HMM also has:
  - The set of possible things we can *observe*, $h = \{h_1, \dots, h_M\}$
  - The probability of a state s_i *emitting* observed value h_j as $B_{i,j}$
- Therefore, an HMM, λ, is defined via a 5-tuple:

$$\lambda = (s, h, \boldsymbol{\pi}, A, B)$$

HMM Notation

$$\lambda = (s, h, \boldsymbol{\pi}, A, B)$$

- Now with an HMM a we have an *observed* sequence of length T

$$\mathbf{o} = \big(o(1), \dots, o(T)\big), \text{ where } o(t) \in h$$

- And a true/hidden sequence of the underlying states:

$$\boldsymbol{q} = \big(q(1), \dots, q(T)\big), \text{ where } q(t) \in s$$

HMM Example: Auto-Correct

- There are approximately 104 standard English alpha-number keys.
- There are the keys we meant to hit:
  - The states s
- And the keys we observed as being hit:
  - The states h
- Each key has a probability of starting the word.
  - This provides the initial state probabilities, $\boldsymbol{\pi}$
- Each key has a probability of being pressed after another.
  - This provides our state transition matrix, A
- And finally, each "true key" has a chance if generating (hitting) an observed key.
  - This provides our emissions matrix, B

HMM Applications

- Just like with Markov Models we have
  - The **evaluation** problem
    - What's the probability of an observed sequence given the current HMM?, $P(\boldsymbol{o}|\lambda)$
    - This could also be used for classification (via Bayes' Rule)
  - The **learning** problem
    - Given an observed sequence, find the HMM that maximizes the probability of generating this sequence.

$$\hat{\lambda} = argmax_{\lambda} P(\boldsymbol{o}|\lambda)$$

- But now we also have the **decoding** problem.
    - Given an observed sequence and an HMM, what is the most probable sequence of (hidden) states?

$$\widehat{\boldsymbol{q}} = argmax_{\boldsymbol{q}} P(\boldsymbol{q}|\boldsymbol{o}, \lambda)$$

- Example: What did the user mean to type?

The Evaluation Problem

HMMs

Evaluation Problem

- Given a HMM, λ, we may want to know the probability of observing the sequence $\boldsymbol{o}$:

$$P(\boldsymbol{o}|\lambda)$$

- Recall from Markov models, the probability of true sequences of states can be computed recursively as

$$\alpha(t) = \begin{cases} \pi_{q(1)} & t = 1 \\ A_{q(t-1),q(t)}\alpha(t-1) & otherwise \end{cases}$$

  - And then $P(\boldsymbol{q}|\lambda) = \alpha(T)$

- How does this have to be changed since now we don't observe the states directly?

Evaluation Problem

$$\alpha(t) = \begin{cases} \pi_{q(1)} & t = 1 \\ A_{q(t-1),q(t)}\alpha(t-1) & otherwise \end{cases}$$

- How does this have to be changed since now we don't observe the states directly?
- We now have to consider the possibility that we can from any of the K states
    - And take into consideration the emission probability.
- So now $\alpha_k(t)$ be the probability of arriving at state k at time t, computed recursively as:

$$\alpha_k(t) = \begin{cases} B_{k,o(t)}\pi_k & t = 1 \\ B_{k,o(t)} \sum_i A_{i,k}\alpha_i(t-1) & otherwise \end{cases}$$

Evaluation Problem

$$\alpha_k(t) = \begin{cases} B_{k,o(t)}\pi_k & t = 1 \\ B_{k,o_t}\sum_i A_{i,k}\alpha_i(t-1) & otherwise \end{cases}$$

- And now $P(\boldsymbol{o}|\lambda)$ is:

$$P(\boldsymbol{o}|\lambda) = \sum_{j=1}^{K} \alpha_j(T)$$

- Then just like with Markov models, we could do this computation for

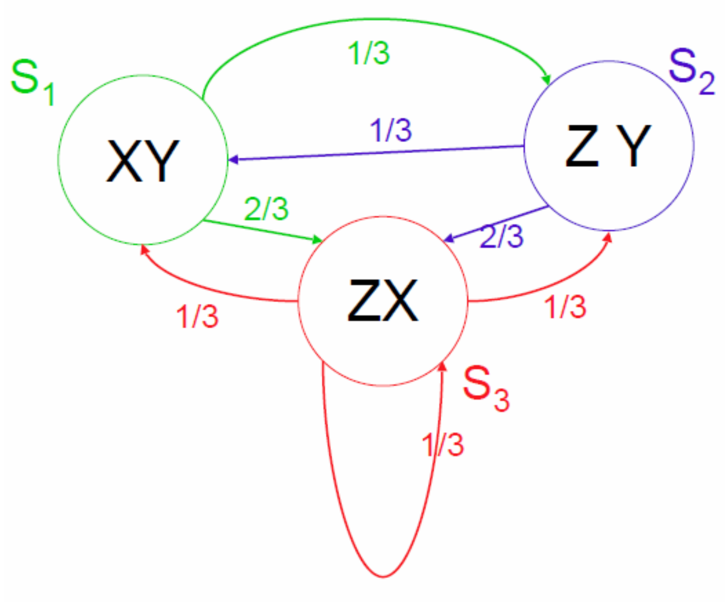
$$P(\lambda|\boldsymbol{o}) \propto P(\lambda)P(\boldsymbol{o}|\lambda)$$

