Hidden Markov Models (Part II)
Reminders

• Practice Problems: Exam 2
  – Out: Fri, Mar. 24

• Exam 2
  – Thu, Mar. 30, 6:30pm – 8:30pm

• Homework 7: Hidden Markov Models
  – Out: Fri, Mar. 31
  – Due: Mon, Apr. 10 at 11:59pm
SUPERVISED LEARNING FOR HMMS
Recipe for Closed-form MLE

1. Assume data was generated i.i.d. from some model (i.e. write the generative story)
   \[ x^{(i)} \sim p(x|\theta) \]
2. Write log-likelihood
   \[ \ell(\theta) = \log p(x^{(1)}|\theta) + \ldots + \log p(x^{(N)}|\theta) \]
3. Compute partial derivatives (i.e. gradient)
   \[ \frac{\partial \ell(\theta)}{\partial \theta_1} = \ldots \]
   \[ \frac{\partial \ell(\theta)}{\partial \theta_2} = \ldots \]
   \[ \ldots \]
   \[ \frac{\partial \ell(\theta)}{\partial \theta_M} = \ldots \]
4. Set derivatives to zero and solve for \( \theta \)
   \[ \frac{\partial \ell(\theta)}{\partial \theta_m} = 0 \text{ for all } m \in \{1, \ldots, M\} \]
   \[ \theta_{\text{MLE}} = \text{solution to system of } M \text{ equations and } M \text{ variables} \]
5. Compute the second derivative and check that \( \ell(\theta) \) is concave down at \( \theta_{\text{MLE}} \)
MLE of Categorical Distribution

1. Suppose we have a **dataset** obtained by repeatedly rolling a $M$-sided (weighted) die $N$ times. That is, we have data

$$\mathcal{D} = \{x^{(i)}\}_{i=1}^{N}$$

where $x^{(i)} \in \{1, \ldots, M\}$ and $x^{(i)} \sim \text{Categorical}(\phi)$.

2. A random variable is **Categorical** written $X \sim \text{Categorical}(\phi)$ iff

$$P(X = x) = p(x; \phi) = \phi_x$$

where $x \in \{1, \ldots, M\}$ and $\sum_{m=1}^{M} \phi_m = 1$. The **log-likelihood** of the data becomes:

$$\ell(\phi) = \sum_{i=1}^{N} \log \phi_{x^{(i)}} \text{ s.t. } \sum_{m=1}^{M} \phi_m = 1$$

3. Solving this constrained optimization problem yields the **maximum likelihood estimator** (MLE):

$$\phi_{m}^{\text{MLE}} = \frac{N_{x=m}}{N} = \frac{\sum_{i=1}^{N} \mathbb{I}(x^{(i)} = m)}{N}$$
Hidden Markov Model (v1)

HMM Parameters:

Emission matrix, $A$, where $P(X_t = k|Y_t = j) = A_{j,k}, \forall t, k$

Transition matrix, $B$, where $P(Y_t = k|Y_{t-1} = j) = B_{j,k}, \forall t, k$

Initial probs, $C$, where $P(Y_1 = k) = C_k, \forall k$

$$P(X, Y) = P(Y_1) \left( \prod_{t=1}^{T} P(X_t|Y_t) \right) \left( \prod_{t=2}^{T} P(Y_t|Y_{t-1}) \right)$$
Hidden Markov Model (v1)

HMM Parameters:

- Emission matrix, $A$, where $P(X_t = k | Y_t = j) = A_{j,k}, \forall t, k$
- Transition matrix, $B$, where $P(Y_t = k | Y_{t-1} = j) = B_{j,k}, \forall t, k$
- Initial probs, $C$, where $P(Y_1 = k) = C_k, \forall k$

Joint Distribution (probability mass function):

$$p(x, y) = p(y_1, C) \left( \prod_{t=1}^{T} p(x_t | y_t, A) \right) \left( \prod_{t=2}^{T} p(y_t | y_{t-1}, B) \right)$$

$$= C_{y_1} \left( \prod_{t=1}^{T} A_{y_t, x_t} \right) \left( \prod_{t=2}^{T} B_{y_{t-1}, y_t} \right)$$
Learning an HMM decomposes into solving two (independent) Mixture Models

Data: \( \mathcal{D} = \{(\mathbf{x}^{(i)}, \mathbf{y}^{(i)})\}_{i=1}^N \) where \( \mathbf{x} = [x_1, \ldots, x_T]^T \) and \( \mathbf{y} = [y_1, \ldots, y_T]^T \)

