CS 615 – Deep Learning

Markov Models

Overview

- Markov Systems
- Markov Chains
- Hidden Markov Models

Time-Series Data

- Up until now everything we done is on observations taken at a single moment in time.
  - Each of which are temporally independent of one another.
- Some applications look to classify time-series data.
- Examples include:
  - Gesture Recognition
  - Audio classification

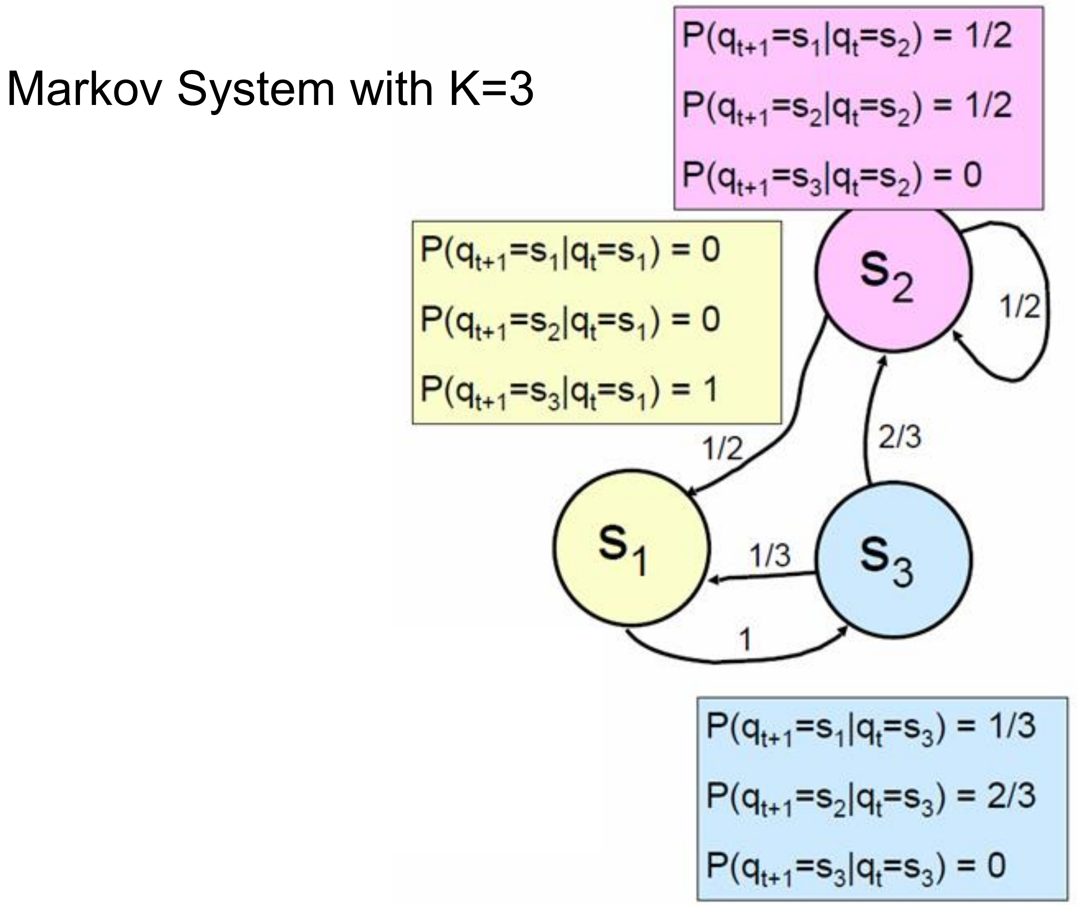
Time Series Data

- When any two samples are not independent
  - In stock prediction, today's price is impacted by yesterday's price
- Should be able to start anywhere
  - Time series data is often continuous
- Type of data must not change
  - All of the changes in the data should be explainable and consistent

A Markov System

- Let a Markov System have:
  - K states, called $s_1, \dots, s_K$
  - Discrete time-steps, $t = 1, \dots, t = T$
- 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, $P\left(q_{t+1} = s_j | q_t = s_i\right)$

A Markov System



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

$$a_{ij} = P\left(q_{t+1} = s_j | q_t = s_i\right)$$

- Often we're also given a vector π such that π_i is the probability that at time $t = 1$ we are in state i

$$\pi_i = P(q_1 = s_i)$$

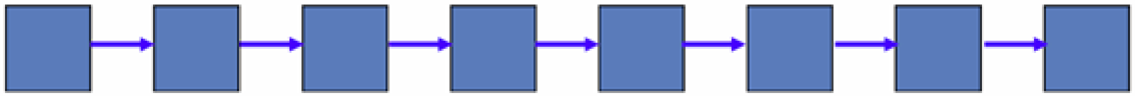
- Together we'll say that $\lambda = (S, A, \pi)$ defines the Markov system.

Markov Chains

- A Markov chain is a sequence of states
$$Q = (q_1, \dots, q_T)$$
- We can use these for things like prediction and classification.
- Although we could use information further back in the chain to help us make our decision at time $t + 1$ typically we only use information from time t
- This is called a *first order Markov chain.*

1st order Markov chain

$$\mathbf{x} = (x_1, x_2, \ldots, x_t, \ldots, x_T)$$



Markov Model

- Given a Markov Model specified by its state transition probability matrix, we easily compute the probability of a sequence of states $Q = (q_1, \dots, q_T)$ occurring, $P(Q|\lambda)$
  - Especially for a 1st order Markov chain

$$P(Q|\lambda) = \pi_{q_1} \prod_{i=1}^{T-1} a_{q_t, q_{t+1}}$$

- We could then use this for classification of sequences.

Markov Model

- If λ_i defines the model for class i then we may be interested in $P(\lambda_i|Q)$ for each $i = 1, \ldots, C$ different classes.

