Hidden Markov Models (HMMs) – Part 1

May 21, 2013
References

- X. Wang, A. Acero, H-W. Hon: Spoken Language Processing, Chapter 8, pp 374-409, Prentice Hall, 2001
- Tapas Kanungo, University Maryland, HMM Tutorial Slides (some of his slides have been reused here)
Outline

• Motivation: Problems with Pattern Matching
• Markov Models
• Hidden Markov Models
  – Introduction, some properties, topologies
• Three Main Problems of HMMs and algorithmic solutions:
  – The Evaluation Problem: Forward Algorithm
  – The Decoding Problem: Viterbi Algorithm
  – The Learning Problem: Forward-Backward Algorithm
• Hidden Markov Models in Speech Recognition
  – Overview of Hidden Markov Models Training
    • Using (Hand-)Labeled Data
    • K-Means
    • Training HMMs with Viterbi
• Components of an HMM Recognizer
What we have seen so far …

- Signal preprocessing, feature extraction

- We model phonemes. However, we want to recognize whole words and sentences.

  → In this lecture:

  Classification of phoneme sequences

  - Problem: We can classify each single phoneme …

    … but not every sequence of recognized phonemes makes sense

  - Furthermore, we want to use a-priori information for the probability of phonemes and words
Dynamic Time Warping (DTW)

- **Goal:** We want to find a distance between two utterances
  → The lower, the better
- **Problem:** We need to consider *all* paths and find the best!
- **Solution:**
  - For each time \( t \), calculate the *cumulative distances* \( \alpha(s,t) \), which describe the distance of the partial utterances up to the states \( q(s,t) \) (\( s=1, \ldots, S \)).
  - The distances for time \( t+1 \) are calculated from those of time \( t \).
  - At this point, the minimization of the distance is applied.
  - Requires a distance measure \( d(s,t) \) for the observed frame \( t \) and the reference frame \( s \) (high \( d(s,t) \) means large distance) – e.g. Euclidean distance

![Diagram](image-url)
Dynamic Time Warping – Application

- We can use the DTW to recognize whole words:
  - Compute the DTW distance for each possible reference pattern
  - The word with the smallest distance is considered to be recognized

- Is still applied in practice, for very small vocabularies

- What are the problems?
Hidden Markov Models (HMMs) - 8

Problems with Pattern Matching

The DTW algorithm can be used to differentiate a small amount of words, but:

- **Needs endpoint detection**
  - If split in smaller units: needs segmentation into these units
- **High computational effort (esp. for large vocabularies), proportional to vocabulary size**
- **Large vocabulary also means: need huge amount of training data**
  - Collection of lots of reference patterns (inconvenient for user)
  - Difficult to train suitable references (or sets of references)
- **Poor performance when the environment changes**
- **Works only well for speaker-dependent recognition (variations)**
- **Unsuitable**
  - Where speaker is unknown, no training is feasible
  - Continuous speech (comb. explosion of patterns, coarticulation)
  - Impossible to recognize untrained words
- **Difficult to train/recognize subword units**

\[\Rightarrow\] We need a different method that allows to train and recognize **smaller units** (syllables, phonemes)
Make a Wish

- We would like to work with speech units shorter than words
  ⇒ each subword unit occurs often, training is easier, less data

- We want to recognize speech from any speaker, without prior training
  ⇒ store "speaker-independent" reference
    (examples from many speakers)

- We want to recognize continuous rather than isolated speech
  ⇒ handle coarticulation effects, handle sequences of words

- We want to recognize words that did not occur in the training set
  ⇒ train subword units and compose any word out of these
    (vocabulary independence)

- We would prefer a solid mathematical foundation
Speech Production seen as Stochastic Process

- The *same* word / phoneme sounds *different* every time it is uttered
- Regard words / phonemes as states of a speech production process
- In a given state we can observe different *acoustic sounds*
- Not all sounds are possible / likely in every state
- We say:
  In a given state the speech process "emits" sounds according to some probability distribution
- The production process makes transitions from one state to another
- Not all transitions are possible, they have different probabilities

⇒ When we specify the probabilities for sound-emissions (emission probabilities) and for the state transitions, we call this a model.
Speech Production seen as Stochastic Process

- Basic principle of our improved recognizer:
  - The speech process is in a state at any time (we cannot observe the state directly)
  - In each state certain sounds are emitted corresponding to a certain probability distribution. These probabilities are called emission probabilities.
  - The transitions between the states also occur according to a certain probability distribution. These probabilities are called transition probabilities.

- states ↔ phonemes 
  (sound units, min. 30ms)

- observations ↔ frames of the acoustic signal 
  (each 10 ms)
What's different?

Reference in terms of state sequence of statistical models, models consists of prototypical references vectors.