Likelihood:

\[
\ell(A, B, C) = \sum_{i=1}^N \log p(\mathbf{x}^{(i)}, \mathbf{y}^{(i)} | A, B, C)
\]

\[
= \sum_{i=1}^N \left[ \log p(y_1^{(i)} | C) + \sum_{t=2}^T \log p(y_t^{(i)} | y_{t-1}^{(i)}, B) \right] + \sum_{t=1}^T \log p(x_t^{(i)} | y_t^{(i)}, A)
\]

MLE:

\( \hat{A}, \hat{B}, \hat{C} = \arg\max_{A,B,C} \ell(A, B, C) \)

\( \Rightarrow \hat{C} = \arg\max_C \sum_{i=1}^N \log p(y_1^{(i)} | C) \)

\( \hat{B} = \arg\max_B \sum_{i=1}^N \sum_{t=2}^T \log p(y_t^{(i)} | y_{t-1}^{(i)}, B) \)

\( \hat{A} = \arg\max_A \sum_{i=1}^N \sum_{t=1}^T \log p(x_t^{(i)} | y_t^{(i)}, A) \)

We can solve the above in closed form, which yields...

\( \hat{C}_k = \frac{\#(y_1^{(i)} = k)}{N}, \forall k \)

\( \hat{B}_{j,k} = \frac{\#(y_t^{(i)} = k \text{ and } y_{t-1}^{(i)} = j)}{\#(y_{t-1}^{(i)} = j)}, \forall j, k \)

\( \hat{A}_{j,k} = \frac{\#(x_t^{(i)} = k \text{ and } y_t^{(i)} = j)}{\#(y_t^{(i)} = j)}, \forall j, k \)
HMM (two ways)

HMM (v1):

\[
P(X, Y) = P(Y_1) \left( \prod_{t=1}^{T} P(X_t | Y_t) \right) \left( \prod_{t=2}^{T} p(Y_t | Y_{t-1}) \right)
\]

HMM (v2):

\[
P(X, Y | Y_0) = \prod_{t=1}^{T} P(X_t | Y_t)p(Y_t | Y_{t-1})
\]
Hidden Markov Model (v2)

HMM Parameters:

- **Emission matrix, A**, where
  \[ P(X_k = w | Y_k = t) = A_{t,w}, \forall k \]

- **Transition matrix, B**, where
  \[ P(Y_k = t | Y_{k-1} = s) = B_{s,t}, \forall k \]

For notational convenience, we fold the initial probabilities \( C \) into the transition matrix \( B \) by our assumption.
Hidden Markov Model (v2)

HMM Parameters:

Emission matrix, $A$, where $P(X_t = k | Y_t = j) = A_{j,k}, \forall t, k$

Transition matrix, $B$, where $P(Y_t = k | Y_{t-1} = j) = B_{j,k}, \forall t, k$

Assumption: $y_0 = \text{START}$

Generative Story:

$Y_t \sim \text{Multinomial}(B_{Y_{t-1}}) \ \forall t$

$X_t \sim \text{Multinomial}(A_{Y_t}) \ \forall t$

For notational convenience, we fold the initial probabilities $C$ into the transition matrix $B$ by our assumption.
Hidden Markov Model (v2)

Joint Distribution (probability mass function):

\[ y_0 = \text{START} \]

\[
p(x, y|y_0) = \prod_{t=1}^{T} p(x_t|y_t)p(y_t|y_{t-1})
\]

\[
= \prod_{t=1}^{T} A_{y_t,x_t} B_{y_{t-1},y_t}
\]
Supervised Learning for HMM (v2)

Learning an HMM decomposes into solving two (independent) Mixture Models

Data: \( D = \{(x^{(i)}, y^{(i)})\}^N_{i=1} \) where \( x = [x_1, \ldots, x_T]^T \) and \( y = [y_1, \ldots, y_T]^T \)

We assume \( y_0^{(i)} = \text{START} \) for all \( i \)

Likelihood:

\[
\ell(A, B) = \sum_{i=1}^{N} \log p(x^{(i)}, y^{(i)} | A, B)
\]

\[
= \sum_{i=1}^{N} \left[ \sum_{t=1}^{T} \log p(y_t^{(i)} | y_{t-1}^{(i)}, B) + \log p(x_t^{(i)} | y_t^{(i)}, A) \right]
\]

MLE:

\[
\hat{A}, \hat{B} = \arg\max_{A, B, C} \ell(A, B)
\]

\[
\Rightarrow \hat{B} = \arg\max_B \sum_{i=1}^{N} \sum_{t=1}^{T} \log p(y_t^{(i)} | y_{t-1}^{(i)}, B)
\]

\[
\hat{A} = \arg\max_A \sum_{i=1}^{N} \sum_{t=1}^{T} \log p(x_t^{(i)} | y_t^{(i)}, A)
\]