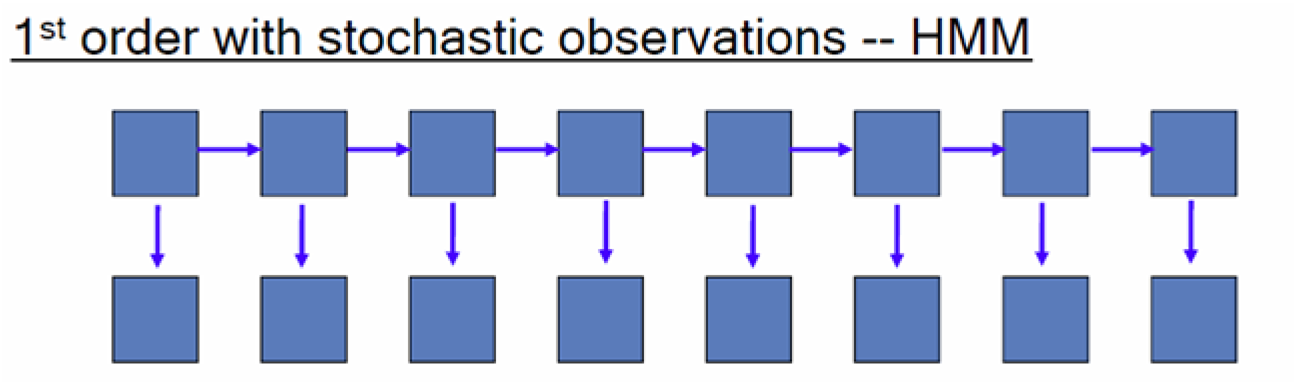
$$P(\lambda_i|Q) = \frac{P(\lambda_i)P(Q|\lambda_i)}{P(Q)}$$

- As usual since $P(Q)$ is class-independent we can just ignore it (or later divide by the sum) and choose:

$$i^* = argmax_i\big(P(\lambda_i)P(Q|\lambda_i)\big)$$

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 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 coins he used?

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: heads or tails?
- Stochastic process #2: which coin?
- 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 $s_1, \dots, s_K$
  - A chain of length T
  - The *true (now hidden)* sequences of states: $Q = q_1 \dots q_T$
  - The state transition matrix, A
  - The initial state values π
- However now we add in:
  - The set of possible things we can *observe*, $h_1, \dots, h_M$
  - The sequence that we observe $\mathrm{O} = o_1, \dots, o_T$
  - The probability of a state i *emitting* observation j: $b_{i,j}$

HMM Definition

- Therefore a HMM, λ, is a 5-tuple consisting of
  - The set of states: $S = \{s_1, \dots, s_K\}$
  - The set of observable values: $H = \{h_1, \dots, h_M\}$
  - The starting state probabilities: $\pi_i = P(q_1 = s_i)$
  - The state transition probabilities : $a_{ij} = P(q_{t+1} = s_j | q_t = s_i)$
    - For $1 >= i, j <= K$
  - The observation/emission probabilities:
    $$b_{ik} = P(o_t = h_k | q_t = s_i)$$
    - For $1 \leq i \leq K$ and $1 \leq k \leq M$
- $\lambda = (S, H, \pi, A, B)$ is the specification of a HMM

HMM Applications

- There are 3 main problems associated with HMMs
  - The evaluation problem
    - What's the probability of an observed sequence given the current HMM?, $P(O|\lambda)$
  - The decoding problem
    - Given an observed sequence and an HMM, what is the most probable sequence of (hidden) states?
$$\hat{Q} = argmax_Q P(Q|O, \lambda)$$
  - The learning problem
    - Given an observed sequence, find the HMM that maximizes the probability of generating this sequence.
$$\hat{\lambda} = argmax_\lambda P(O|\lambda)$$
- Let's look at each of them

The Evaluation Problem

HMMs

Evaluation Problem

- Given a HMM, λ, we want to know the probability of observing the sequence O:

$$P(O|\lambda)$$

- Recall from Markov models:

$$P(Q|\lambda) = \pi_{q_1} \prod_{i=1}^{T-1} a_{q_t,q_{t+1}}$$

- How does this have to be changed since now we don't observe the states directly?
  - We have to consider the possibility that we can from any of the K states
  - And take into consideration the emission probability.

Evaluation Problem

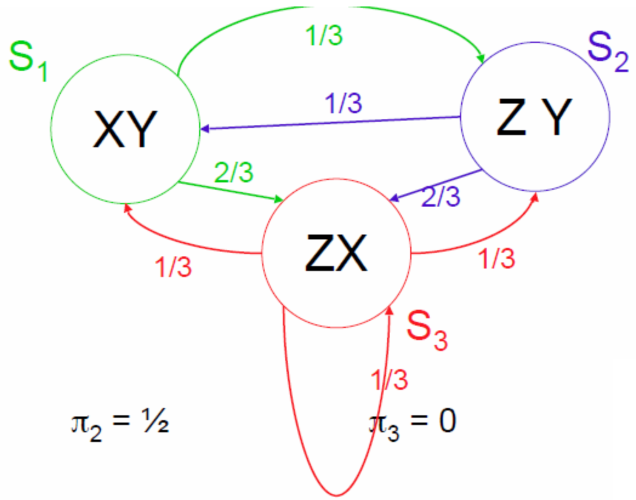
- $P(O|\lambda) = \sum_{k=1}^{K} a_k(T)$ where for $t = 1, \ldots, T$ compute for each $k = 1, \ldots, K$

$$a_k(t) = \begin{cases} b_{ko_1}\pi_k & t = 1 \\ b_{ko_t}\sum_i a_{ik}a_i(t-1) & otherwise \end{cases}$$

- This is most easily computed recursively, where the final probability is $P(O|\lambda) = \sum_{j=1}^{K} a_j(T)$
- Then just like with Markov models, we could do this computation for different models λ_i and use those to decide on which class the sequence belongs to.