Hypothesis = recognized sentence
Markov Models (1)

- Observable states:
  \[ 1, 2, \ldots, N \]

- Observed sequence:
  \[ q_1, q_2, \ldots, q_t, \ldots, q_T \]

- First order Markov assumption:
  \[ P(q_t = j \mid q_{t-1} = i, q_{t-2} = k, \ldots) = P(q_t = j \mid q_{t-1} = i) \]

- Stationarity:
  \[ P(q_t = j \mid q_{t-1} = i) = P(q_{t+1} = j \mid q_{t+1-1} = i) \]
Markov Models (2)

- State transition matrix $A$:

$$A = \begin{bmatrix}
  a_{11} & a_{12} & \cdots & a_{1j} & \cdots & a_{1N} \\
  a_{21} & a_{22} & \cdots & a_{2j} & \cdots & a_{2N} \\
  \vdots & \vdots & \ddots & \vdots & \cdots & \vdots \\
  a_{i1} & a_{i2} & \cdots & a_{ij} & \cdots & a_{iN} \\
  \vdots & \vdots & \ddots & \vdots & \cdots & \vdots \\
  a_{N1} & a_{N2} & \cdots & a_{Nj} & \cdots & a_{NN}
\end{bmatrix}$$

where

$$a_{ij} = P(q_t = j | q_{t-1} = i) \quad 1 \leq i, j, \leq N$$

- Constraints on $a_{ij}$:

$$a_{ij} \geq 0, \quad \forall i, j$$

$$\sum_{j=1}^{N} a_{ij} = 1, \quad \forall i$$
Markov Models - Example

- States:
  1. Rainy ($R$)
  2. Cloudy ($C$)
  3. Sunny ($S$)

- State transition probability matrix:

$$A = \begin{bmatrix} 0.4 & 0.3 & 0.3 \\ 0.2 & 0.6 & 0.2 \\ 0.1 & 0.1 & 0.8 \end{bmatrix}$$

- Compute the probability of observing $SSRRSCS$ given that today is $S$. 
Markov Models - Example

Basic conditional probability rule:

\[ P(A, B) = P(A | B)P(B) = P(B | A)P(A) \]

The Markov chain rule:

\[
\begin{align*}
P(q_1, q_2, \ldots, q_T) &= P(q_T | q_1, q_2, \ldots, q_{T-1})P(q_1, q_2, \ldots, q_{T-1}) \\
&= P(q_T | q_{T-1})P(q_1, q_2, \ldots, q_{T-1}) \\
&= P(q_T | q_{T-1})P(q_{T-1} | q_{T-2})P(q_1, q_2, \ldots, q_{T-2}) \\
&= P(q_T | q_{T-1})P(q_{T-1} | q_{T-2}) \cdots P(q_2 | q_1)P(q_1)
\end{align*}
\]

First order Markov assumption
Markov Models - Example

- Observation sequence $O$:

$$O = (S, S, S, R, R, S, C, S)$$

- Using the chain rule we get:

$$P(O|\text{model}) = P(S, S, S, R, R, S, C, S|\text{model})$$

$$= P(S)P(S|R)P(S|S)P(R|S)P(R|R) \times P(S|R)P(C|S)P(S|C)$$

$$= \pi_3 a_{33} a_{33} a_{31} a_{11} a_{13} a_{32} a_{23}$$

$$= (1)(0.8)^2(0.1)(0.4)(0.3)(0.1)(0.2)$$

$$= 1.536 \times 10^{-4}$$

- The prior probability $\pi_i = P(q_1 = i)$

Today is $S \Rightarrow P(S) = 1$
What differs the process of speech production from weather modeling (as shown on the previous slides)?

- For weather modeling, we compute the probability of a *direct* and *exactly observable* state sequence (either it is sunny or not).
- Consequently, a state and its observation are *exactly the same*.
- However, in speech we have a continuum of possible tokens (typically frames of the speech signal whose distribution follows Gaussians) which should be assigned to a limited number of states (phonemes).
  - Each phoneme can (theoretically) be realized in infinite ways (but with different probability).
  - Also the boundaries of phonemes can not be defined exactly.
  - There is no 1-1 relation between the phonemes uttered by a speaker and its observable acoustics.
- In speech, the states are **hidden**.
  - Observations are indirectly possible via sound emissions.
  - These observations are also *probabilistic*!
Hidden Markov Models

Consequently, we need an extension of the Markov Models.

→ We can solve our problems with *Hidden Markov Models (HMMs)*.

What are *Hidden* Markov Models?

- An **HMM** is a statistical Markov model in which the system being modeled is assumed to be a Markov process with unobserved (*hidden*) states
  - The state sequence is probabilistic.
  - We say: Each state *emits* an observation (a frame of the speech signal): These emissions are *also* probabilistic.
    → Observations are probabilistic functions of states.
  - The state sequences are *hidden*.
    → The states are not observable.

- **HMMs** are Markov models
  - The probabilities to enter a next state depend only on the current state. State transitions are still probabilistic.
Hidden Markov Models

The fact that state sequence is not observable has some consequences:

• Decoding with HMMs
  – Based on the observations we have to draw conclusions about a possible state sequence
  – Thereby we will never find an exact solution, only one with the highest probability.

