Hidden Markov Models for Isolated-Word Speech Recognition

6.962 Week 10 Presentation

Irina Medvedev*
Massachusetts Institute of Technology

April 19, 2001

1 Introduction

Not too long ago, speech recognition was characterized under the category of science fiction. In a popular television show, *Knight Rider*, the ability of *Kit*, the car, to communicate with humans seemed to portray unreachable fantasy in the every day life. In the past few years, however, technological advances have made speech recognition a reality.

This week's seminar will cover the use of hidden Markov models (HMM) for speech recognition. We will consider isolated-word speech recognition, which involves recognizing one word at a time, as opposed to continuous speech recognition, where complete phrases and sentences are recognized. The outline of this summary is as follows. Section 2 provides a background of Markov processes, Markov chains, and hidden Markov models. In Section 3, we give a detailed description of the isolated-word speech recognition system. Section 4 provides conclusions and possible extensions to the problem presented here.

2 Markovianity

2.1 Markov Processes

A first-order Markov process $x(t)$ has the characteristic that the current value of the process is the only one necessary to predict the future. All of the memory is captured in the current value, and the past does not contain any additional information about the future. The process $x(t)$ is Markov if for any set of ordered times, $t_1 < t_2 < \cdots < t_N$,

$$p_{(x(t_N)|x(t_1),x(t_2),\ldots,x(t_{N-1}))}(x_N|x_1,x_2,\ldots,x_{N-1}) = p_{(x(t_N)|x(t_1))}(x_N|x_{N-1}). \tag{1}$$

From the above, it becomes clear that the value of a Markov process at time t_N depends only on the value of the process at time t_{N-1}, and is conditionally independent of the process at all times prior to t_{N-1}.

The key quantity that arises in the statistical description of Markov processes is the transition probability density, defined as

*irinam@mit.edu

$$p_{x(t)|x(s)}(x|z) \tag{2}$$

for $s < t$, which describes how the probabilistic densities transition from one time instant to the next. A complete specification of a Markov process consists of the specification of the first-order density, $p_{x(t)}(x)$, together with the transition density, $p_{x(t)|x(s)}(x|z)$ [6].

2.2 Markov Chains

A Markov chain can be used to describe a system which, at any time, belongs to one of N distinct states, $S_1, S_2, \cdots, S_N$. Figure 1 illustrates a Markov chain with 3 states. At regularly spaced times, $t = 1, 2, \ldots$, the system may either change to a different state or remain in the same state, where the state at time t is denoted by q_t. The state transitions are made according to a set of probabilities associated with each state. These probabilities are stored in the state transition matrix

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1N} \\ a_{21} & a_{22} & \cdots & a_{1N} \\ \vdots & \vdots & \ddots & \vdots \\ a_{N1} & a_{N2} & \cdots & a_{NN} \end{bmatrix}, \tag{3}$$

where N is the number of states in the Markov chain. The state transition probabilities are defined as

$$a_{ij} = P[q_t = S_j \mid q_{t-1} = S_i], \quad 1 \leq i, j \leq N \tag{4}$$

and have the properties

$$a_{ij} \geq 0 \tag{5}$$

$$\sum_{j=1}^{N} a_{ij} = 1. \tag{6}$$

This kind of process can be classified as an observable Markov model since the output is the set of states at each instant of time, and the states are observable events [3].

2.3 Hidden Markov Models

In many cases, the Markov model from the above section is too restrictive because we cannot always assume that the states are observable. For situations where the observation is a probabilistic function of the state rather than the state itself, we use Hidden Markov Models, where the states are not observable events [3].

Throughout this paper, we assume a vector observation, $\mathbf{x}$, in each state, where the observations are characterized by continuous probability density functions (pdf), as will be discussed in subsequent sections. The hidden Markov models discussed here are characterized by the following:

1. N, the number of states in the model.

2. The state transition matrix, $\mathbf{A} = \{a_{ij}\}$, as defined in Section 2.2.

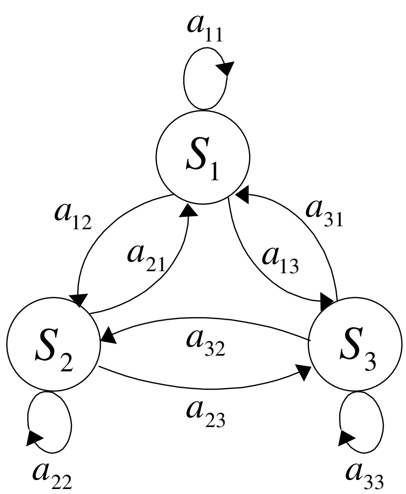

Figure 1: A Markov chain with 3 states

3. The observation probability distribution in state j, $B = \{b_j\}$, where

$$b_j(\mathbf{x}) = P[\mathbf{x} \ at \ t \mid q_t = S_j], \quad 1 \leq i, j \leq N \tag{7}$$