We can solve the above in closed form, which yields...

\[
\hat{B}_{j,k} = \frac{\#(y_t^{(i)} = k \text{ and } y_{t-1}^{(i)} = j)}{\#(y_t^{(i)} = j)}, \forall j, k
\]

\[
\hat{A}_{j,k} = \frac{\#(x_t^{(i)} = k \text{ and } y_t^{(i)} = j)}{\#(y_t^{(i)} = j)}, \forall j, k
\]
BACKGROUND: MESSAGE PASSING
Great Ideas in ML: Message Passing

Count the soldiers

there's 1 of me

1 before you
2 before you
3 before you
4 before you
5 before you
5 behind you
4 behind you
3 behind you
2 behind you
1 behind you

adapted from MacKay (2003) textbook
Great Ideas in ML: Message Passing

Count the soldiers

Belief: Must be
2 + 1 + 3 = 6 of us

there's 1 of me

2 before you

only see my incoming messages

3 behind you

adapted from MacKay (2003) textbook
Great Ideas in ML: Message Passing

**Count the soldiers**

- there's 1 of me
- 1 before you
- only see my incoming messages
- 4 behind you

Belief: Must be $1 + 1 + 4 = 6$ of us

Belief: Must be $1 + 3 = 6$ of us

adapted from MacKay (2003) textbook
Great Ideas in ML: Message Passing

Each soldier receives reports from all branches of tree

adapted from MacKay (2003) textbook
Great Ideas in ML: Message Passing

Each soldier receives reports from all branches of tree

adapted from MacKay (2003) textbook
Great Ideas in ML: Message Passing

Each soldier receives reports from all branches of the tree

adapted from MacKay (2003) textbook
Great Ideas in ML: Message Passing

Each soldier receives reports from all branches of tree

adapted from MacKay (2003) textbook
Great Ideas in ML: Message Passing

Each soldier receives reports from all branches of tree

Belief:
Must be 14 of us

 wouldn't work correctly with a 'loopy' (cyclic) graph

adapted from MacKay (2003) textbook
INFERENCE FOR HMMS
Inference

Question: True or False: The joint probability of the observations and the hidden states in an HMM is given by:

\[ P(X = x, Y = y) = C_{y_1} \prod_{t=1}^{T} A_{y_t, x_t} \prod_{t=1}^{T-1} B_{y_t, y_{t+1}} \]

Recall:

Emission matrix, \( A \), where \( P(X_t = k | Y_t = j) = A_{j,k}, \forall t, k \)

Transition matrix, \( B \), where \( P(Y_t = k | Y_{t-1} = j) = B_{j,k}, \forall t, k \)

Initial probs, \( C \), where \( P(Y_1 = k) = C_k, \forall k \)
Inference

Question: True or False: The probability of the observations in an HMM is given by:

\[ P(X = x) = \prod_{t=1}^{T} A_{x_t, x_{t-1}} \]

Recall:

Emission matrix, A, where \( P(X_t = k | Y_t = j) = A_{j,k}, \forall t, k \)

Transition matrix, B, where \( P(Y_t = k | Y_{t-1} = j) = B_{j,k}, \forall t, k \)

Initial probs, C, where \( P(Y_1 = k) = C_k, \forall k \)
Inference for HMMs

Whiteboard

– Three Inference Problems for an HMM

1. Evaluation: Compute the probability of a given sequence of observations
2. Viterbi Decoding: Find the most-likely sequence of hidden states, given a sequence of observations
3. Marginals: Compute the marginal distribution for a hidden state, given a sequence of observations
THE SEARCH SPACE FOR FORWARD-BACKWARD
Dataset for Supervised Part-of-Speech (POS) Tagging

Data: \[ \mathcal{D} = \{ (x^{(n)}, y^{(n)}) \}_{n=1}^{N} \]

Sample 1:
- time
- flies
- like
- an
- arrow

Sample 2:
- time
- flies
- like
- an
- arrow

Sample 3:
- flies
- fly
- with
- their
- wings

Sample 4:
- with
- time
- you
- will
- see
Example: HMM for POS Tagging

A Hidden Markov Model (HMM) provides a joint distribution over the sentence/tags with an assumption of dependence between adjacent tags.

\[
p(n, v, p, d, n, \text{time, flies, like, an, arrow}) = (0.3 \times 0.8 \times 0.2 \times 0.5 \times \ldots)
\]
Example: HMM for POS Tagging

Could be verb or noun

Could be adjective or verb

Could be noun or verb
Inference for HMMs

Whiteboard

– Brute Force Evaluation
– Forward-backward search space
THE FORWARD-BACKWARD ALGORITHM
How is efficient computation even possible?

