Introduction

Basic definitions

Central tasks and algorithms concerning HMMs

Isolated Word Recognition

Recognition/Training architecture

Some speech details

- HMM states in speech recognition
- HMM observations in speech recognition

Sequential Data

With sequential data there are two kinds of tasks you might want to accomplish:

- assign the data sequence a single label (chosen from some finite set of alternatives)
- map the data sequence into a sequence of labels (again each individual label is chosen from some finite set of alternatives)

isolated word recognition takes a sequence of acoustic data as input and produces a guess as the word which was spoken: it assigns a single label to the input data sequence

continuous speech recognition takes a sequence of acoustic data as input, and produces as output a sequence of recognised words; it assigns a sequence of labels to the input data sequence

Hidden Markov Models are a technique often used for both kinds of task.
Hidden Markov Models

**Introduction**

Pre-eminent application of Hidden Markov Models is in Speech Recognition and they were developed by researchers in that area, but they are used quite widely in rather different areas such as

- 'Part-of-Speech' tagging, 'Named Entity Recognition'
- also used in (micro-)Activity Recognition, interpreting body-worn accelerometers
- also used in (macro-)Activity Recognition, interpreting sensors placed in an environment
- also used in Genomics, analysing sequences
- etc

**Basic definitions**

A so-called Markov Model is a particular way to assign probabilities to sequences, where each element of the sequence is drawn from finite set of alternatives, conventionally called states.

Notationally we will use $s$ for such sequences, and $V_s$ for the set of alternatives available at each point. If there are $N$ alternatives its often convenient just to represent $V_s$ with the set of integers $1 \ldots N$.

A so-called Hidden Markov Model is a particular way to assign probabilities to pairs of equal length sequences, $(s, o)$; each element of $o$ is drawn from some set of possible so-called observations $V_o$.

$V_o$ might be a finite set (like $V_s$) and if there are $M$ alternatives, its handy to represent them with the integers $1 \ldots M$. But in many applications each observation is a vector of real numbers.

Note the MT models were also concerned with pairs of sequences; while there is some overlap with HMMs, in MT case the sequences could be of unequal length, and they also involved a hidden alignment, which is absent from HMMs: $s_t$ goes with $o_t$.

**Weather Example**

The weather/ice-cream HMM (from J&M p212)

- states: indicate hot (1) and cold (2) weather on a given day
- observation symbols: 1, 2, or 3 ice-creams bought
- eg. $P(\text{hot at } t \mid \text{hot at } t-1) = 0.7$ : so hot persists somewhat
- eg. $P(1 \text{ ice cream at } t \mid \text{hot at } t) = 0.2$
  $P(3 \text{ ice cream at } t \mid \text{hot at } t) = 0.4$
  so high ice cream consumption likely on hot day

---

1. in turn from Jason Eisner
Hidden Markov Models

Basic definitions

this weather example provides a model which is intended to assign probabilities
to state-obs sequences like:

\[
\begin{align*}
\text{s:} & \quad \text{hot} \quad \text{hot} \quad \text{cold} \quad \text{cold} \quad \text{hot} \quad \text{hot} \quad \text{hot} \\
\text{o:} & \quad 2 \quad 3 \quad 1 \quad 1 \quad 3 \quad 2 \quad 2
\end{align*}
\]

\[
\begin{align*}
\text{s:} & \quad \text{cold} \quad \text{cold} \quad \text{hot} \quad \text{hot} \quad \text{cold} \quad \text{hot} \quad \text{hot} \quad \text{hot} \\
\text{o:} & \quad 1 \quad 1 \quad 2 \quad 3 \quad 1 \quad 3 \quad 2 \quad 2
\end{align*}
\]

etc

Word Recognition example

the state transition part of a possible HMM for the word 'three'

\[
\begin{align*}
\text{th}_1 & \quad \text{th}_2 & \quad \text{th}_3 & \quad r_1 & \quad r_2 & \quad r_3 & \quad iy_1 & \quad iy_2 & \quad iy_3
\end{align*}
\]

- the phone sequence: \(\text{th}, \ r, \ iy\) is a pronunciation of 'three'
- each phone represented by 3 so-called sub-phone states.
- observations in this case would be would be 10 millisecond snap shots of acoustic data, each one a vector of real numbers
- more details later

