ECE49595NL Lecture 11: Hidden Markov Models

Jeffrey Mark Siskind

Elmore Family School of Electrical and Computer Engineering

Spring 2025

PURDUE UNIVERSITY | Elmore Family School of Electrical and Computer Engineering

Size Hyperparameters and Indices

- L: number of sentences
- l: sentence index
- I_l: number of words in sentence l
- I: number of words when only one sentence
- i: word position in sentence l
- K: number of words in dictionary
- k: word index in dictionary
- J: number of states
- j: state index

Random Variables

- $T_{li} \in \{1, \ldots, J\}$: state index at position i of sentence l
- $W_{li} \in \{1, \ldots, K\}$: word index at position i of sentence l
- $T_i \in \{1, \ldots, J\}$: state index at position i when only one sentence
- $W_i \in \{1, \ldots, K\}$: word index at position i when only one sentence

Model Parameters

- $b_j = \Pr(T_{l1} = j)\ (\forall l)$
- $a_{jj'} = \Pr(T_{li+1} = j' | T_{li} = j)\ (\forall l)(\forall i)$
- $c_{jk} = \Pr(W_{li} = k | T_{li} = j)\ (\forall l)(\forall i)$

Sampling

1. $i := 1$
2. Sample T_i from b.
3. Sample W_i from c_{T_i}.
4. $i := i + 1$
5. Sample T_i from $a_{T_{i-1}}$.
6. Go to 3.

State Sequence Probability

- $\Pr(W_1 = k_1, \ldots, W_I = k_I, T_1 = j_1, \ldots, T_I = j_I | a, b, c)$
- $= \Pr(\vec{W} = \vec{k}, \vec{T} = \vec{j} | a, b, c)$
- $= p(\vec{k}, \vec{j})$
- $= b_{T_1} c_{T_1 W_1} \prod_{i=2}^{I} a_{T_{i-1} T_i} c_{T_i W_i}$

Maximum A Posteriori Probability (MAP estimate)

Naïve Method

- $(\arg)\max_{j_1,\ldots,j_I} \Pr(W_1 = k_1, \ldots, W_I = k_I, T_1 = j_1, \ldots, T_I = j_I | a, b, c)$
- $= (\arg)\max_{\vec{j}} \Pr(\vec{W} = \vec{k}, \vec{T} = \vec{j} | a, b, c)$
- $= (\arg)\max_{\vec{j}} p(\vec{k}, \vec{j})$

Viterbi Probabilities

- $\delta_{ij} = \max_{j_1,\ldots,j_i} \Pr(W_1 = k_1, \ldots, W_i = k_i, T_1 = j_1, \ldots, T_i = j_i | a, b, c)$

Viterbi Algorithm

- $\delta_{1j} = b_j c_{jW_1} \; (\forall j)$
- $\delta_{ij} = \left(\max_{j'} \delta_{i-1j'} a_{j'j}\right) c_{jW_i} \; (\forall i > 1)(\forall j)$

Maximum A Posteriori Probability (MAP estimate)

Fast Method

- $\max_{j_1,\ldots,j_I} \Pr(W_1 = k_1, \ldots, W_I = k_I, T_1 = j_1, \ldots, T_I = j_I | a, b, c)$
- $= \max_{\vec{j}} \Pr(\vec{W} = \vec{k}, \vec{T} = \vec{j} | a, b, c)$
- $= \max_{\vec{j}} p(\vec{k}, \vec{j})$
- $= \max_{j} \delta_{Ij}$

Likelihood Estimation

Naïve Method

- $\Pr(W_{l1} = k_{l1}, \ldots, W_{lI_l} = k_{lI_l} | a, b, c)$
- $= \Pr(\vec{W}_l = \vec{k}_l | a, b, c)$
- $= \sum_{\vec{j}_l} \Pr(\vec{W}_l = \vec{k}_l, \vec{T}_l = \vec{j}_l | a, b, c)$
- $= \sum_{\vec{j}_l} p(\vec{k}_l, \vec{j}_l)$
- $= \sum_{\vec{j}_l} b_{T_1} c_{T_1 W_1} \prod_{i=2}^{I} a_{T_{i-1} T_i} c_{T_i W_i}$

Forward Probabilities

- $\alpha_{lij} = \Pr(W_{l1} = k_{l1}, \ldots, W_{li} = k_{li}, T_{li} = j | a, b, c)$

