NLP Training

Hidden Markov Model

AMIT DHOMNE

CONTENTS

Hidden Markov Models

- **Hidden Markow Models:**
  - **A hidden Markov model (HMM) is a statistical model,in which the system being modeled is assumed to be a Markov process (**Memoryless process: *its future and past are independent* **) with *hidden* states.**

Hidden Markov Models

- **Hidden Markow Models:**
  - **Has a set of states each of which has limited number of transitions and emissions,**
  - **Each transition between states has an assisgned probability,**
  - **Each model strarts from start state and ends in end state,**

Hidden Markov Models

Hidden Markov Model

Hidden + Markov Model

What is 'hidden'? What is 'Markov model'?

Hidden Markov Models

- **Markow Models :**
  - Talk about weather,
  - Assume there are three types of weather:
    - Sunny, 
    - Rainy, 
    - Foggy. 

Markov Models

- Weather prediction is about the what would be the weather tomorrow,
  - Based on the observations on the past.

Markov Models

- Weather at day n is $q_n \in \{sunny, rainy, foggy\}$

  - q_n depends on the known weathers of the past days (q_{n-1}, q_{n-2},…)

Markov Models

- We want to find that:

$$P(q_n | q_{n-1}, q_{n-2}, \ldots, q_1)$$

  – means given the past weathers what is the probability of any possible weather of today.

Markov Models

- **Markow Models:**
- For example:
  - if we knew the weather for last three days was:

  - the probability that tomorrow would be  is:

P(q_4=  | q_3=  , q_2=  q_1= )

Markov Models

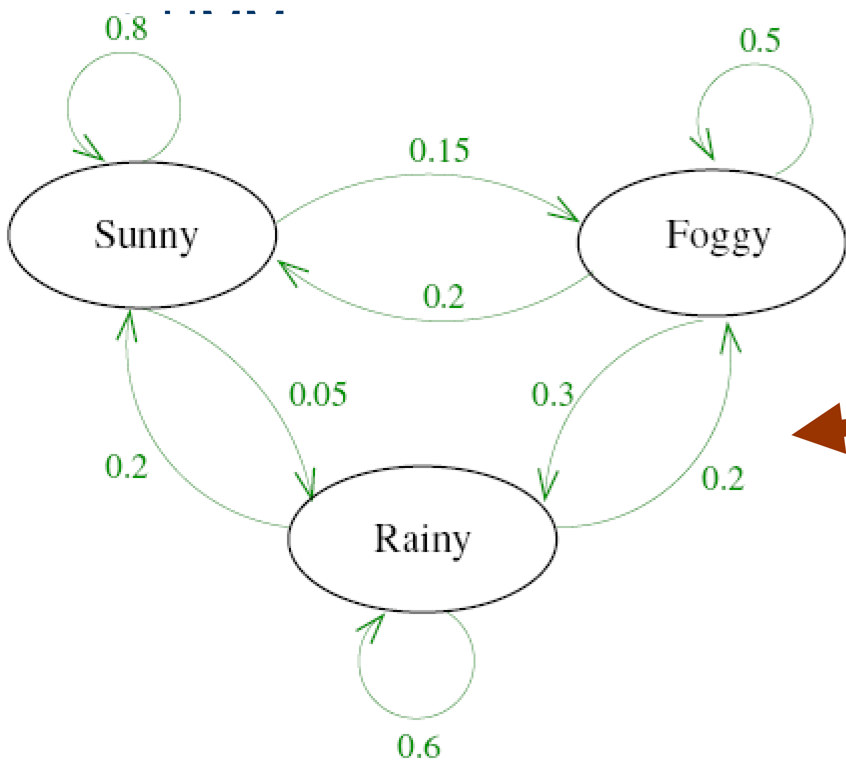
- **Markow Models and Assumption (cont.):**
  - Therefore, make a simplifying assumption Markov assumption:
    - For sequence: $\{q_1, q_2, \ldots, q_n\}$

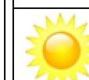
$$P(q_n | q_{n-1}, q_{n-2}, \ldots, q_1) = P(q_n | q_{n-1})$$

    - **the weather of tomorrow only depends on today** (first order Markov model)

Markov Models

- **Markow Models and Assumption (cont.):**
  Examples:



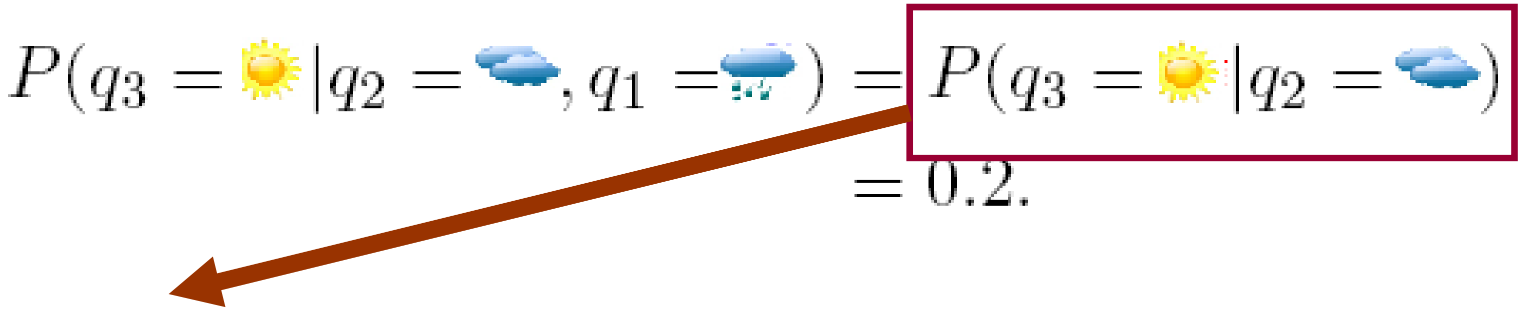
Today's weather	Tomorrow's weather		
			
	0.8	0.05	0.15
	0.2	0.6	0.2
	0.2	0.3	0.5

Markov Models

- **Markow Models and Assumption (cont.):** Examples:
  - If the weather yesterday was rainy and today is foggy what is the probability that tomorrow it will be sunny?

Markov Models

- **Markow Models and Assumption (cont.):**
  - Examples:
    - If the weather yesterday was rainy and today is foggy what is the probability that tomorrow it will be sunny?



Markov assumption

Hidden Markov Models

- **Hidden Markov Models (HMMs):**
  - What is HMM:
    - Suppose that you are locked in a room for several days,
    - you try to predict the weather outside,
    - The only piece of evidence you have is whether the person who comes into the room bringing your daily meal is carrying an umbrella or not.

Hidden Markov Models

- **Hidden Markov Models (HMMs):**
  - What is HMM (cont.):
    - assume probabilities as seen in the table:

Weather	Probability of umbrella
Sunny	0.1
Rainy	0.8
Foggy	0.3

Probability $P(x_i|q_i)$ of carrying an umbrella ($x_i = \text{true}$) based on the weather q_i on some day i

Hidden Markov Models

- **Hidden Markov Models (HMMs):**
  - What is HMM (cont.):
    - Finding the probability of a certain weather

$$q_n \in \{sunny, rainy, foggy\}$$

    - is based on the observations $\mathbf{x}_i$:

Hidden Markov Models

- **Hidden Markov Models (HMMs):**
  - What is HMM (cont.):
    - Using Bayes rule:

$$P(q_i|x_i) = \frac{P(x_i|q_i)P(q_i)}{P(x_i)}$$

    - For n days:

$$P(q_1, \ldots, q_n|x_1, \ldots, x_n) = \frac{P(x_1, \ldots, x_n|q_1, \ldots, q_n)P(q_1, \ldots, q_n)}{P(x_1, \ldots, x_n)}$$

Hidden Markov Models

- **Hidden Markov Models (HMMs):**
  - Examples:
    - Suppose the day you were locked in it was sunny. The next day, the caretaker carried an umbrella into the room.
    - You would like to know, **what the weather was like on this second day**.

Discrete Markov Processes (Markov Chains)

- The goal is to make a sequence of decisions where a particular decision may be influenced by earlier decisions.
- Consider a system that can be described at any time as being in one of a set of N distinct states $w_1, w_2, \ldots, w_N$.
- Let $w(t)$ denote the actual state at time t where $t = 1, 2, \ldots$.
- The probability of the system being in state $w(t)$ is $P(w(t)|w(t-1), \ldots, w(1))$.

Hiddden Markov Models

- We assume that the state $w(t)$ is conditionally independent of the previous states given the predecessor state $w(t-1)$, i.e.,

$$P(w(t)|w(t-1),\ldots,w(1)) = P(w(t)|w(t-1)).$$

- We also assume that the Markov Chain defined by $P(w(t)|w(t-1))$ is time homogeneous (independent of the time t).