• Training of HMMs
  – A related problem is the training of an HMM, where we know the traversed state sequence but not the time of the state transitions.

But these properties model the process of speech production/recognition well!
Example for HMMs – The Urn and Ball Model

- \( n \) urns containing colored balls
- \( \nu \) distinct colors
- Each urn has a (possibly) different distribution of colors

Observation sequence generation algorithm:
1. (Behind the curtain) Pick initial urn according to some random process.
2. (Behind the curtain) Randomly pick ball from the urn.
3. Show it to the audience and put it back.
4. (Behind the curtain) Select another urn according to random selection process associated with the urn.
5. Repeat step 2 and 3.
Example for HMMs – The Urn and Ball Model

Why is this an HMM?
• Current urn: **Not observable state**
• Current ball / the sequence of balls: **Observation sequence**
• Distribution of balls in each urn: **Emission probabilities**
• Jump from urn to urn: **Transition probabilities**

The term "**hidden**" refers to seeing observations and drawing conclusions without knowing the **hidden** sequence of states (urns)
Formal Definition of Hidden Markov Models

A Hidden Markov Model $\lambda=(A, B, \pi)$ is a five-tuple consisting of:

- **$S$**: The set of states $S=\{s_1, s_2, ..., s_n\}$ – $n$ is the number of states
- **$\pi$**: The initial probability distribution, $\pi(s_i) = P(q_1 = s_i)$ probability of $s_i$ being the first state of a sequence
- **$A$**: The matrix of state transition probabilities: $1 \leq i, j \leq n$
  
  $A=(a_{ij})$ with $a_{ij} = P(q_{t+1} = s_j | q_t = s_i)$ going from state $s_i$ to $s_j$
- **$B$**: The set of emission probability distributions/densities, $B=\{b_1, b_2, ..., b_n\}$ where $b_i(x)=P(o_t = x | q_t = s_i)$ is the probability of observing $x$ when the system is in state $s_i$
- **$V$**: Set of symbols -- $v$ is the number of distinct symbols
  
  The observable feature space can be discrete: $V=\{x_1, x_2, ..., x_v\}$, or continuous $V=\mathbb{R}^d$
Some Properties of Hidden Markov Models

- For the initial probabilities we have: $\sum_i \pi(s_i) = 1$
- Often things are simplified by $\pi(s_1) = 1$, and $\pi(s_{i>1}) = 0$
- Obviously: $\sum_j a_{ij} = 1$ for all $i$
- Often: $a_{ij} = 0$ for most $j$ except for a few states
- When $V = \{x_1, x_2, ..., x_v\}$ then $b_i$ are discrete probability distributions, the HMMs are called **discrete HMMs**
- When $V = \mathbb{R}^d$ then $b_i$ are continuous probability density functions, the HMMs are called **continuous (density) HMMs**
- In ASR, we mostly use continuous HMMs. Often the emission probabilities are given by Gaussians.
- Basically, each classifier which provides probabilities or densities can be combined with an HMM.
- For simplicity, most upcoming examples show discrete HMMs.
Some HMM Terminology

The most ambiguously used term is the "model", which can be one of:

• **A Hidden Markov Model** =
  the defined five-tuple

• **The model of a state** =
  the combination of HMM parameters that describe the properties of an HMM state (different states can have the same model)

• *The acoustic model* =
  combination of all parameters of recognizer describing all acoustic features
  (e.g. the parameters of the Gaussians in the continuous case)

• **An (acoustic) model** =
  combination of the parameters that describe acoustic features of a specific unit of speech (e.g. of a sub-phonemes)

• **The language model** =
  combination of all parameters describing probabilities of word sequences
The “Trellis”

A graph showing the states over time with observations.
Some Typical HMM-Topologies

- Linear model:

- Left-to-right model:

- Alternative paths:

- Bakis model:
  every state has transition to self or successor or successor of successor

- Ergodic model:
  every state has transitions to every other state
Some Examples for HMM (-Topologies)

Applications:
Simulation and analysis of complex stochastic systems (weather, traffic, queues); recognition of dynamic patterns (speech, handwriting, video).
Typical Questions

• A magician draws balls from urns behind the curtain, the audience sees the observations sequence
  \[ O = (o_1, o_2, ..., o_T) \]

• Your friend told you about two sets of urns and drawing patterns = models \( \lambda_1 = (A, B, \pi) \) \( \lambda_2 = (A, B, \pi) \) the magician usually uses

• Assume you have an efficient algorithm to compute \( P(O|\lambda) \)
  1. Compute \( P(O|\lambda) \) for both models, which of the models \( \lambda_1 \) or \( \lambda_2 \) was more likely to be used by the magician
  2. Given one model, find the “optimal” aka most likely state sequence that would produce the observation
  3. Find a new model \( \lambda' \) such that \( P(O|\lambda') > P(O|\lambda) \)
Thanks for your interest!