CSME 206A Natural Language & Speech Processing **Spring Semester**

Lecture 4: Markov Chains and Hidden Markov Model for ASR

Lecturer: K.R. Chowdhary *: Professor of CS*

4.1 Introduction

The real-world processes generally produce observable outputs which can be characterised as signals. The signals can be discrete in nature (e.g., characters from a finite alphabet, quantized vectors from a code book, etc.), or continuous in nature (e.g., speech signals, temperature measurements, music, etc.). The signal source can be stationary, i.e.,its statistical properties do not change with time, or non-stationary, where signals properties vary with tine. A signal source can be pure (i.e., coming from a signal source) or can be corrupted when coming from many sources (e.g., noise), or it may be distorted in transmission. The problem with such real-world signals is to characterise them in terms of signal models.

The signal models are classified as *deterministic models* and *statistical models*. In the first are sine-wave, or sum of exponentials, and in the second are Gaussian Processes, Poison processes, Markov Processes, and Hidden Markov processes, in addition to others. The assumption in statistical models is that the underlying model can be characterized as a parametric random random process, and parameters of the stochastic processes can be determined/estimated in precise and well-defined manner. The hidden Markov models are referred to as Markov sources probabilistic functions of Markov chains in the communication literature.

Modern architectures for automatic speech recognition (ASR) are mostly software architectures which generate a sequence of word hypotheses from an acoustic signal. Most popular algorithms implemented in these architectures are based on statistical methods

The dominant speech recognition paradigm is known as hidden Markov models (HMM). An HMM is a *doubly stochastic* model, in which the generation of phoneme string for the input sound and the frame-by-frame surface acoustic realizations, are both represented probabilistically as Markov processes. An important feature of frame-based HMM system is that speech segments identified during the search process, rather than explicitly. An alternate approach is to first identify speech segments, then classify the segments and use the segment scores to recognize the words.

We assume that input text words are separated by small pauses. We will introduce the modern speech recognizer components: the Hidden Markov Model (HMMs), the idea of spectral features, and some new algorithms [4].

4.2 Markov Chains

A direct generalization of the scheme of independent trials (in probability theory) is scheme, called Markov chains, which were studied systematically for the first time by the Russian mathematician A. A. Markov.

Imagine that we have a sequence of trials in each of which one and only one of k mutually exclusive events $A_1^{(s)}$, $A_2^{(s)}$, ...,$A_k^{(s)}$ occur, where subscript denotes the number of trial. We say this sequence of trials form a

Markov chain, or more precisely, *simple Markov chain*, provided that the conditional probability that event $A_i^{(s+1)}(i = 1, 2, ..., k)$ will occur in the $s + 1$st trial ($s = 1, 2, ...$), after a known event has occurred in the sth trial. The later solely depends solely on the event that occurred on the sth trial, and it is not modified by supplementary information about the events that occurred in earlier trials [3].

A different terminology is frequently employed in stating the theory of Markov chains, as per that, a physical system S, which at each instant of time, can be in one of the states A_1, A_2, ..., A_k, and alters its states only at times, t_1, t_2, ..., t_n, ... For Markov chains, the probability of passing to some state A_i ($i = 1, 2, ..., k$) at time τ ($t_s < \tau < t_{s+1}$) depends only on the state the system was in at time $t(t_{s-1} < t < t_s)$ and does not change if we learn what its states were at earlier times. Let let us consider the following example.

Example 4.1 *Imagine that a particle located on a straight line moves along the line via random impacts occurring at times t_1, t_2, t_3, ... The particle can be at points with integral coordinates a, $a+1$, $a+2$, ..., b. At points a and b, there are reflecting barriers. Each impact displaces the particle to the right with probability p and to the left with probability $q = 1 - p$ so long as the particle is not located at a barrier. If the particle is at a barrier, any impact will transfer it one unit inside the gap between the barriers. We see that this instance of a particle walk is a typical Markov chain.* □

Homogeneous Markov chains A Homogeneous Markov chains is one in which conditional probability of the occurrence of an event $A_i^{(s+1)}$ in the $(s+1)$th trial, provided that in the sth trial the event $A_i^{(s)}$ occurred, does not depend on the number of the trial. We call this probability as *transition probability* and denote it by a_{ij}, where first subscript always denotes the result of the previous trial, and the second subscript indicates the state into which the system passes in subsequent instant of time.