Backward Probabilities

- $\beta_{lij} = \Pr(W_{li+1} = k_{li+1}, \ldots, W_{lI_l} = k_{lI_l}, T_{li} = j | a, b, c)$

Forward Algorithm

- $\alpha_{l1j} = b_j c_{jW_{l1}}\ (\forall l)(\forall j)$
- $\alpha_{lij} = \left(\sum_{j'} \alpha_{li-1j'} a_{j'j}\right) c_{jW_{li}}\ (\forall l)(\forall i > 1)(\forall j)$

Backward Algorithm

- $\beta_{lI_lj} = 1\ (\forall l)(\forall j)$
- $\beta_{lij} = \sum_{j'} a_{jj'} c_{j' w_{li+1}} \beta_{li+1j'}\ (\forall l)(\forall i < I_l)(\forall j)$

Likelihood Estimation

Fast Method

- $\Pr(W_{l1} = k_{l1}, \ldots, W_{lI_l} = k_{lI_l} | a, b, c)$
- $= \Pr(\vec{W}_l = \vec{k}_l | a, b, c)$
- $= \sum_{\vec{j}_l} \Pr(\vec{W}_l = \vec{k}_l, \vec{T}_l = \vec{j}_l | a, b, c)$
- $= \sum_{\vec{j}_l} p(\vec{k}_l, \vec{j}_l)$
- $= \sum_{\vec{j}_l} b_{T_1} c_{T_1 W_1} \prod_{i=2}^{I} a_{T_{i-1} T_i} c_{T_i W_i}$
- $= \sum_{j} \alpha_{lI_l j}$
- $= \sum_{j} b_j c_{l W_{l1}} \beta_{l1j}$

Forward-Backward Algorithm (E-step)

- $\Pr(W_{l1} = k_{l1}, \ldots, W_{lI_l} = k_{lI_l}, T_{li} = j|a, b, c)$
- $= \Pr(\vec{W}_l = \vec{k}_l, T_{li} = j|a, b, c)$
- $= \alpha_{lij}\beta_{lij}$
- $\gamma_{lij} = \Pr(T_{li} = j|a, b, c)$
- $= \frac{\alpha_{lij}\beta_{lij}}{\Pr(W_{l1}=k_{l1},\ldots,W_{lI_l}=k_{lI_l}|a,b,c)}$
- $= \frac{\alpha_{lij}\beta_{lij}}{\displaystyle\sum_j \alpha_{lI_lj}}$
- $= \frac{\alpha_{lij}\beta_{lij}}{\displaystyle\sum_j b_j c_{lW_{l1}} \beta_{l1j}}$

Forward-Backward Algorithm (E-step)

- $\Pr(W_{l1}=k_{l1},\ldots,W_{lI_l}=k_{lI_l},T_{li}=j,T_{li+1}=j'|a,b,c)$
- $=\Pr(\vec{W}_l=\vec{k}_l,T_{li}=j,T_{li+1}=j'|a,b,c)$
- $=\alpha_{lij}a_{jj'}c_{j'W_{li+1}}\beta_{li+1j'}$
- $\psi_{lijj'}=\Pr(T_{li}=j,T_{li+1}=j'|a,b,c)$
- $=\frac{\alpha_{lij}a_{jj'}c_{j'W_{li+1}}\beta_{li+1j'}}{\Pr(W_{l1}=k_{l1},\ldots,W_{lI_l}=k_{lI_l}|a,b,c)}$
- $=\frac{\alpha_{lij}a_{jj'}c_{j'W_{li+1}}\beta_{li+1j'}}{\displaystyle\sum_j \alpha_{lI_lj}}$
- $=\frac{\alpha_{lij}a_{jj'}c_{j'W_{li+1}}\beta_{li+1j'}}{\displaystyle\sum_j b_jc_{lW_{l1}}\beta_{l1j}}$

Baum-Welch Reestimation Procedure (M-step)

- $b_j :\propto \sum_{l} \gamma_{l1j}\ (\forall j)$
- $a_{jj'} :\propto \sum_{l,i} \psi_{lijj'}\ (\forall j)(\forall j')$
- $c_{jk} :\propto \sum_{l,i,W_{li}=k} \gamma_{lij}\ (\forall j)(\forall k)$