4. The initial state distribution, $\pi = (\pi_1 \ \pi_2 \ \cdots \ \pi_N)$, where

$$\pi_i = P[q_1 = S_i], \quad 1 \leq i \leq N. \tag{8}$$

A complete description of an HMM requires the specification of the number of states, N, as well as the model parameter set, defined as $\lambda = (\mathbf{A}, B, \pi)$.

2.4 Types of HMMs

The Markov model shown in Figure 1 is a fully connected model because every state of the model can be reached from every other state of the model. As described in Section 2.2, this kind of model has a state transition matrix of the form in (3).

For some applications, such as speech recognition, other times of HMMs have been found to be better models than the fully connected model. One such model is the left-right HMM, in which the states can only transition to a higher state or stay in the same state [3]. Additionally, some models impose the no-skip constraint, where the states can only transition to the next state or remain in the same state. An example of the no-skip left-right HMM is shown in Figure 2.

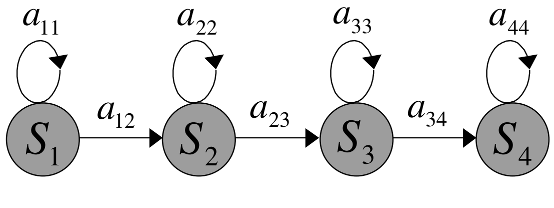

Figure 2: A 4-state left-right HMM with no-skip transitions

The no-skip left-right HMM is the kind we will use for the speech recognition application described in this paper. The state transition matrix for such an HMM becomes

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & 0 & \cdots & 0 \\ 0 & a_{22} & a_{23} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & a_{N-1N} & a_{NN} \end{bmatrix} \tag{9}$$

where the zeros represent illegal transitions of the states.

3 Isolated Word Speech Recognition

This section will address the use of Hidden Markov Models for speech recognition. We focus on isolated word speech recognition, where we attempt to recognize one word at a time. Any speech signal to be processed will be assumed to be of the form: silence - speech - silence.

3.1 Feature Analysis

We assume a discrete-time speech signal obtained by sampling the continuous-time speech signal at 8 kHz. For one second of speech, that amounts to 8000 samples. To decrease complexity, we reduce this number by performing feature analysis to extract observation vectors upon which all subsequent processing will be performed. We use cepstral coefficients as feature observation vectors.

For any vector, $\mathbf{z}$, let $\mathbf{z}^T$ denote the vector transpose. For a discrete-time signal vector, $\mathbf{y} = (y_0 \; y_1 \; \ldots \; y_{V-1})^T$, with discrete Fourier transform (DFT), $\mathbf{Y} = (Y_0 \; Y_1 \; \ldots \; Y_{K-1})^T$, the cepstral coefficients are obtained from the inverse Fourier transform of the logarithm of the power spectral density of $\mathbf{y}$, $S_y(k)$. Since we do not have the exact expression for $S_y(k)$, we use the periodogram estimate, $\hat{S}_y(k) = \frac{1}{V}\left|Y(k)^2\right|$, and the cepstral coefficients are obtained from

$$\mathbf{x}(n) = \frac{1}{K}\sum_{k=0}^{K-1} \log\left(\frac{1}{V}\left|Y(k)^2\right|\right) exp\left(j\frac{2\pi}{K}kn\right), \; n = 0, 1, \cdots, K-1 \tag{10}$$

where $K \geq 2V - 1$, so that the DFT is a linear rather than circular convolution. In speech recognition applications, the original N-dimensional speech vectors are represented by L-dimensional cepstral feature vectors, where $L \ll V$. The zeroth component of the cepstral vector is excluded. One characteristic of cepstral coefficients of speech is that the high-order cepstral components carry speaker dependent pitch information, which is not highly relevant to speech recognition [1]. Therefore, we do not lose much by excluding coefficients beyond the L-th.