The total probabilistic picture of possible changes that occur during a transition from one trial to immediately following one is given by a matrix,

$$\pi = \begin{pmatrix} a_{11} & a_{12} & ... & a_{1k} \\ a_{21} & a_{22} & ... & a_{2k} \\ ... & ... & ... & ... \\ a_{k1} & a_{k2} & ... & a_{kk} \end{pmatrix} \tag{4.1}$$

which is compiled of transition probabilities, we will call this matrix as *transition matrix*, i.e., matrix of transition probabilities.

Let us point out what conditions have to be satisfied by the elements of the transition matrix. First of all, being probabilities, they must be nonnegative numbers, i.e., for all i and j,

$$0 \leq a_{ij} \leq 1.$$

Also, from the fact that in the transition from state $A_i^{(s)}$ prior to the $(s+1)$st trial, the system must definitely pass to one and only one of the states $A_j^{(s+1)}$ after the $(s + 1)$ trial, there follows the equation,

$$\sum_{j=1}^{k} a_{ij} = 1, \ (i = 1, 2, ..., k) \tag{4.2}$$

Thus, the sum of the elements of each row of the transition matrix is equal to unity.

Example 4.2 *The Figure 4.1 shows an example of Markov chain. A chain is a sequence; in this figure a chain is, for example, "What is next end", with probability of* 0.25*, for being it a line; for same being a message the probability is* 0.15*, and has probability of* 0.6 *for being its sentence. Similarity, "Which person", "Which is next end", "Who is next word", "Who that person at end", "Who that person", are shown with probability for being line, message, and sentence. They are all in between marked with probabilities p, which is variable and will have different probabilities.*

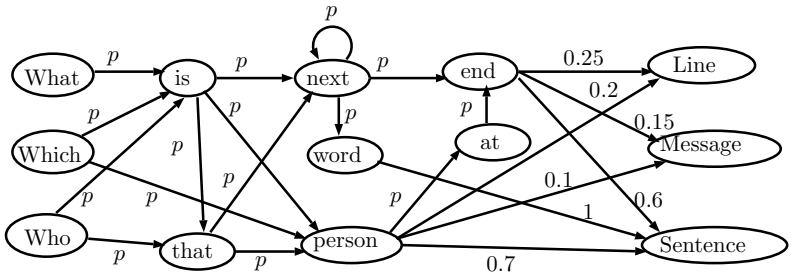

Figure 4.1: Markov Chain.

In other words, we are modelling the sentences, messages, and lines, given a set of vocabulary, in the form of probabilistic models. Hence, a Markov Model is a stochastic model which models temporal or sequential data, i.e., data that are ordered. It provides a way to model dependencies of current information with previous information. It is composed of states, transition schemes between states, and emission of outputs, which may be discrete or continuous. □

There are several goals, which can be accomplished by using Markov models:

- Learn statistics of sequential data
- Do production or estimation
- Recognize patterns.

4.3 Theory of discrete Markov Processes

Consider a system which may be described at any time as being in one of the N distinct states, $S_1, ..., S_N$ as shown in Fig. 4.2, with $N = 5$. At regularly spaced discrete times, the system undergoes a change of states, and may even come back to the same state. This change takes place with certain probabilities associated with each state. Consider that there are time instates $t = 1, 2, 3...$ associated with states. We denote actual state at time t as q_t. A full probabilistic description of this system, in general will require specification of the current state at time t, as well as all the predecessor state [5].

For a special case of a discrete, first order Markov chain, this probabilistic description is truncated to just the current and the predecessor state, i.e.,

$$\begin{aligned} P[q_t = S_j | q_{t-1} = S_i, q_{t-2} = S_k, ...] \\ = P[q_t = S_j | q_{t-1} = Si]. \end{aligned} \tag{4.3}$$