• The short answer is **dynamic programming**!

• The key idea is this:
  – We first come up with a **recursive definition** for the quantity we want to compute
  – We then observe that many of the recursive intermediate terms are **reused** across timesteps and tags
  – We then perform **bottom-up dynamic programming** by running the recursion in reverse, **storing the intermediate quantities** along the way!

• This enables us to search the **exponentially large** space in **polynomial time**!
Inference for HMMs

Whiteboard

– Forward-backward algorithm
  (edge weights version)
Forward-Backward Algorithm

Definitions
\[ \alpha_t(k) \triangleq p(x_1, \ldots, x_t, y_t = k) \]
\[ \beta_t(k) \triangleq p(x_{t+1}, \ldots, x_T \mid y_t = k) \]

Assume
\[ y_0 = \text{START} \]
\[ y_{T+1} = \text{END} \]

1. Initialize
\[ \alpha_0(\text{START}) = 1 \]
\[ \beta_{T+1}(\text{END}) = 1 \]
\[ \alpha_0(k) = 0, \ \forall k \neq \text{START} \]
\[ \beta_{T+1}(k) = 0, \ \forall k \neq \text{END} \]

2. Forward Algorithm
   for \( t = 1, \ldots, T + 1 \):
   for \( k = 1, \ldots, K \):
   \[ \alpha_t(k) = \sum_{j=1}^{K} p(x_t \mid y_t = k) \alpha_{t-1}(j) p(y_t = k \mid y_{t-1} = j) \]

3. Backward Algorithm
   for \( t = T, \ldots, 0 \):
   for \( k = 1, \ldots, K \):
   \[ \beta_t(k) = \sum_{j=1}^{K} p(x_{t+1} \mid y_{t+1} = j) \beta_{t+1}(j) p(y_{t+1} = j \mid y_t = k) \]

4. Evaluation
\[ p(x) = \alpha_{T+1}(\text{END}) \]

5. Marginals
\[ p(y_t = k \mid x) = \frac{\alpha_t(k) \beta_t(k)}{p(x)} \]
Forward-Backward Algorithm

1. Initialize
\[
\alpha_0(\text{START}) = 1 \\
\alpha_0(k) = 0, \; \forall k \neq \text{START} \\
\beta_{T+1}(\text{END}) = 1 \\
\beta_{T+1}(k) = 0, \; \forall k \neq \text{END}
\]

2. Forward Algorithm
\[
\text{for } t = 1, \ldots, T + 1: \\
\quad \text{for } k = 1, \ldots, K: \\
\quad \quad \alpha_t(k) = \sum_{j=1}^{K} p(x_t \mid y_t = k) \alpha_{t-1}(j)p(y_t = k \mid y_{t-1} = j)
\]

3. Backward Algorithm
\[
\text{for } t = T, \ldots, 0: \\
\quad \text{for } k = 1, \ldots, K: \\
\quad \quad \beta_t(k) = \sum_{j=1}^{K} p(x_{t+1} \mid y_{t+1} = j) \beta_{t+1}(j)p(y_{t+1} = j \mid y_t = k)
\]

4. Evaluation
\[
p(x) = \alpha_{T+1}(\text{END})
\]

5. Marginals
\[
p(y_t = k \mid x) = \frac{\alpha_t(k)\beta_t(k)}{p(x)}
\]
EXAMPLE: FORWARD-BACKWARD ON THREE WORDS
Forward-Backward Algorithm

Could be verb or noun

Could be adjective or verb

Could be noun or verb
Forward-Backward Algorithm

$Y_1 \xrightarrow{X_1} Y_2 \xrightarrow{X_2} Y_3 \xrightarrow{X_3} Y_4$

$X_1$ find

$X_2$ preferred

$X_3$ tags
Forward-Backward Algorithm

- Let’s show the possible values for each variable
Let’s show the possible values for each variable
• Let’s show the possible values for each variable
• One possible assignment
Let’s show the possible values for each variable
One possible assignment
And what the 7 transition / emission factors think of it …
• Let’s show the possible values for each variable
• One possible assignment
• And what the 7 transition / emission factors think of it …
Viterbi Algorithm: Most Probable Assignment