The HMM formula

\[
P(\text{o}_{1:T}, \text{s}_{1:T}) = P(\text{s}_1)P(\text{o}_1 | \text{s}_1) \times \prod_{t=2}^{T} P(\text{s}_t | \text{s}_{t-1})P(\text{o}_t | \text{s}_t)
\]

If \(\text{o}_{1:T}, \text{s}_{1:T}\) are regarded as \(2T\) variables the above formula can be derived by making particular conditional independence assumptions about them. Essentially these are

- \(s_t\) depends only on \(s_{t-1}\): nothing prior to that matters
- \(o_t\) depends only on \(s_t\): nothing prior to that matters
The HMM formula goes hand in hand with a characteristic view of a process by which the sequence $s_1:o_1:t$ is generated (its so-called ‘generative story’)

this step-by-step generation of the states/observations pair $s_1:o_1:t$ is shown in the next slides

\[
p(s_1) p(o_1|s_1)
\]

\[
p(s_1) p(o_1|s_1) \times p(s_2|s_1)
\]
**Hidden Markov Models**

**Basic definitions**

\[ p(s_1)p(o_1|s_1) \times p(s_2|s_1)p(o_2|s_2) \]

\[ p(s_1)p(o_1|s_1) \times p(s_2|s_1)p(o_2|s_2) \times \cdots \times p(s_{t-1}|s_{t-2})p(o_{t-1}|s_{t-1}) \]

\[ p(s_1)p(o_1|s_1) \times p(s_2|s_1)p(o_2|s_2) \times \cdots \times p(s_{t-1}|s_{t-2})p(o_{t-1}|s_{t-2}) \times p(s_t|s_{t-1}) \]
Hidden Markov Models

Basic definitions

The standard parameters \((\pi, A, B)\)

- all the probs required for an HMM are standardly given as a triple \((\pi, A, B)\) of parameters, concerning starts, transitions and observations:

\[
\text{an HMM triple } (\pi, A, B)
\]

\[\pi: \text{assigns to state } i \text{ the prob } P(s_1 = i)\]

\[A: \text{assigns to states } i, j \text{ the prob } P(s_t = j | s_{t-1} = i), \text{ abbreviated } a_{ij}\]

\[B: \text{assigns to state-obs } i, k \text{ the prob } P(o_t = k | s_t = i), \text{ abbreviated } b_i(k)\]

- the joint prob can be written in terms of the parameters \(\pi, A, \) and \(B\):

\[
P(o_{1:T}, s_{1:T}) = \pi(s_1) b_i(o_1) \prod_{t=2}^{T} a_{s_{t-1}s_t} \times b_i(o_t)
\]

or

\[
P(o_{1:T}, s_{1:T}) = \pi[s_1] B[s_1, o_1] \prod_{t=2}^{T} A[s_{t-1}, s_t] \times B[s_t, o_t]
\]

Central tasks and algorithms concerning HMMs

Three tasks to be tackled for HMMs

- although the HMM formula simply define a probability \(P(s, o)\), the state-sequence \(s\) is generally thought of as hidden, with the observation sequence \(o\) as the evidence generated from this hidden source.

- ('best path') one natural question given an HMM is the question of determining from the observation sequence \(o\) the most probable hidden state sequence \(s\), and thereby assigning a sequence of labels to an sequence of inputs

- ('total probability') by summing out the hidden state sequence an HMM can also assign each observation sequences \(o\) a probability. Using this to choose between rival HMMs is a way to assign a single label to a sequence of inputs

- ('training') relatedly, different parameter settings of an HMM make a corpus of observations sequences \(o^1 \ldots o^T\) more and less likely, and invites the consideration of EM as a parameter learning mechanism

- we will look mostly at the latter two

Total Observation Probability

By summing over all \(s_{1:T}\), a probability is defined for a particular observation sequence \(o_{1:T}\):

\[
P(o_{1:T}) = \sum_{s_{1:T}} P(o_{1:T}, s_{1:T})
\]

(1)

- this is vital in applications which use HMMs to categorise an observation sequence ie. to assign one label from a particular set

- the key is to give each possible category \(C_i\) its own HMM \(M_i\). Given an observation \(o\), the category \(C_i\) is picked whose model \(M_i\) maximises \(P(o; M_i)\)

\[
\text{pick } M_i = \arg \max \limits_{M_i} P(o; M_i)
\]

- The EM-based parameter estimation procedure for HMMs also makes vital use of \(P(o_{1:T})\)
Total prob contd