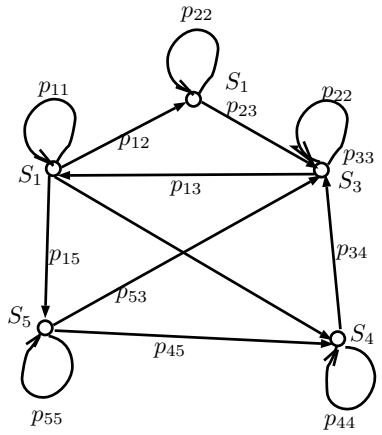

Figure 4.2: Markov Chain with 5-states, $S_1 ... S_5$

Further, we consider only those processes in which the right-hand side of (4.3) is independent of time, thus leading to the set of state transition probabilities a_{ij} of the form,

$$a_{ij} = P[q_t = S_j | q_{t-1} = S_i], \quad 1 \leq i, j \leq N \tag{4.4}$$

With the state transition coefficients having the properties,

$$a_{ij} \geq 0 \tag{4.5}$$

$$\sum_{j=1}^{N} a_{ij} = 1 \tag{4.6}$$

because they obey the standard stochastic constraints.

The above stochastic process could be called as *observable Markov model* since the output of the process is the set of states at each instant of time, where each state corresponds to a physical (observable) event.

Example 4.3 Weather modelling using Markov chain.

To set the ideas, consider a simple 3-state Markov model of the weather. We assume that once a day, (e.g., at noon), the weather is observed as being one of the following:

```
State 1: rain
State 2: cloudy
State 3: sunny
```

We postulate that the weather on day t is characterized by a single one of the three states above, and that the matrix A of state transition probabilities is,

$$A = \{a_{ij}\} = \begin{bmatrix} 0.4 & 0.3 & 0.3 \\ 0.2 & 0.6 & 0.2 \\ 0.1 & 0.1 & 0.8 \end{bmatrix} \tag{4.7}$$

Given that the weather on day 1 (t=1) is sunny (state), we can ask the question: What is the probability (accordingly to this model) that the weather for next 7 days will be "sun-sun-rain-rain-sun-cloudy-sun..."? Stated formally, we define the observation sequence O as $O = \{S_3, S_3, S_3, S_1, S_1, S_3, S_2, S_3\}$ corresponding to t = 1, 2, ..., 8, and we wish to determine the probability of O, given the model. This probability can be expressed (and evaluated) as,

$$\begin{aligned} P(O|Model) =& P[S_3, S_3, S_3, S_1, S_1, S_3, S_2, S_3|Model] \\ =& P[S_3].P[S_3|S_3].P[S_3|S_3].P[S_1|S_3].P[S_1|S_1].P[S_3|S_1].P[S_2|S_3].P[S_3|S_2] \\ =& \pi_3.a_{33}.a_{33}.a_{31}.a_{11}.a_{13}.a_{32}.a_{23} \\ =& 1.(0.8)(0.8)(0.1)(0.4)(0.3)(0.1)(0.2) \\ =& 1.536 \times 10^4 \end{aligned}$$

where we use the notation,

$$\pi_i = P[q_1 = S_i], \qquad 1 \leq i \leq N \tag{4.8}$$

to denote the initial state probabilities.

The other question we can ask is : Given that the model is in a known state, what is the probability that it stays in that state for exactly d days? This can be evaluated as the probability of the observation sequence,

$$O = \{S_{i(1)}, S_{i(2)}, S_{i(3)}, .., S_{i(d)}, S_{j(d+1)} \neq S_i\}, \tag{4.9}$$

Given the model, above is stated as,

$$P(O|Model, q_1 = S_i) = (a_{ii})^{d-1}(1 - a_{jj}) = p_i(d) \tag{4.10}$$

The quantity $p_i(d)$ is the (discrete) probability density function of duration d in state i. This exponential duration density is characteristic of the state duration in a Markov chain. Based on $p_i(d)$, we can readily calculate the expected number of observations (duration) in a state, conditioned on starting in that state as,

$$\begin{aligned} \overline{d_i} =& \sum_{d=1}^{\infty} d p_i(d) \\ =& \sum_{d=1}^{\infty} d(a_{ii})^{d-1}(1 - a_{ii}) \\ =& \frac{1}{1 - a_{ii}} \end{aligned} \tag{4.11}$$

Thus the expected number of consecutive days of sunny weather, according to the model, is $1/(0.2) = 5$; for cloudy it is 2.5; for rain it is 1.67. □

4.4 Hidden Markov Model

So far we have considered Markov models in which each state corresponds to an observable (physical) event. This model is too restrictive to be applicable to many problems of interest. In the following discussions, we extend the concept of Markov models to include the case where the observation is probabilistic function of the state – i.e., the resulting model, called, Hidden Markov Model (HMM) is a doubly embedded stochastic process with an underlying process that is not observable (it is hidden), but can only be observed through another set of stochastic processes that produce the sequence of observations.