- And use this to decide on the class if we several models:

$$P(y = i|\boldsymbol{o}) \propto P(y = i)P\left(\boldsymbol{o}\middle|\lambda^{(i)}\right)$$

Evaluation Example

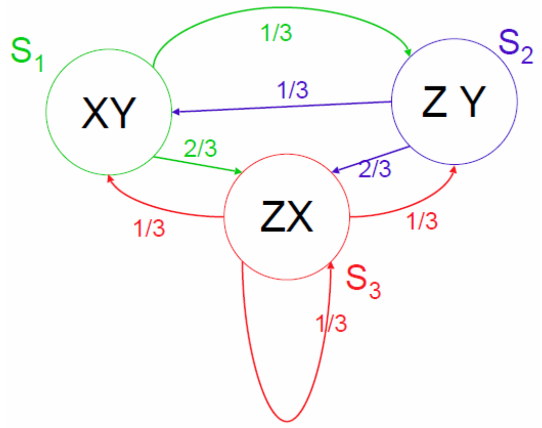
- Suppose we are given the HMM to the right.
- What is the probability that it could have generated the observed sequence $\mathbf{o} = X, X, X$?



$$\pi = \begin{bmatrix} 1/2 \\ 1/2 \\ 0 \end{bmatrix}, A = \begin{bmatrix} 0 & 1/3 & 2/3 \\ 1/3 & 0 & 2/3 \\ 1/3 & 1/3 & 1/3 \end{bmatrix} B = \begin{bmatrix} 1/2 & 1/2 & 0 \\ 0 & 1/2 & 1/2 \\ 1/2 & 0 & 1/2 \end{bmatrix}$$

Evaluation Example

- Time 1 (observed X)
  - $\alpha_1(1) = \frac{1}{4}, \alpha_2(1) = 0, \alpha_3(1) = 0$
- Time 2 (observed X)
  - $\alpha_1(2) = 0, \alpha_2(2) = 0, \alpha_3(2) = \frac{1}{12}$
- Time 3 (observed X)
  - $\alpha_1(3) = \frac{1}{72}, \alpha_2(3) = 0, \alpha_3(3) = \frac{1}{72}$
- $P(\boldsymbol{o}|\lambda) = \frac{1}{72} + 0 + \frac{1}{72} = \frac{1}{36}$



$$\pi = \begin{bmatrix} 1/2 \\ 1/2 \\ 0 \end{bmatrix}, A = \begin{bmatrix} 0 & 1/3 & 2/3 \\ 1/3 & 0 & 2/3 \\ 1/3 & 1/3 & 1/3 \end{bmatrix} B = \begin{bmatrix} 1/2 & 1/2 & 0 \\ 0 & 1/2 & 1/2 \\ 1/2 & 0 & 1/2 \end{bmatrix}$$

$$\alpha_k(t) = \begin{cases} B_{k,o(t)}\pi_k & t = 1 \\ B_{k,o(t)} \sum_i A_{i,k}\alpha_i(t-1) & otherwise \end{cases}$$

The Decoding Problem

HMMs

The Decoding Problem

- Given a sequence of visible states $\boldsymbol{o}$, the decoding problem is to find the most probably sequence of hidden states
  - We call this the *most probably path* (MPP): $P(\boldsymbol{q}|\lambda, \boldsymbol{o})$
- We can solve this using computations similar to the evaluation problem.
- From the evaluation problem:

$$\alpha_k(t) = \begin{cases} B_{k,o(t)}\pi_k & t = 1 \\ B_{k,o(t)} \sum_i A_{i,k}\alpha_i(t-1) & otherwise \end{cases}$$

- For the most probable path we just care about the ***max*** instead of the summation:

$$\alpha_k(t) = \begin{cases} B_{k,o(t)}\pi_k & t = 1 \\ B_{k,o(t)} max_i \left(A_{i,k}\alpha_i(t-1) \right) & otherwise \end{cases}$$

- Then when we arrive at $\mathrm{t} = T$, we choose the $argmax_k\left(\alpha_k(T)\right)$ and *backtrack* the path.