Evaluation Example

- Suppose we are given the HMM to the right.
- What is the probability that it could have generated the observed sequence $O = XXX$?



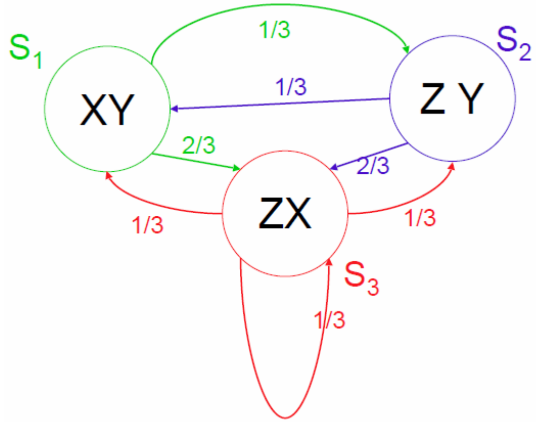
$N = 3$
$M = 3$
$\pi_1 = ½$ $\quad\pi_2 = ½$ $\quad\pi_3 = 0$

$a_{11} = 0$ $\quad a_{12} = ⅓$ $\quad a_{13} = ⅔$
$a_{12} = ⅓$ $\quad a_{22} = 0$ $\quad a_{13} = ⅔$
$a_{13} = ⅓$ $\quad a_{32} = ⅓$ $\quad a_{13} = ⅓$

$b_1 (X) = ½$ $\quad b_1 (Y) = ½$ $\quad b_1 (Z) = 0$
$b_2 (X) = 0$ $\quad b_2 (Y) = ½$ $\quad b_2 (Z) = ½$
$b_3 (X) = ½$ $\quad b_3 (Y) = 0$ $\quad b_3 (Z) = ½$

Evaluation Example

- Time 1 (observed X)
  - $a_1(1) \; = \; ¼, a_2(1) \; = \; 0, a_3(1) = 0$
- Time 2 (observed X)
  - $a_1(2) \; = 0, a_2(1) \; = \; 0, a_3(1) = \frac{1}{12}$
- Time 3 (observed X)
  - $a_{1(3)} = \; 0, a_{2(3)} = \frac{1}{72}, a_3(3) = \frac{1}{72}$
- $P(O|\lambda) = 0 + \frac{1}{72} + \frac{1}{72} \; = \frac{1}{36}$



HMMs for Classification

- We are given a HMM for each class with parameters λ_i
- Compute $P(\lambda_i|O)$ for all classes
  - Proportional to computing $P(O|\lambda_i)P(\lambda_i)$
  - Compute $P(O|\lambda_i)$ using forward/evaluation algorithm
  - Often assume $P(\lambda_i)$ is uniform
- Classifying input according to maximum

$$i^* = \mathrm{argmax}_{\mathrm{i}} P(\lambda_i|O)$$

The Decoding Problem

HMMs

The Decoding Problem

- Given a sequence of visible states O, the decoding problem is to find the most probably sequence of hidden states
  - We call this the *most probably path* (MPP): $P(Q|\lambda, O)$
- This is solved via the Viterbi algorithm
  - A Dynamic Programming (DP) approach

DP MPP Computation

- The general idea is:
  - At time t, for each state i
    - Find the path of length $t-1$, $\{q_1, \ldots q_{t-1}\}$, that has the highest probability of
      - Occurring, given observation chain $o_1, \ldots, o_{t-1}$
      - Ending up at time t with $q_t = s_i$
      - Emitting o_t at time t given s_i
    - Let $\delta_i(t)$ be that probability

The Viterbi Algorithm

- For $t = 1, \ldots, T$ compute for each state $k = 1, \ldots, K$:

$$\delta_k(t) = \begin{cases} b_{ko_1}\pi_k & t = 1 \\ b_{ko_t} max_i(\delta_i(t-1)a_{ik}) & otherwise \end{cases}$$

- And keep track of what which route we chose as you go:

$$\mathrm{q_t} = \operatorname*{argmax}_i \delta_i(t-1)a_{ik}$$

Decoding Example

- What's the most probably path Q for the observed sequence $O = (2,5,4)$
- Let's assume starting probabilities:

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

- And

$$a_{ij} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0.2 & 0.3 & 0.1 & 0.4 \\ 0.2 & 0.5 & 0.2 & 0.1 \\ 0.8 & 0.1 & 0 & 0.1 \end{bmatrix} \quad b_{jk} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0.3 & 0.4 & 0.1 & 0.2 \\ 0 & 0.1 & 0.1 & 0.7 & 0.1 \\ 0 & 0.5 & 0.2 & 0.1 & 0.2 \end{bmatrix}$$

Example

$$a_{ij} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0.2 & 0.3 & 0.1 & 0.4 \\ 0.2 & 0.5 & 0.2 & 0.1 \\ 0.8 & 0.1 & 0 & 0.1 \end{bmatrix} \quad b_{jk} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0.3 & 0.4 & 0.1 & 0.2 \\ 0 & 0.1 & 0.1 & 0.7 & 0.1 \\ 0 & 0.5 & 0.2 & 0.1 & 0.2 \end{bmatrix}$$

$$\boxed{O = (\mathbf{2}, 5, 4)} \quad \boxed{\pi = [\frac{1}{2}, \frac{1}{2}, 0, 0]}$$

$$\boxed{\delta_i(1) = \pi_i b_{io_1}}$$

$$\boxed{\delta_j(t) = b_{jo_t} \max_i \delta_i(t-1) a_{ij}}$$

- $t = 1$ (observed 2):
  - $\delta_1(1) = \frac{1}{2} b_{12} = 0$
  - $\delta_2(1) = \frac{1}{2} b_{22} = 0.15,$
  - $\delta_3(1) = 0, \delta_3(0) = 0$

$$\boxed{O = (2, \mathbf{5}, 4)}$$

- $t = 2$ (observed 5)
  - $\delta_1(2) = 0 \cdot \max(\dots) = 0$
    - Dead End
  - $\delta_2(2) = 0.2 \cdot \max(0, 0.15 * 0.3, 0, 0) = 0.009$
    - $\text{mpp}_2(2)$=(2)
  - $\delta_3(2) = 0.1 \cdot \max(0, 0.15 * 0.1, 0, 0) = 0.0015$
    - $\text{mpp}_3(2)$=(2)
  - $\delta_4(2) = 0.2 \cdot \max(0, 0.15 * 0.4, 0, 0) = 0.012$
    - $\text{mpp}_4(2)$=(2)