A HMM is characterized by following:

1) N, the number of states in the model. Although the states are hidden, for many practical applications there is often some physical significance attached to the states or to sets of states of the model. Generally, the states are interconnected in such a way that any state can be reached from any other state. However, other possible interconnections of the states are often of interest. We denote the individual states as $S = \{S_1, S_2, ..., S_N\}$, and the state at time t as q_t.

2) M, the number of distinct observation symbols per state, i.e., the discrete alphabet size. The observation symbols correspond to the physical output of the system being modelled. For example, for a coin toss experiments the observation symbols are simply the head and tail. We denote that the individual symbols are $V = \{V_1, V_2, ..., V_M\}$.

3) The transition probability distribution $A = \{a_{ij}\}$, where,

$$a_{ij} = P[q_{t+1} = S_j | q_t = S_i], \quad 1 \leq i, j \leq N. \tag{4.12}$$

For the special case where any state can reach any other in a single step, we have $a_{ij} \geq 0$ for all i, j. For other types of HMMs, we would have $a_{ij} = 0$, for one or more (i, j) pairs.

4) The observation symbol probability distribution $B = \{b_j(k)\}$, in state j, where

$$b_j(k) = P[V_k \text{ at } t | q_t = S_j], \quad 1 \leq j \leq N, \quad 1 \leq k \leq M. \tag{4.13}$$

5) The initial state distribution $\pi = \{\pi_i\}$ where,

$$\pi_i = P[q_1 = S_i], \quad 1 \leq i \leq N. \tag{4.14}$$

4.4.1 Algorithm

Given the appropriate values of N, M, A, B, and π, the HMM can be used as a generator to give an observation sequence,

$$O = O_1 \; O_2 \; ... \; O_T \tag{4.15}$$

where O_t is one of the symbols in V, and T is number of observations in sequence. The steps are as follows:

1. Choose initial state $q_1 = S_i$ according to the initial state distribution π.
2. Set $t = 1$.
3. Choose $O_t = V_k$ according to the symbol probability distribution in state S_i, i.e., $b_i(k)$.
4. Transit to a new state $q_{t+1} = S_j$ according to the state transition probability distribution for state S_i, i.e., a_{ij}.
5. Set $t = t + 1$; return to step 3, if $t < T$; otherwise terminate the procedure.

The above procedure can be used as both generator of observations, and as a model for how a given observation sequence was generated by an appropriate HMM.

Note from the above discussions that a complete specification of a HMM requires the specification of two model parameters (N and M), specification of observation symbols, and the specification of three probability symbols A, B, and π. For convenience, we use the compact notation,

$$\lambda = (A, B, \pi) \tag{4.16}$$

to indicate the complete parameter set of the model.

4.4.2 Basic Problems for HMM

Given the form of HMM, there are three basic problems of interest that must be solved for the model to be useful in real-world applications. Following are these problems.

Problem 1: Given the observation sequence $O = O_1\ O_2\ ...O_T$, and model $\lambda = (A, B, \pi)$, how to compute efficiently $P(O|\lambda)$, the probability of the observation sequence, given the model?

This problem is the evaluation problem, namely given a model and a sequence of observations, how do we compute the probability that the observed sequence was produced by the model. We can also view the problem as one of scoring how well a given model matches a given observation sequence. The latter viewpoint is extremely useful. For example, if we consider the case in which we are trying to choose among several competing models, the solution to problem allows us to choose the model which best matches the observation.