Decoding Example

- Given the following HMM:

$$\boldsymbol{\pi} = \left[\frac{1}{2}, \frac{1}{2}, 0, 0\right], A = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0.2 & 0.3 & 0.1 & 0.4 \\ 0.2 & 0.5 & 0.2 & 0.1 \\ 0.8 & 0.1 & 0 & 0.1 \end{bmatrix}, B = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0.3 & 0.4 & 0.1 & 0.2 \\ 0 & 0.1 & 0.1 & 0.7 & 0.1 \\ 0 & 0.5 & 0.2 & 0.1 & 0.2 \end{bmatrix}$$

- What's the most probably path $\boldsymbol{q}$ for the observed sequence

$$\boldsymbol{o} = (2,5,4)$$

Example

$$A = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0.2 & 0.3 & 0.1 & 0.4 \\ 0.2 & 0.5 & 0.2 & 0.1 \\ 0.8 & 0.1 & 0 & 0.1 \end{bmatrix}, B = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0.3 & 0.4 & 0.1 & 0.2 \\ 0 & 0.1 & 0.1 & 0.7 & 0.1 \\ 0 & 0.5 & 0.2 & 0.1 & 0.2 \end{bmatrix}$$

$\boldsymbol{o} = (\mathbf{2}, 5, 4)$

$\boldsymbol{\pi} = [\frac{1}{2}, \frac{1}{2}, 0, 0]$

$\alpha_k(1) = B_{k,o(1)}\pi_k$

$\alpha_k(t) = B_{k,o(t)} \max_i \left(A_{i,k}\alpha_i(t-1)\right)$

- $t = 1$ (observed 2):
  - $\alpha_1(1) = \left(\frac{1}{2}\right)(0) = 0$
  - $\alpha_2(1) = \left(\frac{1}{2}\right)(0.3) = 0.15,$
  - $\alpha_3(1) = (0)(0.1) = 0,$
  - $\alpha_4(1) = (0)(0.5) = 0$
- $t = 2$ (observed 5)
  - $\alpha_1(2) = 0 \cdot \max(\ldots) = 0$
    - Dead End
  - $\alpha_2(2) = 0.2 \cdot \max(0, 0.3 \cdot 0.15, 0, 0) = 0.009$
    - $\text{mpp}_2(2)=(2)$
  - $\alpha_3(2) = 0.1 \cdot \max(0, 0.1 \cdot 0.15, 0, 0) = 0.0015$
    - $\text{mpp}_3(2)=(2)$
  - $\alpha_4(2) = 0.2 \cdot \max(0, 0.4 \cdot 0.15, 0, 0) = 0.012$
    - $\text{mpp}_4(2)=(2)$

$\boldsymbol{o} = (2, \mathbf{5}, 4)$

Example

$$A = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0.2 & 0.3 & 0.1 & 0.4 \\ 0.2 & 0.5 & 0.2 & 0.1 \\ 0.8 & 0.1 & 0 & 0.1 \end{bmatrix}, B = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0.3 & 0.4 & 0.1 & 0.2 \\ 0 & 0.1 & 0.1 & 0.7 & 0.1 \\ 0 & 0.5 & 0.2 & 0.1 & 0.2 \end{bmatrix}$$

$$\alpha_k(t) = B_{k,o(t)} \max_i \left(A_{i,k}\alpha_i(t-1)\right)$$

$$\boldsymbol{o} = (2,5,\boldsymbol{4})$$

- $t = 2$:
  - $\alpha_1(2) = 0$, $mpp_1(2)$=N/A
  - $\alpha_2(2) = 0.009$, $mpp_2(2)$=(2)
  - $\alpha_3(2) = 0.0015$, $mpp_3(2)$ = (2)
  - $\alpha_4(2) = 0.012$, $mpp_4(2)$ = (2)
- $t = 3$ (observed 4)
  - $\alpha_1(3) = 0 \cdot \max(\dots) = 0$
    - $mpp_1(3)$=N/A
  - $\alpha_2(3) = 0.1 \cdot \max(0, \mathbf{0.3 \cdot 0.009}, 0.5 \cdot 0.0015, 0.1 \cdot 0.012) = 0.00027$
    - $mpp_2(3)$=(2,2)
  - $\alpha_3(3) = 0.7 \cdot \max(0, \mathbf{0.1 \cdot 0.009}, \ 0.2 \cdot 0.0015, 0 \cdot 0.012) = 0.00063$
    - $mpp_3(3)$=(2,2)
  - $\alpha_4(3) = 0.1 \cdot \max(0, \mathbf{0.4 \cdot 0.009}, 0.1 \cdot 0.0015, 0.1 \cdot 0.012) = 0.00036$
    - $mpp_4(3)$=(2,2)
- So most likely path was 2→2→3

The Learning Problem

HMMs

The Learning Problem

- For both the evaluation and decoding problems we need to know the model already.
- How can we learn it?
- It’s not quite as simple as with a (non-hidden) Markov Model since now the true state sequence is hidden!
- Instead, we’re going to use an expectation maximization (EM) algorithm to find the parameters of our HMM that best explain the observed sequence.

EM for HMMs

- An expectation-maximization (EM) algorithm looks to learn a model by iterating between the following two (until convergence):

1. Expectation – Use the current model to make predictions (expectations)
2. Maximization – Use the expectations to update the model to better fit these expectations.