Example

$$a_{ij} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0.2 & 0.3 & 0.1 & 0.4 \\ 0.2 & 0.5 & 0.2 & 0.1 \\ 0.8 & 0.1 & 0 & 0.1 \end{bmatrix} \quad b_{jk} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0.3 & 0.4 & 0.1 & 0.2 \\ 0 & 0.1 & 0.1 & 0.7 & 0.1 \\ 0 & 0.5 & 0.2 & 0.1 & 0.2 \end{bmatrix}$$

$$\boxed{\delta_j(t) = b_{jo_t} \max_i \delta_i(t-1) a_{ij}}$$

$$\boxed{O = (2,5,\mathbf{4})}$$

- $t = 2$:
  - $\delta_1(2) = 0$, mpp$_1$(2)=N/A
  - $\delta_2(2) = 0.009$, mpp$_2$(2)=(2)
  - $\delta_3(2) = 0.0015$, mpp$_3$(2) = (2)
  - $\delta_4(2) = 0.012$, mpp$_4$(2) = (2)
- $t = 3$ (observed 4)
  - $\delta_1(3) = 0 \cdot \max(\dots) = 0$
    - mpp$_1$(3)=N/A
  - $\delta_2(3) = 0.1 \cdot \max(0, \mathbf{0.009} * \mathbf{0.3}, 0.0015 * 0.5, 0.012 * 0.1) = 0.00027$
    - mpp$_2$(3)=(2,2)
  - $\delta_3(3) = 0.7 \cdot \max(0, \mathbf{0.009} * \mathbf{0.1}, \ 0.0015 * 0.2, 0.012 * 0) = 0.00063$
    - mpp$_3$(3)=(2,2)
  - $\delta_4(3) = 0.1 \cdot \max(0, \mathbf{0.009} * \mathbf{0.4}, 0.0015 * 0.1, 0.012 * 0.1) = 0.00036$
    - 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.
- Where does the model come from?
- Maybe the application and/or prior knowledge allows us to “manually” create it.
- Or maybe we could *learn* it given some labeled data.
  - Given an observed sequence, O, we want to find a λ^* that maximizes the likelihood of creating the observed sequence:

$$\lambda^* = arg\max_{\lambda} P(O|\lambda)$$

EM for HMMs

- Given a current model, $\lambda(t)$ we can say stuff about our observation sequence O
- Given what we say about O can we updated our model $\lambda(t) \rightarrow \lambda(t+1)$ to better fit this?
- This sort of algorithm, where we iterative between making predictions using our current model, then update our model to better fit our predictions, is called *expectation maximization*.
- Technically we already saw one EM algorithm with k-means!

EM for HMMs

<u>Expectation</u>

- From the evaluation problem we know
  - $\delta_i(1) = \pi_i b_{io_1}$
  - $\delta_i(t+1) = b_{io_{t+1}} \sum_{j=1}^{N} \delta_j(t) a_{ji}$
- Similarly we can compute $\beta_i(t)$ (backwards evaluation?) as:
  - $\beta_i(T) = 1$
  - $\beta_i(t) = \sum_{j=1}^{N} \beta_j(t+1) a_{ij}\, b_{jo_{t+1}}$

EM for HMMs

Expectation

- Now lets use these values of δ and β to set up information related to state probabilities and transition probabilities.
- Let $\gamma_i(t)$ be the probability of being at state s_i at time t given our sequence: $\gamma_i(t) = P(q_t = s_i|O, \lambda)$
- We can use our forward and backwards estimations to compute this

$$\delta_i(t)\beta_i(t)$$

- And the normalize it by sum of the probabilities of being at each state:

$$\gamma_i(t) = P(q_t = s_i|O, \lambda) = \frac{\delta_i(t)\beta_i(t)}{\sum_{j=1}^{N} \delta_j(t)\beta_j(t)}$$

EM for HMMs

<u>Expectation</u>

- Similarly, let $\epsilon_{i,j}(t)$ be the probability of transitioning from state i to state j at time t:

$$\varepsilon_{ij}(t) = P(q_t = s_i, q_{t+1} = j|O, \lambda)$$

- We can again use the forwards and backwards estimations (and again normalize) but also incorporate the current state transition and the current state emission:
  - $\varepsilon_{ij}(t) = P(q_t = s_i, q_{t+1} = j|O, \lambda) = \frac{\delta_i(t) a_{ij} \beta_j(t+1) b_{j o_{t+1}}}{\sum_{k=1}^{N} \delta_k(t) \beta_k(t)}$

EM for HMMs

Maximization

- Now we need to maximize!
- Given $\gamma_i(t)$ and $\epsilon_{i,j}(t)$ this should be somewhat straight forward:
  - The initial state probabilities are just taken directly from $\gamma_i(t)$
  $$\pi_i = \gamma_i(1)$$
  - The state transition matrix values are take from $\epsilon_{i,j}(t)$ (summed over all times) but normalized by the probabilities of being at state i at any given time:
  $$a_{i,j} = \frac{\sum_{t=1}^{T-1} \epsilon_{i,j}(t)}{\sum_{t=1}^{T-1} \gamma_i(t)}$$
  - The emission matrix values basically for each state s_i, add up the probabilities of that state occurring whenever h_j is emitted, again normalized
  $$b_{ij} = \frac{\sum_{t=1}^{T}(o_t == j)\gamma_i(t)}{\sum_{t=1}^{T} \gamma_i(t)}$$