- $N$ possible states, $N^T$ possible state sequences for $o_{1:T}$: not feasible to simply enumerate the possible state sequences.
- The forward algorithm is a dynamic programming algorithm to avoid this.
- *Forward algorithm* computes a time and state dependent quantity $\alpha_t(i)$.

$$\alpha_t(i) = \sum_{s_1:s_t = i} [P(s_{1:t}, o_{1:t})]$$

or more compactly

$$\alpha_t(i) = P(o_1 \ldots o_t, s_t = i)$$

- if $T$ is the total length of the observation sequence, the total observation probability is obtained by summing the final $\alpha_T(i)$ values.
- Details later.

Training

- so for example can do classification tasks via the total probability *forward* algorithm.
- ... but nothing can be done till model parameters are known.
- there is a famous training algorithm, the *Baum-Welch* algorithm, which adjusts initial estimates of the probabilities given a corpus of observation sequences.
- this is an efficient implementation of *Expectation Maximisation* approach to HMMs.
- in diverse HMM application areas, this training algorithm has been found to yield HMMs which can then be practically be deployed.
- Details later.

Isolated Word Recognition/training architecture

- each possible word $w_i$ has its own HMM $M_i$, whose states output (representations of) acoustic data snap-shots.
- given an acoustic observation sequence $o_{1:T}$, the word $w_i$ is picked whose model $M_i$ maximises $P(o)$, as calculated by forward algorithm.
- eg. for recognising words one, two, three

\[
\begin{align*}
\text{Unknown } O & = \text{ [ ] [ ] [ ] [ ] [ ] [ ] [ ] [ ] [ ] [ ] [ ] [ ] } \\
\text{Choose Max: } & \text{ P(O|M_1) \quad P(O|M_2) \quad P(O|M_3)}
\end{align*}
\]

Training Examples

<table>
<thead>
<tr>
<th></th>
<th>one</th>
<th>two</th>
<th>three</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>[ ] [ ] [ ]</td>
<td>[ ] [ ] [ ]</td>
<td>[ ] [ ] [ ] [ ]</td>
</tr>
<tr>
<td>2</td>
<td>[ ] [ ] [ ]</td>
<td>[ ] [ ] [ ]</td>
<td>[ ] [ ] [ ] [ ]</td>
</tr>
<tr>
<td>3</td>
<td>[ ] [ ] [ ]</td>
<td>[ ] [ ] [ ]</td>
<td>[ ] [ ] [ ] [ ]</td>
</tr>
</tbody>
</table>

*Estimate Models:*

- $M_1$
- $M_2$
- $M_3$
HMM states in ASR: phones

- In the first instance, each word to be recognised is assigned possible pronunciations, with each pronunciation represented as a phone sequence.
- A phone is a more-or-less linguistic notion. Roughly each phone stands for a particular configuration of larynx, tongue, lips etc which creates a particular (i.e. word distinguishing) sound.
- The widely used CMU pronouncing dictionary, has 39 'phones' and is based on the ARPAbet symbol set developed for speech recognition uses. The table that follows lists the 39 phones and gives example pronunciations of 39 words using them.