EM for HMMs

- In the context of hidden Markov models, this looks like:
  - Expectation
    - Given a model, λ we can say stuff about our observation sequence $\boldsymbol{o}$
  - Maximization:
    - Given what we say about $\boldsymbol{o}$ can we updated our model to better fit this?

EM for HMMs

Expectation

- Let's compute the probabilities of arriving at state k at time t given the sequence $(o(1), o(2), \dots, o(t))$
- **And** coming from it to generate the sequence $(o(t+1), o(t+2), \dots, o(T))$
- From the evaluation problem we can determine the probability of arriving at s_k at time t:

$$\alpha_k(t) = \begin{cases} B_{k,o(t)}\pi_k & t = 1 \\ B_{k,o(t)} \sum_i A_{i,k}\alpha_i(t-1) & otherwise \end{cases}$$

EM for HMMs

<u>Expectation</u>

- Similarly, we can compute the probability of generating the remainder of the sequence if we start from s_k at time t.
- This is most easily done by recurring *backwards* from time T to time t.
- Or, better yet, we can again leverage the recurrent relation to compute this for $t = 1,2, \ldots, T$, but now using *backwards recursion*
- For $t = T, T-1, \ldots, 1$

$$\beta_k(t) = \begin{cases} 1 & t = T \\ \sum_i A_{k,i} B_{i,o(t+1)} \beta_i(t+1) & otherwise \end{cases}$$

EM for HMMs

Expectation

- Now let's use $\alpha_k(t)$ and $\beta_k(t)$ to compute a value proportional to the probability of state s_k at time t:

$$\gamma_k(t) = P(q(t) = s_k | \boldsymbol{o}, \lambda) = \alpha_k(t)\beta_k(t)$$

EM for HMMs

Maximization

- Now we need to maximize!
- Given $\gamma_k(t)$, we can update $\boldsymbol{\pi}, A, B$
- Let's first compute values *proportional* to these, then normalize things so they add to one.
- The initial state probabilities, π_k are just taken directly from γ at time $t = 1$!

$$\pi_k \propto \gamma_k(1)$$

- The state transition matrix probabilities are computing using γ

$$A_{i,j} \propto \sum_{t=1}^{T-1} \gamma_i(t)\gamma_j(t+1)$$

- And finally, the emission matrix probabilities are computed using the values of γ when on observed value h_j is observed.

$$B_{i,j} \propto \sum_{t=1}^{T} (o(t) == j)\gamma_i(t)$$

EM for HMMs

Maximization

$$\pi_k \propto \gamma_k(1), \qquad A_{i,j} \propto \sum_{t=1}^{T-1} \gamma_i(t)\gamma_j(t+1)\,, \qquad B_{i,j} \propto \sum_{t=1}^{T} (o(t) == j)\gamma_i(t)$$

- Finally we must *normalize* these so we get our proper probability distributions:
  - Divide $\boldsymbol{\pi}$ by its sum, so $\boldsymbol{\pi}$ now sums to one.
  - Divide each *row* of A and B by their sum, so that each row sums to one.

EM for HMMs

Here's the pseudocode for the learning process (known as the Baum-Welch algorithm):

1. Get your observations $\boldsymbol{o} = o(1), \dots, o(T)$
2. Guess your first model λ. Random?
3. Until convergence do steps 4 and 5
4. Do expectation via estimation
   - $\alpha_k(t), \beta_k(t), \gamma_k(t)$
5. Do maximization
   - $\pi_i \propto \gamma_i(1)$
   - $A_{i,j} \propto \sum_{t=1}^{T-1} \gamma_i(t)\gamma_j(t+1)$
   - $B_{i,j} \propto \sum_{t=1}^{T}(o(t) == j)\gamma_i(t)$
   - Normalize these to get proper distributions.

Example

- Let's try to find the HMM of a criminal traveling between LA and NY!
- At any given moment, we observe one of three things:
  - We're told they are in NY
  - We're told they are in LA
  - No one knows where they are (null)
- Let's start with no prior knowledge, i.e uniform distributions:

$$\pi = \begin{bmatrix} 1/2 \\ 1/2 \end{bmatrix}$$

	LA	NY
LA	1/2	1/2
NY	1/2	1/2

	LA	NY	Null
LA	1/3	1/3	1/3
NY	1/3	1/3	1/3

Example

- The FBI has been tracking reports over 4 time instances and observed the sequence:
$$\boldsymbol{o} = (NULL, LA, LA, NY)$$
- Using our current model and these observations we can already do things like:
  1. How good is our model? Evaluation Problem
  2. What was likely his/her actual states? Decoding problem
  3. What's the probability that we're in a given ending state?
  4. What's the probability distribution at the next period $t = 5$ (so we can catch him/her!):
- Can we update the model to make it better!?
  - Learning Problem