The straightforward way of doing this is through enumerating every possible state sequence of length T (the number of observations). Consider one such fixed state sequence,

$$Q = q_1\ q_2\ ...q_T \tag{4.17}$$

where q_1 is the initial state. The probability of observation sequence O for the state sequence (4.17) is,

$$P(O|Q, \lambda) = \prod_{t=1}^{T} P(O_t|q_t, \lambda) \tag{4.18}$$

In the above we have assumed the statistical independence of observation. Thus we get,

$$P(O|Q,\lambda) = b_{q_1}(O_1).b_{q_2}(O_2)...b_{q_T}(O_T). \tag{4.19}$$

The probability of such a state sequence Q can be written as,

$$P(Q|\lambda) = \pi_{q_1} a_{q_1 q_2} a_{q_2 q_3} ... a_{q_{T-1} q_T}. \tag{4.20}$$

Problem 2: Given the observation sequence $O = O_1\ O_2\ ...O_T$, and model $\lambda = (A, B, \pi)$, how to choose a corresponding state sequence $Q = q_1 q_2 ... q_T$ which is optimal in some meaningful sense (i.e., best "explains" the observation)?

This problem is the one in which we attempt to uncover the hidden part of the model, i.e., to find the "correct" state sequence. It should be clear that for all but the case of *degenerate models*, there is no "correct" state sequence to be found. Hence, for practical situations, we usually use an optimality criterion to solve this problem as best possible. Unfortunately, there are several reasonable criteria that can be imposed, and hence the choice of criterion is a strong function of the intended use for the uncovered state sequence. Typical use might be to learn about the structure of the model, to find optimal state sequences for continuous speech recognition, or to get average statistics of individual states, etc.

Viterbi Algorithm: To find the single best state sequence, $Q = \{q_1, q_2, ..., q_T\}$, for the given observation sequence $O = \{O_1 O_2 ... O_T\}$, we need to define the quantity,

$$\delta_t(i) = argmax_{q_1, q_2, ..., q_{t-1}} \quad P[q_1 q_2 ... q_t = i, O_1 O_2 ... O_t | \lambda] \tag{4.21}$$

where $\delta_t(i)$ is the best score (highest probability) along a single path, at time t, which accounts for the first t observations and ends in state S_i [8], [2].

Problem 3: How do we adjust the model parameters $\lambda = (A, B, \pi)$ to maximize $P(O|\lambda)$?

In the Problem 3, we attempt to optimize the model parameters so as to best describe how a given observation sequence comes about. The observation sequence used to adjust the model parameters is called a *training sequence*, since it is used to "train" the HMM. The training problem is crucial one for most applications of HMMs, since it allows us to optimally adapt modern parameters to observed training data – i.e., to create best models for real phenomena.

There is no standard method to analytically solve for the model which maximizes the probability of the observation sequence. In fact, given any finite observation sequence as training data, there is no optimal way of estimating the model parameters. We can however, choose the $\lambda = (A, B, \pi)$ such that $P(O|\lambda)$ is locally maximized using an iterative procedure such as Baum-Welch method or EM (expectation-modification) method [1], or using gradient techniques [6].

4.5 Speech recognition using HMM

Each node in the HMM models spectral characteristics of a quasi-stationary segment of speech. At every time instance (frame of speech), the system either continuous to stay in a state or makes a transition to another state in a probabilistic manner. The $(ij)^{th}$ element of the transition matrix a_{ij}, denotes the probability of

transition from i^{th} state to j^{th} state. The expected duration of state i is $1/(1-a_{ii})$ (see equation (4.11), page no. 4-5). Hence, if probability of loop a_{ii} is zero, the expected duration is 1.

The HMM is doubly stochastic model, such that probability distribution $P(\mathbf{x})$ associated with a state gives the likelihood of a feature vector $\mathbf{x}$ belonging to that state [7].

The popularity of HMM is due to the existence of efficient algorithms for estimation of parameters of the model, i.e., M, N, A, B, π (see page no. 4-6), from the training data, and due to efficient algorithm of recognition.

The HMMs function as probabilistic finite state machines. The model consists of a set of states, and its topology specifies the allowed transitions between them. At every time frame, an HMM makes a probabilistic transition from one state to another and emits a feature vector with each transition [7].

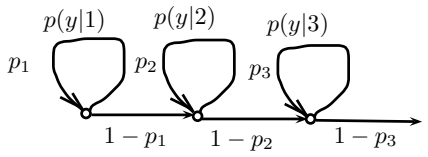

Figure 4.3: Hidden Markov model for a phoneme.