Another nice property of cepstral coefficients is that they can be well approximated by a Gaussian probability density function. Additionally, cepstral coefficients have very low cross-correlation values due to the logarithmic function. The HMM assumes that the Markovian states generate the cepstral vectors, where each state represents a Gaussian source with a certain mean vector and diagonal covariance matrix. Thus, each feature vector can be modeled as a sample vector of an L-dimensional Gaussian random vector with mean vector, μ_i, and diagonal covariance matrix, Λ_i, where the index i refers to the state, $1 \leq i \leq N$. A weighted sum of Gaussian distributions, called Gaussian mixtures, can also be used to represent the states, but we will assume a single Gaussian distribution through this paper [1].

Figure 3 shows the process by which the cepstral feature vectors are obtained from a discrete-time speech signal. First, the speech signal is divided into 256-sample segments. Then, we obtain the cepstral coefficients of each segment and keep coefficients $1 - 11$, such that the dimensionality

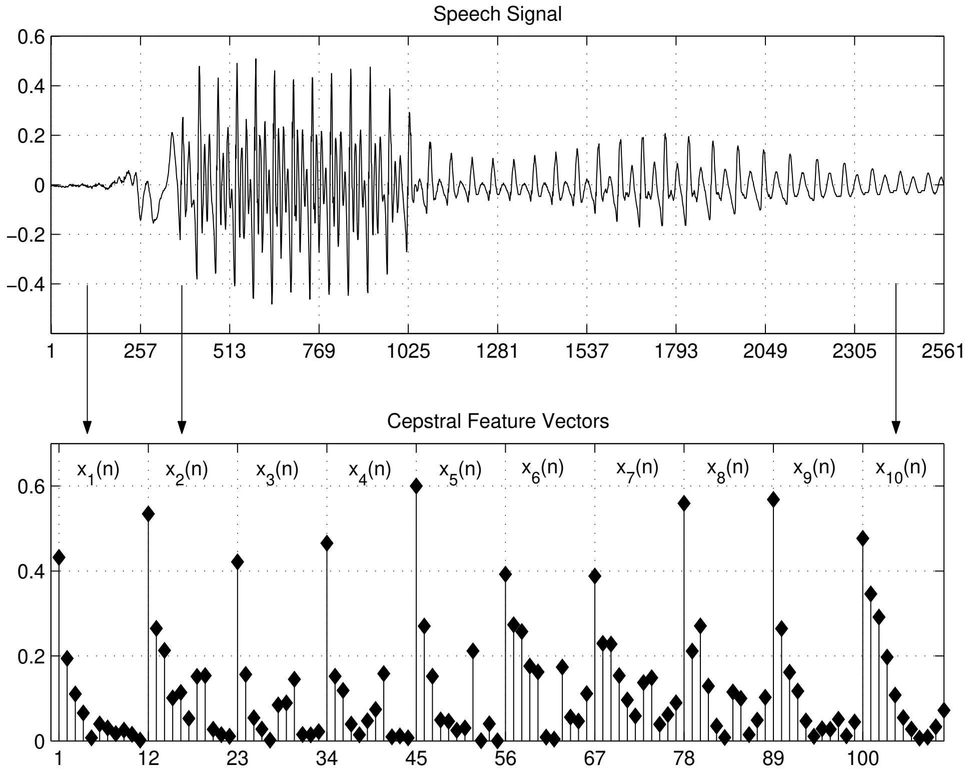

Figure 3: Formulation of Feature Vectors

is reduced by a factor of 256/11. Each 256-sample segment is now represented by a vector of 11 cepstral coefficients.

3.2 Unit Matching

The first goal is to obtain an HMM for each speech recognition unit, where the choice of the speech unit is usually dependent upon the vocabulary size. For large vocabulary speech recognition, the use of sub-word units, called phonemes, is almost required since we would need much fewer models of phonemes than words. For small vocabulary speech recognition, it is practical to consider the word as a basic speech unit and thus, have one HMM for each word in the vocabulary [3]. We will assume the latter model in this paper.

3.3 Training

We now consider an isolated-word speech recognition system for a small vocabulary of M words. Before we proceed, we introduce some notation.

- The observation is the vector $\mathbf{x} = (\mathbf{x}_1\ \mathbf{x}_2\ \cdots\ \mathbf{x}_T)$, where each $\mathbf{x}_t,\ 1 \le t \le T$, represents a cepstral feature vector, as shown in Figure 3, and T is the number of feature vectors for the given observation.
- The state sequence is $\mathbf{q} = (q_1\ q_2\ \cdots\ q_T)$, where each $q_t \in \{S_1, S_2, \cdots, S_N\}$.

- The term *model* will be used interchangeably for the HMM and the parameter set that describes the HMM, λ.
- The index $i,\ 1 \leq i \leq N$, refers to the state.
- The index $k,\ 1 \leq k \leq M$, refers to the word.
- The index $t,\ 1 \leq t \leq T$, refers to the time.