Example EM for HMM

$$\pi = \begin{bmatrix} 1/2 \\ 1/2 \end{bmatrix}$$

	LA	NY
LA	1/2	1/2
NY	1/2	1/2

	LA	NY	Null
LA	1/3	1/3	1/3
NY	1/3	1/3	1/3

$$\alpha_i(1) = \pi_i B_{i,o(1)}$$

$$\alpha_i(t+1) = B_{i,o(t+1)} \sum_{j=1}^{N} \alpha_j(t) A_{j,i}$$

- $\boldsymbol{o} = (NULL, LA, LA, NY)$
- Iteration 1: Forward Estimation
  - $\alpha_{LA}(1) = \pi_{LA} B_{LA,NULL} = 0.17$
  - $\alpha_{NY}(1) = \pi_{NY} B_{NY,NULL} = 0.17$
  - $\alpha_{LA}(2) = B_{LA,LA}\left(\alpha_{LA}(1)A_{LA,LA} + \alpha_{NY}(1)A_{NY,LA}\right) = 0.33 * (0.17 * 0.5 + 0.17 * 0.5) = 0.06$
  - $\alpha_{NY}(2) = B_{NY,LA}\left(\alpha_{LA}(1)A_{LA,NY} + \alpha_{NY}(1)A_{NY,NY}\right) = 0.33 * (0.17 * 0.5 + 0.17 * 0.5) = 0.06$
  - $\alpha_{LA}(3) = B_{LA,LA}\left(\alpha_{LA}(2)A_{LA,LA} + \alpha_{NY}(2)A_{NY,LA}\right) = 0.33 * (0.06 * 0.5 + 0.06 * 0.5) = 0.02$
  - $\alpha_{NY}(3) = B_{NY,LA}\left(\alpha_{LA}(2)A_{LA,NY} + \alpha_{NY}(2)A_{NY,NY}\right) = 0.33 * (0.06 * 0.5 + 0.06 * 0.5) = 0.02$
  - $\alpha_{LA}(4) = B_{LA,NY}\left(\alpha_{LA}(3)A_{LA,LA} + \alpha_{NY}(3)A_{NY,LA}\right) = 0.33 * (0.02 * 0.5 + 0.02 * 0.5) = 0.006$
  - $\alpha_{NY}(4) = B_{NY,NY}\left(\alpha_{LA}(3)A_{LA,NY} + \alpha_{NY}(3)A_{NY,NY}\right) = 0.33 * (0.02 * 0.5 + 0.02 * 0.5) = 0.006$

Example EM for HMM

	LA	NY
LA	1/2	1/2
NY	1/2	1/2

	LA	NY	Null
LA	1/3	1/3	1/3
NY	1/3	1/3	1/3

$$\beta_i(T) = 1$$
$$\beta_i(t) = \sum_{j=1}^{N} \beta_j(t+1) A_{i,j}\, B_{j,o(t+1)}$$

- $\boldsymbol{o} = (NULL, LA, LA, NY)$
- Iteration 1: Backwards Procedure
  - $\beta_{LA}(4) = 1$
  - $\beta_{NY}(4) = 1$
  - $\beta_{LA}(3) = \left(\beta_{LA}(4)A_{LA,LA}B_{LA,NY} + \beta_{NY}(4)A_{LA,NY}B_{NY,NY}\right) = 1 * 0.5 * 0.33 + 1 * 0.5 * 0.33 = 0.33$
  - $\beta_{NY}(3) = \left(\beta_{LA}(4)A_{NY,LA}B_{LA,NY} + \beta_{NY}(4)A_{NY,NY}B_{NY,NY}\right) = 1 * 0.5 * 0.33 + 1 * 0.5 * 0.33 = 0.33$
  - $\beta_{LA}(2) = \left(\beta_{LA}(3)A_{LA,LA}B_{LA,LA} + \beta_{NY}(3)A_{LA,NY}B_{NY,LA}\right) = 0.33 * 0.5 * 0.33 + 0.33 * 0.5 * 0.33 = 0.11$
  - $\beta_{NY}(2) = \left(\beta_{LA}(3)A_{NY,LA}B_{LA,LA} + \beta_{NY}(3)A_{NY,NY}B_{NY,LA}\right) = 0.33 * 0.5 * 0.33 + 0.33 * 0.5 * 0.33 = 0.11$
  - $\beta_{LA}(1) = \left(\beta_{LA}(2)A_{LA,LA}B_{LA,LA} + \beta_{NY}(2)A_{LA,NY}B_{NY,LA}\right) = 0.11 * 0.5 * 0.33 + 0.11 * 0.5 * 0.33 = 0.04$
  - $\beta_{NY}(1) = \left(\beta_{LA}(2)A_{NY,LA}B_{LA,LA} + \beta_{NY}(2)A_{NY,NY}B_{NY,LA}\right) = 0.11 * 0.5 * 0.33 + 0.11 * 0.5 * 0.33 = 0.04$