Fig. 4.3 shows an HMM for a phoneme. A set of state transition probabilities–p_1, p_2, and p_3 governs the possible transitions between states. They specify the probability of going from one state at time t to another state at time $t+1$. The feature vectors emitted while making a particular transition, represent the spectral characteristics of the speech at that point, which vary corresponding to different pronunciations of the phoneme. A probability distribution or probability density function models this variation. The functions $P(y|1)$, $P(y|2)$, and $P(y|3)$ – could be different for different transitions. Typically, these distributions are modelled as parametric distributions–a mixture of multidimensional Gaussian.

The HMM shown in Fig. 4.3 consists of three states. The phoneme's pronunciation corresponds to starting from the first state and making a sequence of transitions to eventually arrive at the third state. The duration of the phoneme equals the number of time frames required to complete the transition sequence. The three transition probabilities implicitly specify a probability distribution that governs this duration. If any of these transitions exhibits high self-loop probabilities, the model spends more time in the same state, consequently taking longer to go from the first to the third state. The probability density functions associated with the three transitions govern the sequence of output feature vectors.

A fundamental operation is the computation of the likelihood that an HMM produces a given sequence of acoustic feature vectors. For example, assume that the system extracted T feature vectors from speech corresponding to the pronunciation of a single phoneme, and that the system seeks to infer which phoneme from a set of 50 was spoken. The procedure for inferring the phoneme assumes that the ith phoneme was spoken and finds the likelihood that the HMM for this phoneme produced the observed feature vectors. The system then hypothesizes that the spoken phoneme model is the one with the highest likelihood of matching the observed sequence of feature vectors.

If we know the sequence of HMM states, we can easily compute the probability of a sequence of feature vectors. In this case, the system computes the likelihood of the t-th feature vector, y_t, using the probability density function for the HMM state at time t. The likelihood of the complete set of T feature vectors is the product of all these individual likelihoods. However, because we generally do not know the actual sequence of transitions, the likelihood computation process sums all possible state sequences. Given that all HMM dependencies are local, we can derive efficient formulas for performing these calculations recursively.

4.6 Noisy channel model for speech recognition

The speech recognition system treats the acoustic input as noisy channel, i.e, the sentence available is a noisy version of original sentence. In order to "decode" this noisy sentence, we consider all possible sentences and then find out the probability that it is the original sentence. For this, choose the one having maximum probability of likelihood. This approach is shown in Fig. 4.4.

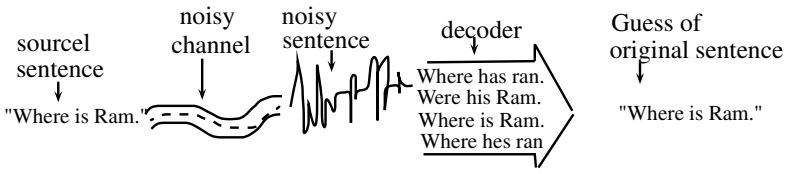

Figure 4.4: Noisy channel model.

The modern speech recognition works by searching huge space of potential "source" sentence, and choosing the one which having the highest probability of generating the noisy sentence. This requires the solution of two problems:

1. Complete metric for best match of noisy and original sentence. This is because these two sentences can never match exactly. Hence, we make use of probabilistic method discussed above.
2. English has large number of sentences, hence use some search technique, which does not require to search the entire space, but still can locate the current correct sentence.

We can treat the acoustic input (observation) O as a sequence of individual symbols (e.g. slices of inputs $(o_1, o_2, o_3, \dots)$, say one symbol for every 10 milli-secs., and represent each slice by *frequency* or *energy* of that slice. Let us assume,

$$O = o_1, o_2, \dots o_T. \tag{4.22}$$

Let the corresponding sentence, which might have caused this sentence, is a string of words:

$$W = w_1, w_2, \dots, w_n \tag{4.23}$$

The probabilistic implementation of the above mentioned intuition, i.e., the probability of occurrence of this sentence, given that acoustic observation O is evidenced, is

$$\hat{W} = argmax_{w \in \mathcal{L}}\ P(W|O) \tag{4.24}$$

where, $\mathcal{L}$ is English Language.