Hidden Markov Models

- A particular *sequence of states* of length T is denoted by

$$\mathcal{W}^T = \{w(1), w(2), \ldots, w(T)\}.$$

- The model for the production of any sequence is described by the *transition probabilities*

$$a_{ij} = P(w(t) = w_j | w(t-1) = w_i)$$

where $i, j \in \{1, \ldots, N\}$, $a_{ij} \geq 0$, and $\sum_{j=1}^{N} a_{ij} = 1, \forall i$.

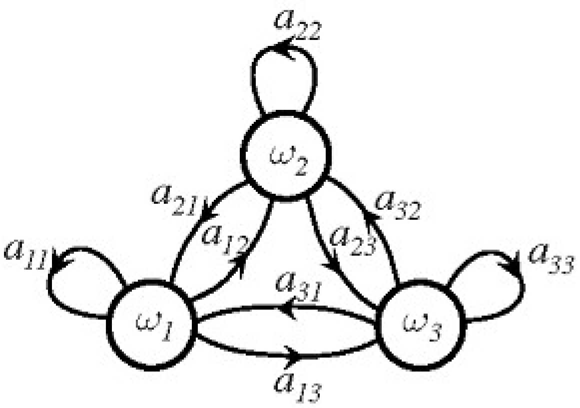
Hidden Markov Models

- There is no requirement that the transition probabilities are symmetric ($a_{ij} \neq a_{ji}$, in general).
- Also, a particular state may be visited in succession ($a_{ii} \neq 0$, in general) and not every state need to be visited.
- This process is called an *observable Markov model* because the output of the process is the set of states at each instant of time, where each state corresponds to a physical (observable) event.

Hidden Markov Model Examples

- Consider the following 3-state first-order Markov model of the weather in Ankara:
  - w_1: rain/snow
  - w_2: cloudy
  - w_3: sunny

$$\Theta = \{a_{ij}\}$$
$$= \begin{pmatrix} 0.4 & 0.3 & 0.3 \\ 0.2 & 0.6 & 0.2 \\ 0.1 & 0.1 & 0.8 \end{pmatrix}$$



Hidden Markov Models

- We denote the observation at time t as $v(t)$ and the probability of producing that observation in state $w(t)$ as $P(v(t)|w(t))$.
- There are many possible state-conditioned observation distributions.
- When the observations are discrete, the distributions

$$b_{jk} = P(v(t) = v_k | w(t) = w_j)$$

are probability mass functions where $j \in \{1, \ldots, N\}$, $k \in \{1, \ldots, M\}$, $b_{jk} \geq 0$, and $\sum_{k=1}^{M} b_{jk} = 1, \forall j$.

Hidden Markov Models

- When the observations are continuous, the distributions are typically specified using a parametric model family where the most common family is the Gaussian mixture

$$b_j(\mathbf{x}) = \sum_{k=1}^{M_j} \alpha_{jk}\, p(\mathbf{x} | \boldsymbol{\mu}_{jk}, \boldsymbol{\Sigma}_{jk})$$

  where $\alpha_{jk} \geq 0$ and $\sum_{k=1}^{M_j} \alpha_{jk} = 1, \forall j$.

- We will restrict ourselves to discrete observations where a particular sequence of visible states of length T is denoted by

$$\mathcal{V}^T = \{v(1), v(2), \ldots, v(T)\}.$$

Hidden Markov Models

- An HMM is characterized by:
  - N, the number of hidden states
  - M, the number of distinct observation symbols per state
  - $\{a_{ij}\}$, the state transition probability distribution
  - $\{b_{jk}\}$, the observation symbol probability distribution
  - $\{\pi_i = P(w(1) = w_i)\}$, the initial state distribution
  - $\mathbf{\Theta} = (\{a_{ij}\}, \{b_{jk}\}, \{\pi_i\})$, the complete parameter set of the model

Three Fundamental Problems for HMMs

- *Evaluation problem:* Given the model, compute the probability that a particular output sequence was produced by that model (solved by the forward algorithm).
- *Decoding problem:* Given the model, find the most likely sequence of hidden states which could have generated a given output sequence (solved by the Viterbi algorithm).
- *Learning problem:* Given a set of output sequences, find the most likely set of state transition and output probabilities (solved by the Baum-Welch algorithm).