Example: EM for HMM

$\gamma_i(t) = P(q_t = s_i|\boldsymbol{o}, \lambda) = \alpha_i(t)\beta_i(t)$

- $\boldsymbol{o} = (NULL, LA, LA, NY)$
- Iteration 1: Gamma
  - $\gamma_{LA}(1) = \alpha_{LA}(1)\beta_{LA}(1) = 0.0062$
  - $\gamma_{NY}(1) = \alpha_{NY}(1)\beta_{NY}(1) = 0.0062$
  - $\gamma_{LA}(2) = \alpha_{LA}(2)\beta_{LA}(2) = 0.0062$
  - $\gamma_{NY}(2) = \alpha_{NY}(2)\beta_{NY}(2) = 0.0062$
  - $\gamma_{LA}(3) = \alpha_{LA}(3)\beta_{LA}(3) = 0.0062$
  - $\gamma_{NY}(3) = \alpha_{NY}(3)\beta_{NY}(3) = 0.0062$
  - $\gamma_{LA}(4) = \alpha_{LA}(4)\beta_{LA}(4) = 0.0062$
  - $\gamma_{NY}(4) = \alpha_{NY}(4)\beta_{NY}(4) = 0.0062$

$$\begin{aligned}
\gamma_{LA}(1) &= \alpha_{LA}(1)\beta_{LA}(1) = 0.0062 \\
\gamma_{NY}(1) &= \alpha_{NY}(1)\beta_{NY}(1) = 0.0062 \\
\gamma_{LA}(2) &= \alpha_{LA}(2)\beta_{LA}(2) = 0.0062 \\
\gamma_{NY}(2) &= \alpha_{NY}(2)\beta_{NY}(2) = 0.0062 \\
\gamma_{LA}(3) &= \alpha_{LA}(3)\beta_{LA}(3) = 0.0062 \\
\gamma_{NY}(3) &= \alpha_{NY}(3)\beta_{NY}(3) = 0.0062 \\
\gamma_{LA}(4) &= \alpha_{LA}(4)\beta_{LA}(4) = 0.0062 \\
\gamma_{NY}(4) &= \alpha_{NY}(4)\beta_{NY}(4) = 0.0062
\end{aligned}$$

Example EM for HMM

$A_{i,j} \propto \sum_{t=1}^{T-1} \gamma_i(t)\gamma_j(t+1)$
$\pi_i \propto \gamma_i(1)$

- Iteration 1: Maximization
  - $\pi_{LA} \propto \gamma_{LA}(1) = 0.0062$
  - $\pi_{NY} \propto \gamma_{NY}(1) = 0.0062$
  - $A_{LA,LA} \propto \sum_{t=1}^{T-1} \gamma_{LA}(t)\gamma_{LA}(t+1) = 0.00014$
  - $A_{LA,NY} \propto \sum_{t=1}^{T-1} \gamma_{LA}(t)\gamma_{NY}(t+1) = 0.00014$
  - $A_{NY,LA} \propto \sum_{t=1}^{T-1} \gamma_{NY}(t)\gamma_{LA}(t+1) = 0.00014$
  - $A_{NY,NY} \propto \sum_{t=1}^{T-1} \gamma_{NY}(t)\gamma_{NY}(t+1) = 0.00014$

$\gamma_{LA}(1) = \alpha_{LA}(1)\beta_{LA}(1) = 0.0062$
$\gamma_{NY}(1) = \alpha_{NY}(1)\beta_{NY}(1) = 0.0062$
$\gamma_{LA}(2) = \alpha_{LA}(2)\beta_{LA}(2) = 0.0062$
$\gamma_{NY}(2) = \alpha_{NY}(2)\beta_{NY}(2) = 0.0062$
$\gamma_{LA}(3) = \alpha_{LA}(3)\beta_{LA}(3) = 0.0062$
$\gamma_{NY}(3) = \alpha_{NY}(3)\beta_{NY}(3) = 0.0062$
$\gamma_{LA}(4) = \alpha_{LA}(4)\beta_{LA}(4) = 0.0062$
$\gamma_{NY}(4) = \alpha_{NY}(4)\beta_{NY}(4) = 0.0062$

Example EM for HMM

$B_{ij} \propto \sum_{t=1}^{T}(o(t) == j)\gamma_i(t)$

$\boldsymbol{o} = (NULL, LA, LA, NY)$

$\gamma_{LA}(1) = \alpha_{LA}(1)\beta_{LA}(1) = 0.0062$
$\gamma_{NY}(1) = \alpha_{NY}(1)\beta_{NY}(1) = 0.0062$
$\gamma_{LA}(2) = \alpha_{LA}(2)\beta_{LA}(2) = 0.0062$
$\gamma_{NY}(2) = \alpha_{NY}(2)\beta_{NY}(2) = 0.0062$
$\gamma_{LA}(3) = \alpha_{LA}(3)\beta_{LA}(3) = 0.0062$
$\gamma_{NY}(3) = \alpha_{NY}(3)\beta_{NY}(3) = 0.0062$
$\gamma_{LA}(4) = \alpha_{LA}(4)\beta_{LA}(4) = 0.0062$
$\gamma_{NY}(4) = \alpha_{NY}(4)\beta_{NY}(4) = 0.0062$

