Boğaziçi University
Electrical & Electronic Engineering
Fall 2003 EE 473 Digital Signal Processing
Term Project

Connected-Word Speech Recognition Application with HTK

Submitted by:
Can Işın
Çağdaş Kayra Akman

Submitted to: Assoc. Prof. Levent Arslan

20.01.2004

Theory

A Markov Chain is a way of representing probabilistic processes. In a Markov Chain a number of interconnected states are defined. And between these states state transition probabilities are defined. These probabilities are usually expressed in matrix form where, an element a_{ij} corresponds to the probability

$$a_{ij} = P[q_t = S_j | q_{t-1} = S_i],$$

the probability of the process transitioning to state j, given that it was in state i in the previous observation interval. A diagonal element of the state transition probabilities matrix represents the probability of the process remaining in its previous state for another observation interval. The last parameters of a Markov Chain are the initial state probabilities

$$\pi_i = P[q_1 = S_i],$$

Certain restrictions may be placed upon the state transition probabilities matrix of a Markov Chain. Among these the most significant one for our task is the definition of a left to right chain. A left to right chain has the property

$$a_{ij} = 0, \quad j < i$$

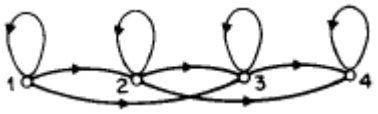
imposed on its transition probabilities. In such a chain the states always transition from left to right with time. This class of Markov Chains is important to us, because the observations from a speech process have this property of changing in a particular way in time. For such a chain another constraint is obviously

$$\pi_i = \begin{cases} 0, & i \neq 1 \\ 1, & i = 1 \end{cases}$$

Furthermore a third constraint may be applied to L-R Markov Chains, the above definitions allow a chain where it is possible to jump from any one state to any other as long as it is in the left to right direction, this, in many cases, is not desirable, then jumps longer than a certain constant may be disallowed:

$$a_{ij} = 0, \quad j > i + \Delta$$

These constraints together produce a state transition probabilities matrix with all the entries below the diagonal filled with zeros, and only the entries on the diagonal and along Δ more diagonal lines right above the diagonal are allowed to be non-zero.



$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} & 0 \\ 0 & a_{22} & a_{23} & a_{24} \\ 0 & 0 & a_{33} & a_{34} \\ 0 & 0 & 0 & a_{44} \end{bmatrix}.$$

The idea of modeling speech with Hidden Markov Models uses these Markov Chains, but as the name implies the Markov Chains are hidden in this representation. The idea is that the Markov Chain process is not suitable for the representation of the speech process. So a more complicated, doubly embedded stochastic process is defined. This is the HMM, in the Markov Chains the observable event was the state of the process, in the HMM the actual state is hidden from the observer, and then the observable event is defined to be a probabilistic function of the (hidden) state. So for the HMM each state is matched with a pdf (continuous or discrete) of producing a certain observable event.
So an HMM process results in two sequences being produced, the observable events sequence O, and the sequence of the hidden states Q that spawned these events. This sequence of states is hidden in the sense that an observation O_i may have been spawned while system was in any state capable of spawning it. Of course determining the most likely sequence of states that led to the observed sequence of events O is one of the most important problems HMM's. At each time interval an HMM transitions from one state to another (possibly back to the same state) according to the state transition probabilities matrix, and upon arrival at the new state spawn an observable event according to the pdf associated with this new state.
If we consider the set of observable events to be discrete, this said pdf may be represented by

$$b_j(k) = P[v_k \text{ at } t | q_t = S_j],$$

We had previously mentioned the initial state probabilities, π_i, compromising the π vector, and the state transition probabilities, a_{ij}, making up the A matrix, with Markov Chains, in addition to the $b_j(k)$,making up the B vector, two other parameters N and M, the number of states in the model and the number of discrete observable events, respectively (actually the sizes of the aforementioned A matrix and the B vector, completely define and HMM, we use the symbol

$$\lambda = (A, B, \pi)$$

to indicate completely parameterized HMM.

For the HMM's a set of three very important problems are defined, the solutions to which are used as the algorithms of an HMM based speech recognizer. These problems are:

(i) Given an observation sequence O and an HMM λ, how can we best (most efficiently) compute the probability of this observation sequence being emitted from this model,
$P(O|\lambda)$

(ii) Again given the observation sequence O and the model λ, how can we choose the most likely state sequence Q? And what is the most meaningful way of defining the "most likely" condition?

(iii) Given an observation sequence O, how do we adjust λ so that $P(O|\lambda)$ is maximized?

Application

We designed a connected-word speech recognition application using Hidden Markov Models Tool Kit (HTK) and following the third chapter of the HTK Book provided with the toolkit.
The system specified in the tutorial was a phoneme-based recognition system with "mixture Gaussian tied-state triphones". We modified this system into a word-based system by introducing each possible word in the dictionary as individual phonemes. The dictionary was composed of the Turkish words from "sıfır" to "dokuz", "on" to "doksan" and "yüz". An at most three digit number or a sequence of individual digits can were to recognized.