HMM Evaluation Problem

- A particular *sequence of observations* of length T is denoted by

$$\mathcal{V}^T = \{v(1), v(2), \ldots, v(T)\}.$$

- The probability of observing this sequence can be computed by enumerating every possible state sequence of length T as

$$\begin{aligned} P(\mathcal{V}^T|\boldsymbol{\Theta}) &= \sum_{\text{all } \mathcal{W}^T} P(\mathcal{V}^T, \mathcal{W}^T|\boldsymbol{\Theta}) \\ &= \sum_{\text{all } \mathcal{W}^T} P(\mathcal{V}^T|\mathcal{W}^T, \boldsymbol{\Theta}) P(\mathcal{W}^T|\boldsymbol{\Theta}). \end{aligned}$$

HMM Evaluation Problem

- This summation includes N^T terms in the form

$$P(\mathcal{V}^T|\mathcal{W}^T)P(\mathcal{W}^T) = \left(\prod_{t=1}^{T} P(v(t)|w(t))\right)\left(\prod_{t=1}^{T} P(w(t)|w(t-1))\right)$$
$$= \prod_{t=1}^{T} P(v(t)|w(t))P(w(t)|w(t-1))$$

  where $P(w(t)|w(t-1))$ for $t = 1$ is $P(w(1))$.
- It is unfeasible with computational complexity $O(N^T T)$.
- However, a computationally simpler algorithm called the *forward algorithm* computes $P(\mathcal{V}^T|\Theta)$ recursively.

HMM Evaluation Problem

- Define $\alpha_j(t)$ as the probability that the HMM is in state w_j at time t having generated the first t observations in $\mathcal{V}^T$

$$\alpha_j(t) = P(v(1), v(2), \ldots, v(t), w(t) = w_j | \Theta).$$

- $\alpha_j(t), j = 1, \ldots, N$ can be computed as

$$\alpha_j(t) = \begin{cases} \pi_j b_{jv(1)} & t = 1 \\ \left(\sum_{i=1}^{N} \alpha_i(t-1) a_{ij}\right) b_{jv(t)} & t = 2, \ldots, T. \end{cases}$$

- Then, $P(\mathcal{V}^T | \Theta) = \sum_{j=1}^{N} \alpha_j(T)$.

HMM Evaluation Problem

- Similarly, we can define a *backward algorithm* where

$$\beta_i(t) = P(v(t+1), v(t+2), \ldots, v(T) | w(t) = w_i, \boldsymbol{\Theta})$$

  is the probability that the HMM will generate the observations from $t+1$ to T in $\mathcal{V}^T$ given that it is in state w_i at time t.

- $\beta_i(t), i = 1, \ldots, N$ can be computed as

$$\beta_i(t) = \begin{cases} 1 & t = T \\ \sum_{j=1}^{N} \beta_j(t+1) a_{ij} b_{jv(t+1)} & t = T-1, \ldots, 1. \end{cases}$$

- Then, $P(\mathcal{V}^T | \boldsymbol{\Theta}) = \sum_{i=1}^{N} \beta_i(1) \pi_i b_{iv(1)}$.

HMM Evaluation Problem

- The computations of both $\alpha_j(t)$ and $\beta_i(t)$ have complexity $O(N^2T)$.
- For classification, we can compute the posterior probabilities

$$P(\Theta|\mathcal{V}^T) = \frac{P(\mathcal{V}^T|\Theta)P(\Theta)}{P(\mathcal{V}^T)}$$

  where $P(\Theta)$ is the prior for a particular class, and $P(\mathcal{V}^T|\Theta)$ is computed using the forward algorithm with the HMM for that class.
- Then, we can select the class with the highest posterior.

HMM Decoding Problem

- Given a sequence of observations $\mathcal{V}^T$, we would like to find the most probable sequence of hidden states.
- One possible solution is to enumerate every possible hidden state sequence and calculate the probability of the observed sequence with $O(N^T T)$ complexity.
- We can also define the problem of finding the optimal state sequence as finding the one that includes the states that are individually most likely.
- This also corresponds to maximizing the expected number of correct individual states.

HMM Decoding Problem

- Define $\gamma_i(t)$ as the probability that the HMM is in state w_i at time t given the observation sequence $\mathcal{V}^T$

$$\gamma_i(t) = P(w(t) = w_i | \mathcal{V}^T, \mathbf{\Theta})$$
$$= \frac{\alpha_i(t)\beta_i(t)}{P(\mathcal{V}^T|\mathbf{\Theta})} = \frac{\alpha_i(t)\beta_i(t)}{\sum_{j=1}^N \alpha_j(t)\beta_j(t)}$$

where $\sum_{i=1}^N \gamma_i(t) = 1$.