EM for HMMs

1. Get your observations $o_1 \ldots o_T$
2. Guess your first model $\lambda(0), k = 0$. Random?
3. Until convergence do steps 4 and 5
4. Do expectation via estimation
   - $\delta_i(t), \beta_i(t)$
   - $\gamma_i(t), \epsilon_{i,j}(t)$
5. Do maximization
   - $\pi_i = \gamma_i(1)$
   - $a_{i,j} = \frac{\sum_{t=1}^{T-1} \epsilon_{i,j}(t)}{\sum_{t=1}^{T-1} \gamma_i(t)}$
   - $b_{ij} = \frac{\sum_{t=1}^{T} (o_t == j)\gamma_i(t)}{\sum_{t=1}^{T} \gamma_i(t)}$

- This is known as the Baum-Welch algorithm

Example

from\to	LA	NY
LA	0.5	0.5
NY	0.5	0.5

where \ report	LA	NY	null
LA	0.4	0.1	0.5
NY	0.1	0.5	0.4

- Let’s try to find the HMM of a criminal traveling between LA and NY!
- The FBI doesn’t know where the criminal started his/her activity
  - Uniform probability of starting either place: $\pi = \left(\frac{1}{2}, \frac{1}{2}\right)$
- The FBI has no clue on if he/she will go from one place to another at any time
  - State Transition Matrix, A
- The FBI has some historic data based on where people said the criminal was and if he was actually there
  - This is our emissions matrix, B

where \ report	LA	NY	null
LA	0.4	0.1	0.5
NY	0.1	0.5	0.4

Example

from\to	LA	NY
LA	0.5	0.5
NY	0.5	0.5

- The FBI has been tracking reports over 5 time instances and observed the sequence:
  - $O = (-, LA, LA, NY)$
  - 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!):
  5. Can we update the model to make it better!? Learning Problem
- Let's use this example to make our model better!

where \ report	LA	NY	null
LA	0.4	0.1	0.5
NY	0.1	0.5	0.4

from\to	LA	NY
LA	0.5	0.5
NY	0.5	0.5

Example EM for HMM

$$\delta_i(1) = \pi_i b_{io_1}$$

$$\delta_i(t+1) = b_{jo_{t+1}} \sum_{j=1}^{N} \delta_j(t) a_{ji}$$

- $\pi = \left(\frac{1}{2}, \frac{1}{2}\right)$
- $O = (-, LA, LA, NY)$
- Iteration 1: Forward Estimation
  - $\delta_{LA}(1) = \pi_{LA} b_{LA}(-) = 0.25$
  - $\delta_{NY}(1) = \pi_{NY} b_{NY}(-) = 0.2$
  - $\delta_{LA}(2) = b_{LA}(LA)\left(\delta_{LA}(1)a_{LA,LA} + \delta_{NY}(1)a_{NY,LA}\right) = 0.4 * (0.25 * 0.5 + 0.2 * 0.5) = 0.09$
  - $\delta_{NY}(2) = b_{NY}(LA)\left(\delta_{LA}(1)a_{LA,NY} + \delta_{NY}(1)a_{NY,NY}\right) = 0.1 * (0.25 * 0.5 + 0.2 * 0.5) = 0.0225$
  - $\delta_{LA}(3) = b_{LA}(LA)\left(\delta_{LA}(2)a_{LA,LA} + \delta_{NY}(2)a_{NY,LA}\right) = 0.4 * (0.09 * 0.5 + 0.0225 * 0.5) = 0.0225$
  - $\delta_{NY}(3) = b_{NY}(LA)\left(\delta_{LA}(2)a_{LA,NY} + \delta_{NY}(2)a_{NY,NY}\right) = 0.1 * (0.09 * 0.5 + 0.0225 * 0.5) = 0.0056$
  - $\delta_{LA}(4) = b_{LA}(NY)\left(\delta_{LA}(3)a_{LA,LA} + \delta_{NY}(3)a_{NY,LA}\right) = 0.1 * (0.0225 * 0.5 + 0.0056 * 0.5) = 0.0014$
  - $\delta_{NY}(4) = b_{NY}(NY)\left(\delta_{LA}(3)a_{LA,NY} + \delta_{NY}(3)a_{NY,NY}\right) = 0.5 * (0.0225 * 0.5 + 0.0056 * 0.5) = 0.0070$