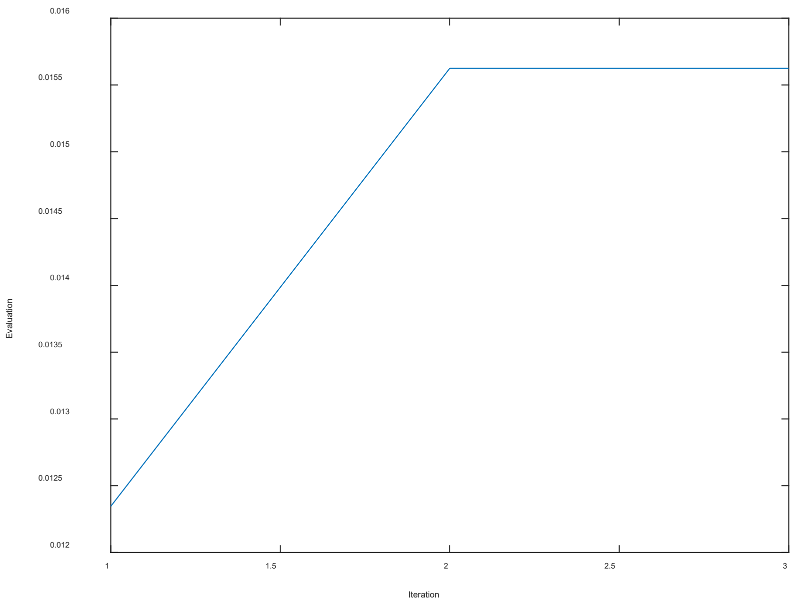
- Iteration 1: Maximization
  - $B_{LA,LA} \propto \sum_{t=1}^{T}(o_t == LA)\gamma_{LA}(t) = 0.0123$
  - $B_{LA,NY} \propto \sum_{t=1}^{T}(o_t == NY)\gamma_{LA}(t) = 0.0062$
  - $B_{LA,NULL} \propto \sum_{t=1}^{T}(o_t == NULL)\gamma_{LA}(t) = 0.0062$
  - $B_{NY,LA} \propto \sum_{t=1}^{T}(o_t == LA)\gamma_{NY}(t) = 0.0123$
  - $B_{NY,NY} \propto \sum_{t=1}^{T}(o_t == NY)\gamma_{NY}(t) = 0.0062$
  - $B_{NY,NULL} \propto \sum_{t=1}^{T}(o_t == NULL)\gamma_{NY}(t) = 0.0062$

Example EM for HMM

- $\pi \propto \begin{bmatrix} 0.0062 \\ 0.0062 \end{bmatrix}$
- $A \propto \begin{bmatrix} 0.00014 & 0.00014 \\ 0.00014 & 0.00014 \end{bmatrix}$
- $B \propto \begin{bmatrix} 0.0123 & 0.0062 & 0.0062 \\ 0.0123 & 0.0062 & 0.0062 \end{bmatrix}$
- Now we must normalize these to be proper probabilities:
- $\pi \rightarrow \begin{bmatrix} 1/2 \\ 1/2 \end{bmatrix}$
- $A \rightarrow \begin{bmatrix} 1/2 & 1/2 \\ 1/2 & 1/2 \end{bmatrix}$
- $B \rightarrow \begin{bmatrix} 1/2 & 1/4 & 1/4 \\ 1/2 & 1/4 & 1/4 \end{bmatrix}$

Example EM for HMM

- Sanity Check
- Let's evaluate using our original (random) HMM
  - $P(\boldsymbol{o}|\lambda) = 0.0123$
- Let's evaluate using our (slightly) updated HMM (one iteration)
  - $P(\boldsymbol{o}|\lambda) = 0.0156$
- Converges after just 3 epochs (after all, there's just one observation)
  - $P(\boldsymbol{o}|\lambda) = 0.0156$
  - $\boldsymbol{\pi} = \left[\frac{1}{2}, \frac{1}{2}\right]^T$
  - $A = \begin{bmatrix} 1/2 & 1/2 \\ 1/2 & 1/2 \end{bmatrix}$
  - $B = \begin{bmatrix} 1/2 & 1/4 & 1/4 \\ 1/2 & 1/4 & 1/4 \end{bmatrix}$
  - Does anything look odd with this?
  - How can we deal with it