- Then, the individually most likely state $w(t)$ at time t becomes

$$w(t) = w_{i'} \text{ where } i' = \arg \max_{i=1,\ldots,N} \gamma_i(t).$$

HMM Decoding Problem

- One problem is that the resulting sequence may not be consistent with the underlying model because it may include transitions with zero probability ($a_{ij} = 0$ for some i and j).
- One possible solution is the *Viterbi algorithm* that finds the single best state sequence $\mathcal{W}^T$ by maximizing $P(\mathcal{W}^T|\mathcal{V}^T, \boldsymbol{\Theta})$ (or equivalently $P(\mathcal{W}^T, \mathcal{V}^T|\boldsymbol{\Theta})$).
- This algorithm recursively computes the state sequence with the highest probability at time t and keeps track of the states that form the sequence with the highest probability at time T

HMM Learning Problem

- The goal is to determine the model parameters $\{a_{ij}\}$, $\{b_{jk}\}$ and $\{\pi_i\}$ from a collection of training samples.
- Define $\xi_{ij}(t)$ as the probability that the HMM is in state w_i at time $t-1$ and state w_j at time t given the observation sequence $\mathcal{V}^T$

$$
\begin{aligned}
\xi_{ij}(t) &= P(w(t-1) = w_i, w(t) = w_j | \mathcal{V}^T, \mathbf{\Theta}) \\
&= \frac{\alpha_i(t-1)\, a_{ij}\, b_{jv(t)}\, \beta_j(t)}{P(\mathcal{V}^T | \mathbf{\Theta})} \\
&= \frac{\alpha_i(t-1)\, a_{ij}\, b_{jv(t)}\, \beta_j(t)}{\sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i(t-1)\, a_{ij}\, b_{jv(t)}\, \beta_j(t)}.
\end{aligned}
$$

HMM Learning Problem

- $\gamma_i(t)$ defined in the decoding problem and $\xi_{ij}(t)$ defined here can be related as

$$\gamma_i(t-1) = \sum_{j=1}^{N} \xi_{ij}(t).$$

- Then, $\hat{a}_{ij}$, the estimate of the probability of a transition from w_i at $t-1$ to w_j at t, can be computed as

$$\hat{a}_{ij} = \frac{\text{expected number of transitions from } w_i \text{ to } w_j}{\text{expected total number of transitions away from } w_i}$$
$$= \frac{\sum_{t=2}^{T} \xi_{ij}(t)}{\sum_{t=2}^{T} \gamma_i(t-1)}.$$

HMM Learning Problem

- Similarly, $\hat{b}_{jk}$, the estimate of the probability of observing the symbol v_k while in state w_j, can be computed as

$$\hat{b}_{jk} = \frac{\text{expected number of times observing symbol } v_k \text{ in state } w_j}{\text{expected total number of times in } w_j}$$
$$= \frac{\sum_{t=1}^{T} \delta_{v(t),v_k} \gamma_j(t)}{\sum_{t=1}^{T} \gamma_j(t)}$$

  where $\delta_{v(t),v_k}$ is the Kronecker delta which is 1 only when $v(t) = v_k$.

- Finally, $\hat{\pi}_i$, the estimate for the initial state distribution, can be computed as $\hat{\pi}_i = \gamma_i(1)$ which is the expected number of times in state w_i at time $t = 1$.

HMM Learning Problem

- These are called the *Baum-Welch* equations (also called the *EM estimates for HMMs* or the *forward-backward algorithm*) that can be computed iteratively until some convergence criterion is met (e.g., sufficiently small changes in the estimated values in subsequent iterations).
- See (Bilmes, 1998) for the estimates $\hat{b}_j(\mathbf{x})$ when the observations are continuous and their distributions are modeled using Gaussian mixtures.

Application Areas of HMM

- On-line handwriting recognition
- Speech recognition
- Gesture recognition
- Language modeling
- Motion video analysis and tracking
- Stock price prediction

and many more….

References

- R.O. Duda, P.E. Hart, and D.G. Stork, Pattern Classification, New York: John Wiley, 2001.
- Selim Aksoy, “Pattern Recognition Course Materials”, Bilkent University, 2011.