The grammar file is as follows:

```
$birler = BIR | IKI | UC | DORT | BES | ALTI | YEDI | SEKIZ | DOKUZ;

$birler2 = IKI | UC | DORT | BES | ALTI | YEDI | SEKIZ | DOKUZ;

$onlar  = ON | YIRMI | OTUZ | KIRK | ELLI | ATMIS | YETMIS | SEKSEN | DOKSAN;

$yuzler = [$birler2] YUZ;


$turkce = SIFIR | $birler | $onlar | $yuzler | ($onlar $birler) | ($yuzler $birler) | ($yuzler $onlar) |
($yuzler $onlar $birler);

(SENT-START ( $turkce | <(SIFIR | $birler)> ) SENT-END)
```

The dictionary file is as follows:

```
ALTI            p6      sp
ATMIS           p60     sp
BES             p5      sp
BIR             p1      sp
DOKSAN          p90     sp
DOKUZ           p9      sp
DORT            p4      sp
ELLI            p50     sp
IKI             p2      sp
KIRK            p40     sp
ON              p10     sp
OTUZ            p30     sp
SEKIZ           p8      sp
SEKSEN          p80     sp
SENT-END []     sil     sp
SENT-START []   sil     sp
SIFIR           p0      sp
SILENCE         sil
UC              p3      sp
YEDI            p7      sp
YETMIS          p70     sp
```

```
YIRMI           p20     sp
YUZ             p100    sp
```

The grammar file is converted into a word network file using HPARSE . HSGEN is used to generate random training prompt files. First, a training data of 100 prompts, then 500 and 1000 are generated. The final system is based on to the training data recorded using 1000 items prompt file with HSLAB. The same file is used to compose word-based *Master Label File* and together with HLED to compose phoneme-based *Master Label File*, which in our case defines single words as phonemes to HTK.

First couple of lines of the prompt file is as follows:

```
1. BIR
2. IKI YUZ IKI
3. ATMIS
4. UC IKI BIR IKI ALTI ALTI UC BIR BES
5. DOKUZ DOKUZ DORT IKI IKI UC BIR
6. DORT YUZ OTUZ DORT
7. IKI YUZ YETMIS
8. ELLI
9. DOKUZ YUZ IKI
10. YETMIS IKI
```

HCOPY is used to extract MFCCs from the waveform files. Afterwards HCOMPV is used to let all variance and mean values Gaussians in HMMs to be the same. A *proto* file is used to specifty reasonable initial values for variance, means, transition probability matrix as follows:

```
~o <VecSize> 39 <MFCC_0_D_A>
~h "proto"

<BeginHMM>

        <NumStates> 17
        <State> 2
                <Mean> 39
                0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
                <Variance> 39
                1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0
1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0
.....
        <TransP> 17
        0.0     1.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0
        0.0     0.0     0.0     0.0     0.0     0.0
        0.0     0.0     0.6     0.4     0.0     0.0     0.0     0.0     0.0     0.0     0.0
        0.0     0.0     0.0     0.0     0.0     0.0
.....
0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0
        0.0     0.0     0.0     0.0     0.0
        <EndHMM>
```

HEREST is the main tool to estimate the HM parameters. It was run several times to improve the system parameters. The system is defined by the following files:

- hmmdefs : *Master Macro File* containing HMM parameters for individual words
- macros : contains global options macro and variance floor values macro
- grammar file
- dictionary file
- phoneme list file

Silence models for ***sil*** (Silence) and ***sp*** (short pause) are added using HHED with a script file defining the backward and forward probabilities for the ***sil*** model, and the forward transition probability for ***sp*** model. It also connects the center state of the ***sil*** model with the ***sp*** model.

HEREST is invoked between each intermediate state explained above to increase the robustness of the system with re-estimated system parameters.

To use the system, HVITE is used for running the application in real-time.

Since the system was trained using training data from only one person, the results of recognizing that person's utterances were very high. Especially, numbers of various digits (up to three) are recognized with a slightly higher rate in case the sequences of discrete digits are uttered in a slow manner.

The recognizer can be run with the following MS-DOS Prompt command:

```
hvite -H macros -H hmmdefs -w wdnet.txt -p 0.0 -s 5.0 -C liveconfig.txt dict.txt phonelist1.txt,
```

provided that the following files are in the same directory:

1. hmmdefs
2. macros
3. wdnet.txt : word network file
4. liveconfig.txt : HVITE configuration file
5. dict.txt : dictionary file
6. phonelist1.txt : list of phonemes
7. hvite.exe : executable of the HTK HVITE tool

References:

Furui, Sadaoki, *Digital Speech Processing, Synthesis, and Recognition*, (New York: Marcel Dekker, 2001)

Rabiner, Lawrence R., "A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition", *Proceedings of the IEEE*, Vol. 77, No. 2, February 1989, pp. 257-286

Young, Steven et.al., *The HTK Book*, http://htk.eng.cam.ac.uk/docs/docs.shtml