where \ report	LA	NY	null
LA	0.4	0.1	0.5
NY	0.1	0.5	0.4

Example EM for HMM

- $O = (-, LA, LA, NY)$
- Iteration 1: Backwards Procedure
  - $\beta_{LA}(4) = 1$
  - $\beta_{NY}(4) = 1$
  - $\beta_{LA}(3) = \left(\beta_{LA}(4)a_{LA,LA}b_{LA}(NY) + \beta_{NY}(4)a_{LA,NY}b_{NY}(NY)\right) = 1 * 0.5 * 0.1 + 1 * 0.5 * 0.5 = 0.3$
  - $\beta_{NY}(3) = \left(\beta_{LA}(4)a_{NY,LA}b_{LA}(NY) + \beta_{NY}(4)a_{NY,NY}b_{NY}(NY)\right) = 1 * 0.5 * 0.1 + 1 * 0.5 * 0.5 = 0.3$
  - $\beta_{LA}(2) = \left(\beta_{LA}(3)a_{LA,LA}b_{LA}(LA) + \beta_{NY}(3)a_{LA,NY}b_{NY}(LA)\right) = 0.3 * 0.5 * 0.4 + 0.3 * 0.5 * 0.1 = 0.075$
  - $\beta_{NY}(2) = \left(\beta_{LA}(3)a_{NY,LA}b_{LA}(LA) + \beta_{NY}(3)a_{NY,NY}b_{NY}(LA)\right) = 0.3 * 0.5 * 0.4 + 0.3 * 0.5 * 0.1 = 0.075$
  - $\beta_{LA}(1) = \left(\beta_{LA}(2)a_{LA,LA}b_{LA}(LA) + \beta_{NY}(2)a_{LA,NY}b_{NY}(LA)\right) = 0.075 * 0.5 * 0.4 + 0.075 * 0.5 * 0.1 = 0.0187$
  - $\beta_{NY}(1) = \left(\beta_{LA}(2)a_{NY,LA}b_{LA}(LA) + \beta_{NY}(2)a_{NY,NY}b_{NY}(LA)\right) = 0.075 * 0.5 * 0.4 + 0.075 * 0.5 * 0.1 = 0.0187$

$$\boxed{\begin{aligned}\beta_i(T) &= 1 \\ \beta_i(t) &= \sum_{j=1}^{N} \beta_j(t+1)a_{ij}\, b_{jo_{t+1}}\end{aligned}}$$

Example: EM for HMM

$$\gamma_i(t) = P(q_t = s_i | O, \lambda) = \frac{\delta_i(t)\beta_i(t)}{\sum_{j=1}^{N} \delta_j(t)\beta_j(t)}$$

- O=(-,LA,LA,NY)
- Iteration 1: Gamma
- $\gamma_{LA}(1) = \frac{\delta_{LA}(1)\beta_{LA}(1)}{\delta_{LA}(1)\beta_{LA}(1)+\delta_{NY}(1)\beta_{NY}(1)} = \frac{0.25*0.0187}{(0.25*0.0187+0.2*0.0187)} = 0.5556$
- $\gamma_{NY}(1) = \frac{\delta_{NY}(1)\beta_{NY}(1)}{\delta_{LA}(1)\beta_{LA}(1)+\delta_{NY}(1)\beta_{NY}(1)} = \frac{0.2*0.0187}{(0.25*0.0187+0.22*0.0187)} = 0.4444$
- $\gamma_{LA}(2) = \frac{\delta_{LA}(2)\beta_{LA}(2)}{\delta_{LA}(2)\beta_{LA}(2)+\delta_{NY}(2)\beta_{NY}(2)} = \frac{0.09*0.075}{(0.09*0.075+0.0225*0.075)} = 0.8$
- $\gamma_{NY}(2) = \frac{\delta_{NY}(2)\beta_{NY}(2)}{\delta_{LA}(2)\beta_{LA}(2)+\delta_{NY}(2)\beta_{NY}(2)} = \frac{0.0225*0.075}{(0.09*0.075+0.0225*0.075)} = 0.2$
- $\gamma_{LA}(3) = \frac{\delta_{LA}(3)\beta_{LA}(3)}{\delta_{LA}(3)\beta_{LA}(3)+\delta_{NY}(3)\beta_{NY}(3)} = \frac{0.0225*0.3}{(0.0225*0.3+0.0056*0.3)} = 0.8$
- $\gamma_{NY}(3) = \frac{\delta_{NY}(3)\beta_{NY}(3)}{\delta_{LA}(3)\beta_{LA}(3)+\delta_{NY}(3)\beta_{NY}(3)} = \frac{0.0056*0.3}{(0.0225*0.3+0.0056*0.3)} = 0.2$
- $\gamma_{LA}(4) = \frac{\delta_{LA}(4)\beta_{LA}(4)}{\delta_{LA}(4)\beta_{LA}(4)+\delta_{NY}(4)\beta_{NY}(4)} = \frac{0.0014*1}{(0.0014*1+0.0070*1)} = 0.1667$
- $\gamma_{NY}(4) = \frac{\delta_{NY}(4)\beta_{NY}(4)}{\delta_{LA}(4)\beta_{LA}(4)+\delta_{NY}(4)\beta_{NY}(4)} = \frac{0.0070*1}{(0.0014*1+0.0070*1)} = 0.8333$

Example: EM for HMM

$$\varepsilon_{ij}(t) = P(q_t = s_i, q_{t+1} = j | O, \lambda) = \frac{\delta_i(t) a_{ij} \beta_j(t+1) b_{j o_{t+1}}}{\sum_{k=1}^{N} \delta_k(t) \beta_k(t)}$$