Note that, $argmax_x f(x)$ means, "the x such that $f(x)$ is maximum." The equation 4.24 is guaranteed to give optimal sentence W. For a given sentence W and acoustic sequence O we need to compute $P(W|O)$. The equation (4.24) can be expressed as:

$$\hat{W} = argmax_{w \in \mathcal{L}} \frac{P(O|W) * P(W)}{P(O)}. \tag{4.25}$$

The prior probability $P(W)$ is estimated by n-gram language model. We can ignore the probability $P(O)$, because it is a common denominator for all the sentences. Hence, what we need to compute, reduces to simply:

$$\hat{W} = argmax_{w \in \mathcal{L}} P(O|W) * P(W) \tag{4.26}$$

where, W is most probable sentence, $P(W)$ is prior probability (it is called the *language model*), and $P(O|W)$ is observation likelihood.

We will compute the acoustic model $P(O|W)$ in two steps: first we make assumption that input sequence is a sequence of phones in F rather than sequence of acoustic observations. We will show that probability of these phone automata are special case of HMM, and we will show how to extend these models to give probability of a phone sequence given the entire sentence.

There are two algorithms, that simultaneously compute the likelihood of an observation sequence given each sentence, and give us the most likely sentence. These are *Viterbi* [8] and A^* algorithms. Finally, we will introduce the HMM approach. These steps are shown in Fig. 4.5.

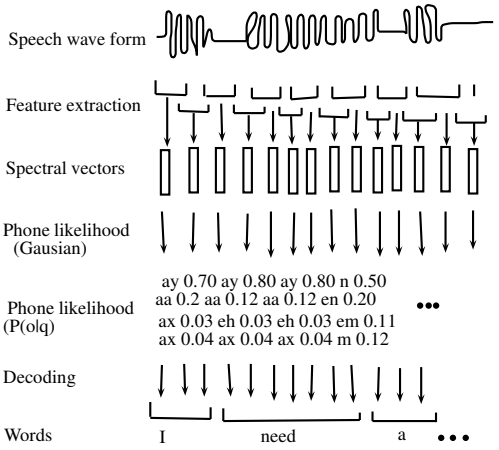

Figure 4.5: Schematic structure of a Speech Recognizer.

4.7 Weighted Automata vs HMM

The weighted automata is also called Markov chains. Here, the automata consists of a sequence of states $q = (q_0 q_1 \dots q_n)$, each corresponding to a phone, and set of transition probabilities between states, a_{01}, a_{12}, a_{13}, encodes the probability of one phone following to other (see Fig. 4.6). To compute the transition probability of a sequence of phones $O = (o_1 o_2 \dots o_t)$, we make use of *forward algorithm*. The Fig. 4.6 shows the Markov chain for the word "need". The figure shows the transition probabilities and a sample observation sequence. The probabilities are 1 unless otherwise specified.

The Fig. 4.6 shows the speech input as sequence of symbols. However, in real world, the speech is sliced, ambiguous, real valued input, called *features* or *spectral features*. The second simplification in the Markov chain model above is that, when we see the input symbols [b], we move into the state [b]. In HMM, we cannot look at the input symbols and know which state to move into. In fact, the input symbols do not uniquely determine the next state. For the weighted Automata or Markov chains (called simple Markov model), we

use a set of *observation likelihood B*. The probability $b_i(o_t)$ is 1 if state i is matched to the observation (i.e., phone) o_t and 0 if they did not match.

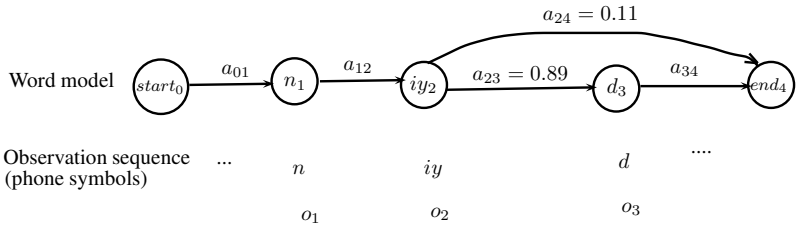

Figure 4.6: Weighted Automata or Markov chain.

An HMM formally differs from a Markov model by adding two more requirements. First, it has a separate set of observable symbols O, which is not drawn from the same alphabet as the state set Q, second, the observation likelihood function B is not limited to the values 1 and 0 (Fig. 4.6 shows n_1 and iy_2 both as 1). In an HMM the probability $b_i(o_t)$ can take any value from 0 to 1.