<table>
<thead>
<tr>
<th>Arpabet</th>
<th>Example</th>
<th>Translation</th>
</tr>
</thead>
<tbody>
<tr>
<td>AA</td>
<td>odd</td>
<td>AA D</td>
</tr>
<tr>
<td>AE</td>
<td>at</td>
<td>AE T</td>
</tr>
<tr>
<td>AH</td>
<td>hut</td>
<td>HH AH T</td>
</tr>
<tr>
<td>AO</td>
<td>ought</td>
<td>AO T</td>
</tr>
<tr>
<td>AW</td>
<td>cow</td>
<td>K AW</td>
</tr>
<tr>
<td>AW</td>
<td>hide</td>
<td>HH AH Y D</td>
</tr>
<tr>
<td>B</td>
<td>be</td>
<td>B IY</td>
</tr>
<tr>
<td>CH</td>
<td>cheese</td>
<td>CH IY Z</td>
</tr>
<tr>
<td>D</td>
<td>dee</td>
<td>D IY</td>
</tr>
<tr>
<td>DH</td>
<td>thee</td>
<td>DH IY</td>
</tr>
<tr>
<td>EH</td>
<td>Ed</td>
<td>EH D</td>
</tr>
<tr>
<td>ER</td>
<td>hurt</td>
<td>HH ER T</td>
</tr>
<tr>
<td>EY</td>
<td>ate</td>
<td>EY T</td>
</tr>
<tr>
<td>F</td>
<td>fee</td>
<td>F IY</td>
</tr>
<tr>
<td>G</td>
<td>green</td>
<td>G R IY N</td>
</tr>
<tr>
<td>HH</td>
<td>he</td>
<td>HH IY</td>
</tr>
<tr>
<td>IH</td>
<td>it</td>
<td>IH T</td>
</tr>
<tr>
<td>IY</td>
<td>eat</td>
<td>IY T</td>
</tr>
<tr>
<td>JH</td>
<td>gee</td>
<td>JH IY</td>
</tr>
<tr>
<td>K</td>
<td>key</td>
<td>K IY</td>
</tr>
<tr>
<td>L</td>
<td>lee</td>
<td>L IY</td>
</tr>
<tr>
<td>M</td>
<td>me</td>
<td>M IY</td>
</tr>
<tr>
<td>N</td>
<td>knee</td>
<td>N IY</td>
</tr>
<tr>
<td>NG</td>
<td>ping</td>
<td>P IY NG</td>
</tr>
<tr>
<td>OW</td>
<td>oat</td>
<td>OW T</td>
</tr>
<tr>
<td>OY</td>
<td>toy</td>
<td>T OY</td>
</tr>
<tr>
<td>P</td>
<td>pee</td>
<td>P IY</td>
</tr>
<tr>
<td>R</td>
<td>read</td>
<td>R IY D</td>
</tr>
<tr>
<td>S</td>
<td>sea</td>
<td>S IY</td>
</tr>
<tr>
<td>SH</td>
<td>she</td>
<td>SH IY</td>
</tr>
<tr>
<td>T</td>
<td>tea</td>
<td>T IY</td>
</tr>
<tr>
<td>TH</td>
<td>theta</td>
<td>TH EY T AH</td>
</tr>
<tr>
<td>UH</td>
<td>hood</td>
<td>HH UH D</td>
</tr>
<tr>
<td>UW</td>
<td>two</td>
<td>T UW</td>
</tr>
<tr>
<td>V</td>
<td>vee</td>
<td>V IY</td>
</tr>
<tr>
<td>W</td>
<td>we</td>
<td>W IY</td>
</tr>
<tr>
<td>Y</td>
<td>yield</td>
<td>Y IY L D</td>
</tr>
<tr>
<td>Z</td>
<td>zee</td>
<td>Z IY</td>
</tr>
<tr>
<td>ZH</td>
<td>seizure</td>
<td>S IY ZH ER</td>
</tr>
</tbody>
</table>

HMM states in ASR contd: Sub-phones

- Conceivably the states of an HMM for a word could be identified with phones, so an HMM for three might have 3 states th, r, iy.
- But better results seem to be obtained by breaking every phone into sub-phones, basically giving each phone its own 3 state HMM, with the so-called Bakis topology, as illustrated by the following picture showing just the state-to-state transition structure.

- So topology is linear left-to-right structure, and allows self-loops.
- So the HMM for three might look like:

HMM observations in speech recognition

- The observations are snapshots of acoustic data, typically covering 10 ms.
- Sound physically is variation in air pressure over time (left, vowel in she).
- In ASR, instead of pressure-vs-time a so-called spectral representation is used. This is based on the fact that a complex periodic function over time can be represented as a superposition of simple waves of particular frequencies and amplitudes (= energy) known as its spectrum (see picture to the right).
- For ASR, t is moved on in small steps (eg. 10 ms) and at each t a spectral slice is calculated from a small window of the raw signal around t. Each slice is a vector of numbers.
HMM observations contd: Spectral slices for phone 'iy'

- A visualisation of a sequence of spectra based on the vowel sound in *three* (the 'phone' iy), so with vertical axis = frequency (up to 5000Hz), horizontal axis = time, darkness = amount of a frequency component
- the observations for ASR are vectors representing vertical strips in this picture

Time variation in a phone

- change is happening at pretty minute timescales in speech. Having several sub-phones within a phone gives the model a better chance to capture this.
- for example the observations with the two phone utterance *Ike*, pronounced [AY K] are:

- it is visible that the nature of the observations changes during the AY phone