- O=(-,LA,LA,NY)
- Iteration 1: Epsilon
- $\varepsilon_{LA,LA}(1) = \frac{\delta_{LA}(1) a_{LA,LA} \beta_{LA}(2) b_{LA}(LA)}{(\delta_{LA}(1)\beta_{LA}(1)+\delta_{NY}(1)\beta_{NY}(1))} = \frac{0.25*0.5*0.075*0.4}{(0.25*0.0187+0.20*0.0187)} = 0.4444$
- $\varepsilon_{LA,NY}(1) = \frac{\delta_{LA}(1) a_{LA,NY} \beta_{NY}(2) b_{NY}(LA)}{(\delta_{LA}(1)\beta_{LA}(1)+\delta_{NY}(1)\beta_{NY}(1))} = \frac{0.25*0.5*0.075*0.1}{(0.25*0.0187+0.20*0.0187)} = 0.1111$
- $\varepsilon_{NY,LA}(1) = \frac{\delta_{NY}(1) a_{NY,LA} \beta_{LA}(2) b_{LA}(LA)}{(\delta_{LA}(1)\beta_{LA}(1)+\delta_{NY}(1)\beta_{NY}(1))} = \frac{0.20*0.5*0.075*0.4}{(0.25*0.0187+0.20*0.0187)} = 0.3556$
- $\varepsilon_{NY,NY}(1) = \frac{\delta_{NY}(1) a_{NY,NY} \beta_{NY}(2) b_{NY}(LA)}{(\delta_{LA}(1)\beta_{LA}(1)+\delta_{NY}(1)\beta_{NY}(1))} = \frac{0.20*0.5*0.075*0.1}{(0.25*0.0187+0.20*0.0187)} = 0.0891$
- $\varepsilon_{LA,LA}(2) = \frac{\delta_{LA}(2) a_{LA,LA} \beta_{LA}(3) b_{LA}(LA)}{(\delta_{LA}(2)\beta_{LA}(2)+\delta_{NY}(2)\beta_{NY}(2))} = \frac{0.09*0.5*0.3*0.4}{(0.09*0.075+0.0225*0.075)} = 0.64$
- $\varepsilon_{LA,NY}(2) = \frac{\delta_{LA}(2) a_{LA,NY} \beta_{NY}(3) b_{NY}(LA)}{(\delta_{LA}(2)\beta_{LA}(2)+\delta_{NY}(2)\beta_{NY}(2))} = \frac{0.09*0.5*0.3*0.1}{(0.09*0.075+0.0225*0.075)} = 0.16$
- $\varepsilon_{NY,LA}(2) = \frac{\delta_{NY}(2) a_{NY,LA} \beta_{LA}(3) b_{LA}(LA)}{(\delta_{LA}(2)\beta_{LA}(2)+\delta_{NY}(2)\beta_{NY}(2))} = \frac{0.0025*0.5*0.3*0.4}{(0.09*0.075+0.0225*0.075)} = 0.16$
- $\varepsilon_{NY,NY}(2) = \frac{\delta_{NY}(2) a_{NY,NY} \beta_{NY}(3) b_{NY}(LA)}{(\delta_{LA}(2)\beta_{LA}(2)+\delta_{NY}(2)\beta_{NY}(2))} = \frac{0.0025*0.5*0.3*0.1}{(0.09*0.075+0.0225*0.075)} = 0.04$

where \ report	LA	NY	null
LA	0.4	0.1	0.5
NY	0.1	0.5	0.4

Example: EM for HMM

$$\varepsilon_{ij}(t) = P(q_t = s_i, q_{t+1} = j | O, \lambda) = \frac{\delta_i(t) a_{ij} \beta_j(t+1) b_{j o_{t+1}}}{\sum_{k=1}^{N} \delta_k(t) \beta_k(t)}$$

- $O = (-, LA, LA, NY)$
- Iteration 1: Epsilon
- $\varepsilon_{LA,LA}(3) = \frac{\delta_{LA}(3) a_{LA,LA} \beta_{LA}(4) b_{LA}(NY)}{(\delta_{LA}(3)\beta_{LA}(3)+\delta_{NY}(3)\beta_{NY}(3))} = \frac{0.0225*0.5*1*0.1}{(0.0225*0.3+0.0056*0.3)} = 0.1333$
- $\varepsilon_{LA,NY}(3) = \frac{\delta_{LA}(3) a_{LA,NY} \beta_{NY}(4) b_{NY}(NY)}{(\delta_{LA}(3)\beta_{LA}(3)+\delta_{NY}(3)\beta_{NY}(3))} = \frac{0.0225*0.5*1*0.5}{(0.0225*0.3+0.0056*0.3)} = 06667$
- $\varepsilon_{NY,LA}(3) = \frac{\delta_{NY}(3) a_{NY,LA} \beta_{LA}(4) b_{LA}(NY)}{(\delta_{LA}(3)\beta_{LA}(3)+\delta_{NY}(3)\beta_{NY}(3))} = \frac{0.0056*0.5*1*0.1}{(0.0225*0.3+0.0056*0.3)} = 0.0333$
- $\varepsilon_{NY,NY}(3) = \frac{\delta_{NY}(3) a_{NY,NY} \beta_{NY}(4) b_{NY}(NY)}{(\delta_{LA}(3)\beta_{LA}(3)+\delta_{NY}(3)\beta_{NY}(3))} = \frac{0.0056*0.5*1*0.5}{(0.0225*0.3+0.0056*0.3)} = 0.1667$

Example EM for HMM

$$a_{ij} = \frac{\sum_{t=1}^{T-1} \epsilon_{ij}(t)}{\sum_{t=1}^{T-1} \gamma_i(t)}$$
$$\pi_i = \gamma_i(1)$$

- Iteration 1: Maximization
  - $\pi_{LA} = \gamma_{LA}(1) = 0.5556$
  - $\pi_{NY} = \gamma_{NY}(1) = 0.4444$
  - $a_{LA,LA} = \frac{\sum_{t=1}^{T-1} \varepsilon_{LA,LA}(t)}{\sum_{t=1}^{T-1} \gamma_{LA}(t)} = \frac{0.4444+0.64\ +0.1333}{0.5556+0.8+0.8} = 0.5649$
  - $a_{LA,NY} = \frac{\sum_{t=1}^{T-1} \varepsilon_{LA,NY}(t)}{\sum_{t=1}^{T-1} \gamma_{LA}(t)} = 0.4357$
  - $a_{NY,LA} = \frac{\sum_{t=1}^{T-1} \varepsilon_{NY,LA}(t)}{\sum_{t=1}^{T-1} \gamma_{NY}(t)} = 0.65$
  - $a_{NY,NY} = \frac{\sum_{t=1}^{T-1} \varepsilon_{NY,NY}(t)}{\sum_{t=1}^{T-1} \gamma_{NY}(t)} = 0.35$