The Fig. 4.7 shows the HMM for the word *need* and sample observation sequence. The observation sequences are now vectors of spectral features representing the speech signals. Also, we have allowed one state to generate the multiple copies of the same observation, by having a loop on that state. This loop allows HMM to model the variable duration of phones; longer phones requires more loops through the HMM.

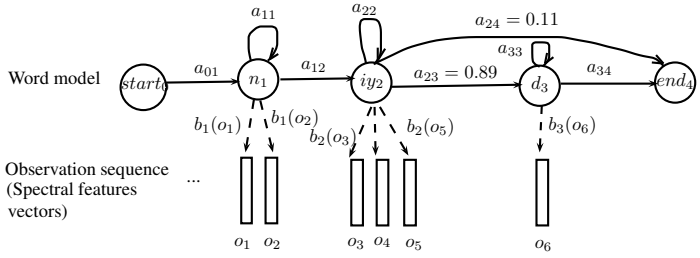

Figure 4.7: An HMM Pronunciation Network for the word *need*.

In summary, we have following parameters for the HMM:

- States: $Q = q_1 q_2 \ldots q_N$, each representing one or more phones,
- Transition probabilities: $A = a_{01} a_{02} \ldots a_{n1} \ldots a_{nn}$. Each a_{ij} represents the probability of transitioning from state i to state j. It is a matrix $[a_{ij}]$ (see equation (4.7), page no. 4-5).
- Observation likelihood: It is a set of observation likelihood $B = b_i(o_t)$, each expressing the probability of an observation o_t being generated from state i (see equation (4.13), page no. 4-6).

Review Questions

1. What is Markov Chain?
2. What is difference between Markov chain and Finite automaton?

3. Does a Markov chain has final state?
4. What is difference between probabilistic FA and Markov chain?
5. What is homogeneous Markov chain?
6. How the probability matrix in Markov chain is different from probability matrix of graph like Bayesian probability distribution?
7. What is discrete Markov Process?
8. Why a Markov chain is not sufficient for probability model, and there is need of Hidden Markov Model?
9. In the above, explain the parameters A, B, M, N, π, discussed above.

Exercises

1. What is noisy channel model for speech recognition? Discuss its characteristics.
2. How a weighted FA is useful for speech recognition? Explain its working for speech recognition.
3. What is Markov chains? Explain its working in general, and as a speech recognizer.
4. Why a Markov chain is not sufficient for general speech recognition? Discuss its limitations.
5. Why Markov chains is considered as superior model for speech recognition, in comparison to simple Markov model? Justify.
6. Explain the working of Fig. 4.7 in detail.
7. Explain the working of Fig. 4.3 in detail.
8. Explain the Viterbi algorithm in your own word.
9. Explain the working of noisy channel for speech recognition.
10. What are the major differences between weighted automata and HMM?

References

[1] Dempster A P et al., *maximum likelihood from incomplete data via the EM algorithm*, *J. Roy. Stat. Soc.*, vol. 39, no. 1, pp. 1-38, 1977.

[2] Forney G D, *The Viterbi algorithm*, *Proc. IEEE*, vol. 61, pp. 268-278, Mar. 1973.

[3] Gnedenko B V, *The theory of probability*, MIR Publisher, Moscow, 1969

[4] Jurafsky D and Martin J, *Speech and Language Processing, 3rd Ed.*, Pearson India, isbn: 3257227892, Nov. 2005.

[5] Lawrence R. Rabiner, *A tutorial on Hidden Markov Models and Selected Applications in Speech Recognition*, *Proc. of IEE*, Vol. 77, No.2 Feb 1989, pp. pp. 257-286.

[6] Levinson S E et al., *An introduction to the application of the theory of probabilistic functions of a Markov process to automatic speech recognition, Bell Syst. Tech. J.,* Vol. 62, no. 4, pp. 1035-1074, Apr. 1983.

[7] Padmanabhan M and Picheny M, *Large-vocabulary speech recognition algorithms*, *Computer*, Vol. 35, No. 4, pp. 42-50, 2002, doi=10.1109/MC.2002.993770

[8] Viterbi A J, *Error bounds for convolutional codes and asymptotically optimal decoding algorithm*, *IEEE Trans. Informat. Theory*, vol. IT-13, pp. 260-269, Apr. 1967.