- So $p(v a n) = (1/Z) \times \text{product of 7 numbers}$
- Numbers associated with edges and nodes of path
- Most probable assignment = path with highest product
Viterbi Algorithm: Most Probable Assignment

- So \( p(v \ a \ n) = (1/Z) \times \) product weight of one path
Forward-Backward Algorithm: Finds Marginals

- So $p(v a n) = (1/Z) \times \text{product weight of one path}$
- Marginal probability $p(Y_2 = a) = (1/Z) \times \text{total weight of all paths through } a$
Forward-Backward Algorithm: Finds Marginals

- So \( p(v \ a \ n) = \frac{1}{Z} \) * product weight of one path
- Marginal probability \( p(Y_2 = n) = \frac{1}{Z} \) * total weight of all paths through \( \nabla n \)
Forward-Backward Algorithm: Finds Marginals

- So $p(v \ a \ n) = (1/Z) \times$ product weight of one path
- Marginal probability $p(Y_2 = v) = (1/Z) \times$ total weight of all paths through $v$
Forward-Backward Algorithm: Finds Marginals

- So $p(v \ a \ n) = (1/Z) *$ product weight of one path
- Marginal probability $p(Y_2 = n) = (1/Z) *$ total weight of all paths through $n$
Forward-Backward Algorithm: Finds Marginals

\[ \alpha_2(n) = \text{total weight of these path prefixes} \]

(found by dynamic programming: matrix-vector products)
Forward-Backward Algorithm: Finds Marginals

$\beta_2(n) = \text{total weight of these path suffixes}$

(found by dynamic programming: matrix-vector products)
Forward-Backward Algorithm: Finds Marginals

\[ \alpha_2(n) = \text{total weight of these path prefixes } (a + b + c) \]

\[ \beta_2(n) = \text{total weight of these path suffixes } (x + y + z) \]

Product gives \( (a+b+c)(x+y+z) \) = total weight of paths
Oops! The weight of a path through a state also includes a weight at that state. So \( \alpha(n) \cdot \beta(n) \) isn’t enough.

The extra weight is the opinion of the emission probability at this variable.

Forward-Backward Algorithm: Finds Marginals

"belief that \( Y_2 = n \)"

Total weight of all paths through

\[
= \alpha_2(n) \cdot A(\text{pref.}, n) \cdot \beta_2(n)
\]
Forward-Backward Algorithm: Finds Marginals

"belief that $Y_2 = v$"

"belief that $Y_2 = n$"

total weight of all paths through

$$= \alpha_2(v) \ A(\text{pref.}, v) \ \beta_2(v)$$
Forward-Backward Algorithm: Finds Marginals

\[ \text{total weight of all paths through } a = \alpha_2(a) \cdot A(\text{pref.}, a) \cdot \beta_2(a) \]

“belief that \( Y_2 = v \)”

“belief that \( Y_2 = n \)”

“belief that \( Y_2 = a \)”

sum = \( Z \) (total weight of all paths)

divide by \( Z=0.5 \) to get marginal probs

\( v \) 0.1
\( n \) 0
\( a \) 0.4

\( v \) 0.2
\( n \) 0
\( a \) 0.8

\( X_2 \) preferred
Forward-Backward Algorithm

Could be verb or noun
find

Could be adjective or verb
preferred

Could be noun or verb
tags
THE FORWARD-BACKWARD ALGORITHM
**Forward-Backward Algorithm**

1. Initialize

   \[ \alpha_0(\text{START}) = 1 \]
   \[ \beta_{T+1}(\text{END}) = 1 \]
   \[ \alpha_0(k) = 0, \ \forall k \neq \text{START} \]
   \[ \beta_{T+1}(k) = 0, \ \forall k \neq \text{END} \]

2. **Forward Algorithm**

   for \( t = 1, \ldots, T + 1 \):
   
   for \( k = 1, \ldots, K \):
   
   \[ \alpha_t(k) = \sum_{j=1}^{K} p(x_t | y_t = k) \alpha_{t-1}(j)p(y_t = k | y_{t-1} = j) \]

3. **Backward Algorithm**

   for \( t = T, \ldots, 0 \):
   
   for \( k = 1, \ldots, K \):
   
   \[ \beta_t(k) = \sum_{j=1}^{K} p(x_{t+1} | y_{t+1} = j) \beta_{t+1}(j)p(y_{t+1} = j | y_t = k) \]

4. **Evaluation**

   \[ p(x) = \alpha_{T+1}(\text{END}) \]

5. **Marginals**

   \[ p(y_t = k | x) = \frac{\alpha_t(k) \beta_t(k)}{p(x)} \]