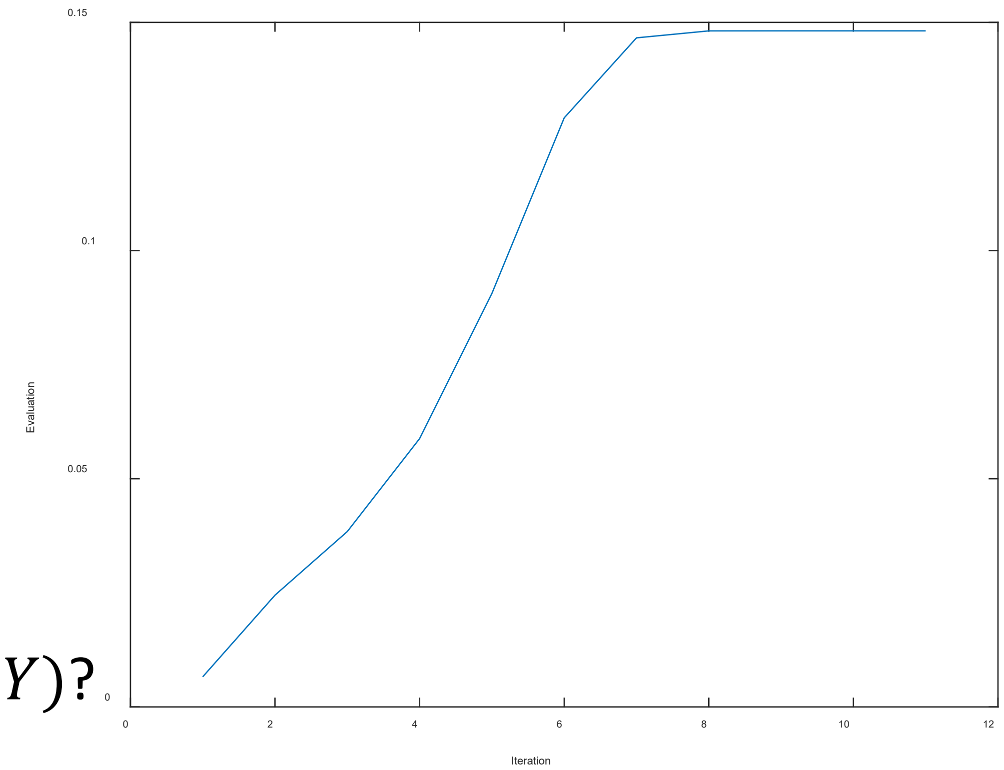


Example EM for HMM

- Just like with gradient based learning, if we don't have prior information, it's best to initialize our weights to be random values.
  - Helps avoid getting stuck in a bad maxima.
- So I'll initialize them using random numbers, then normalize as distributsions, as necessary.
- Doing this, I got initializations of:
  - $\boldsymbol{\pi} = [0.47, 0.53]^T$
  - $A = \begin{bmatrix} 0.17 & 0.83 \\ 0.90 & 0.10 \end{bmatrix}$
  - $B = \begin{bmatrix} 0.2 & 0.69 & 0.11 \\ 0.22 & 0.39 & 0.39 \end{bmatrix}$
- And an initial evaluation of:
  - $P(\boldsymbol{o}|\lambda) = 0.0066$
  - Even worst than when we did uniform assignment!

Example EM for HMM

- Things converged after 11 epochs:
  - $P(\boldsymbol{o}|\lambda) = 0.1481$
    - Much better than with our uniform initialization!
  - $\boldsymbol{\pi} = [0,1]^T$
  - $A = \begin{bmatrix} 1 & 0 \\ 1 & 0 \end{bmatrix}$
  - $B = \begin{bmatrix} 2/3 & 1/3 & 0 \\ 0 & 0 & 1 \end{bmatrix}$
- Does this make sense for $\boldsymbol{o} = (NULL, LA, LA, NY)$?



Random HMM

- How good (relatively) is this HMM at generating this sequence of observations?
- In a purely random HMM, observed values will each occurs with a probability of $\frac{1}{M}$ (regardless of the state, again, since it's purely random).
- Therefore, we can compute $P(\boldsymbol{o}|\lambda)$ as:

$$P(\boldsymbol{o}|\lambda) = \prod_{t=1}^{T} o(t) = \left(\frac{1}{M}\right)^{T}$$

- For our example this is:

$$P(\boldsymbol{o}|\lambda) = \left(\frac{1}{3}\right)^{4} = 0.0123$$

- Compare this to our last results

$$P(\boldsymbol{o}|\lambda) = 0.1481$$

Continuous HMM

- Often, we observe continuous values.
- How can we make an learn/use an HMM where our observations are continuous?
- Our formulas require:

$$P(o(t)|q(t) = s_i)$$

- We'll still have discrete states, $S = \{s_1, \dots, s_K\}$
- To work in discrete space, we'll now need to convert our observed values into categorical ones, so that we have discrete states.
- To work in natively with continuous values, we'll need to assume some distributions…
  - More on this later.

References

- http://www.cs.rochester.edu/u/james/CSC248/Lec11.pdf
- http://ocw.mit.edu/courses/aeronautics-and-astronautics/16-410-principles-of-autonomy-and-decision-making-fall-2010/lecture-notes/MIT16_410F10_lec21.pdf
- http://personal.ee.surrey.ac.uk/Personal/P.Jackson/tutorial/hmm_tut4.pdf