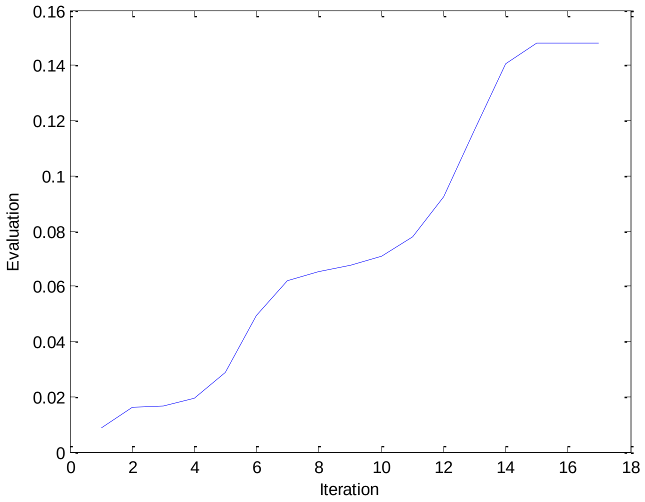
Example EM for HMM

$$b_{ij} = \frac{\sum_{t=1}^{T}(o_t == j)\gamma_i(t)}{\sum_{t=1}^{T}\gamma_i(t)}$$

- Iteration 1: Maximization
  - $b_{LA}(LA) = \frac{\sum_{t=1}^{T}(o_t==LA)\gamma_{LA}(t)}{\sum_{t=1}^{T}\gamma_{LA}(t)} = \frac{(0+0.8+0.8+0)}{(0.5556+0.8+0.8+0.1667)} = 0.689$
  - $b_{LA}(NY) = \frac{\sum_{t=1}^{T}(o_t==NY)\gamma_{LA}(t)}{\sum_{t=1}^{T}\gamma_{LA}(t)} = 0.0718$
  - $b_{LA}(-) = \frac{\sum_{t=1}^{T}(o_t==-)\gamma_{LA}(t)}{\sum_{t=1}^{T}\gamma_{LA}(t)} = 0.2392$
  - $b_{NY}(LA) = \frac{\sum_{t=1}^{T}(o_t==LA)\gamma_{NY}(t)}{\sum_{t=1}^{T}\gamma_{NY}(t)} = 0.2384$
  - $b_{NY}(NY) = \frac{\sum_{t=1}^{T}(o_t==NY)\gamma_{NY}(t)}{\sum_{t=1}^{T}\gamma_{NY}(t)} = 0.4967$
  - $b_{NY}(-) = \frac{\sum_{t=1}^{T}(o_t==-)\gamma_{NY}(t)}{\sum_{t=1}^{T}\gamma_{NY}(t)} == 0.2649$

Example EM for HMM

- Sanity Check
- Let's evaluate using our original HMM
  - $P(O|\lambda(1)) = 0.0084$
- Let's evaluate using our (slightly) updated HMM
  - $P(O|\lambda(2)) = 0.0160$
- After 17 iterations
  - $P(O|\lambda(17)) = 0.1481$
  - $\pi = [0,1]^T$
  - $A = \begin{bmatrix} 1 & 0 \\ 1 & 0 \end{bmatrix}$
  - $B = \begin{bmatrix} 0.6667 & 0.3333 & 0 \\ 0 & 0 & 1 \end{bmatrix}$
  - Does anything look odd with this?
  - How can we deal with it



Random HMM

- How good (relatively) is this HMM at generating this sequence of observations?
- Consider a random HMM for this problem

$$\pi_i = \frac{1}{N}, a_{ij} = \frac{1}{N}, b_{ik} = \frac{1}{M}$$

- And since we have 2 states and a chain of length 4 we have $2^4 = 16$ possible paths
- Recall the *exhaustive* equation

$$P(O|\lambda) = \sum_{r=1}^{r_{max}} \prod_{t=1}^{T} P(o_t|q_t)P(q_t|q_{t-1})$$

Example

- In a random system $P(o_t|q_t) = \frac{1}{M}$ for all t and $P(q_t|q_{t-1}) = \frac{1}{N}$ for all t
- So for a particular path of length T we have

$$P(Q_r|\lambda, O) = \prod_{t=1}^{T} \frac{1}{M}\frac{1}{N} = (MN)^{-T}$$

- For this example we then get $P(Q_r|\lambda, O) = (3 \cdot 2)^{-4}$
- And we have this for all 2^4 possible paths so

$$P(O|\lambda) = 2^4(3 \cdot 2)^{-4} = 0.0123$$

- Even simpler, if a HMM is random then for all t, $o_t = \frac{1}{M}$ and $P(O|\lambda) = \prod_{t=1}^{T} o_t = \left(\frac{1}{M}\right)^T$ which for this example is $\left(\frac{1}{3}\right)^4 = 0.0123$

Continuous HMM

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

$$P(o_t|q_t = s_i)$$

- We'll still have discrete states, $\{s_1, \ldots, s_K\}$
- However, now each state has a probability of emitting a values according to some distribution.

Continuous HMM

- Again, a common distribution is Gaussian:
  - Each state s_i's emission distribution is parameterized by (u_i, σ_i)

$$P(o_t | q_t = s_i) \propto \frac{1}{\sqrt{2\pi\sigma_i^2}} e^{-\frac{(o_t - \mu_i)^2}{2\sigma^2}}$$

  - Or we can do a multi-variate version
- Either we're given those parameters or we learn them via an EM algorithm.
- Then we can use this $P(o_t | q_t = s_i)$ for our evaluation/classification and decoding problems.

The Good and Bad

- Bad
  - There are lots of local minima
- Good news
  - The local minima are usually adequate models of the data
- Other things:
  - EM doesn’t estimate the number of states. That must be given
    - Trade-off between too few (inadequately modeling the structure) and too many (fitting the noise)
  - Often HMMs are forced to have some links with zero probability. This is done by setting $a_{ij} = 0$ in initial estimate $\lambda(0)$

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