As mentioned earlier, we will have a separate HMM for each word. Each HMM is characterized by the number of states, N, which remains the same for all word models, and the model parameter set, $\lambda_k = (\mathbf{A}, B, \pi),\ 1 \leq k \leq M$, as described in Section 2.3. In Section 3.1, we saw that the cepstral feature vector, $\mathbf{x}_t$, in each state, $S_i,\ 1 \leq i \leq N$, can be modeled as having an L-dimensional Gaussian probability density function

$$p(\mathbf{x}_t \mid q_t = S_i) = \frac{1}{(2\pi)^{L/2} |\Lambda_i|^{1/2}} \exp\left(-\frac{1}{2}(\mathbf{x}_t - \mu_i)^T \Lambda_i^{-1} (\mathbf{x}_t - \mu_i)\right) \tag{11}$$

where μ_i and Λ_i are the mean vector and covariance matrix per state, respectively [2]. Because a Gaussian pdf is completely characterized by the mean and covariance matrix, we can modify the model parameter set to be

$$\lambda_k = (\mathbf{A}, \{\mu_i\}, \{\Lambda_i\}, \pi),\ 1 \leq i \leq N,\ 1 \leq k \leq M.$$

For each word, we need to build an HMM, which consists of obtaining the model parameter set. The process of building the HMMs is called training, where the training is done using spoken utterances of each word. The training procedure is the same for each word, and thus, we drop the subscript k from the parameter set, λ_k, for convenience in this section.

Building an HMM requires determining the model parameter set that maximizes the likelihood of the observation vectors for that word. This criterion can be restated as the maximization of the joint probability of the observation, state sequence, and the model

$$\max_{\lambda} \max_{\mathbf{q}} P(\mathbf{x}, \mathbf{q}, \lambda) \tag{12}$$

which is a double maximization over the state sequence and the model, which are the two degrees of freedom in the maximization. The double maximization can be obtained by maximizing over each variable individually. We obtain the initial state sequence by a process called uniform segmentation, where we try to make the initial state sequence take on a uniform distribution. This entails placing as close as possible to an equal number of feature vectors into each state. Because we are using the left-right HMM with no-skip transitions, the first few feature vectors are always in state one, while the last few are always in state N.

Now that we have the initial state sequence, we maximize over the model, given the state sequence. This maximization results in

$$\max_{\lambda} P(\mathbf{x}, \lambda \mid \mathbf{q}) \tag{13}$$

where the best model given the state sequence is obtained using the Baum-Welch re-estimation formulas, shown in Table 1.

From the table, we observe that the re-estimation formulas are intuitive. First, we introduce a new variable, N_i, defined as the number of feature vectors in state S_i

state transition matrix	$\hat{a}_{ij} = \dfrac{\sum\limits_{t:\ q_{t-1}=S_i, q_t=S_j} 1}{\sum\limits_{t:\ q_{t-1}=S_i} 1},\ 1 \leq i, j \leq N$
mean vector per state	$\hat{\mu}_i = \dfrac{1}{N_i} \sum\limits_{t:\ q_t=S_i} \mathbf{x}_t,\ 1 \leq i \leq N$
covariance matrix per state	$\hat{\Lambda}_i = \dfrac{1}{N_i} diag \left(\sum\limits_{t:\ q_t=S_i} (\mathbf{x}_t - \hat{\mu}_i)(\mathbf{x}_t - \hat{\mu}_i)^T \right),\ 1 \leq i \leq N$
initial state distribution	$\hat{\pi}(1) = 1, \hat{\pi}(i) = 0,\ 2 \leq i \leq N$

Table 1: Baum-Welch Re-estimation Formulas

$$N_i = \sum_{t:\ q_t=S_i} 1. \tag{14}$$

To obtain each $\hat{a}_{ij}$, the probability of transitioning from S_i to S_j, we take the ratio of the number of transitions from S_i to S_j and the number of transitions from S_i to any other state. The mean vector per state is obtained by taking the average of all feature vectors in the state. The covariance matrix per state is formulated by taking the average of the estimated covariance matrices of the feature vectors in the same state. We assume that the initial state is always the first state, S_1, which explains the form of π.

After the estimation of the model, λ, we perform the second maximization over the state sequence given the model

$$\max_{\mathbf{q}} P(\mathbf{x}, \mathbf{q} \mid \lambda). \tag{15}$$

The above probability can be rewritten as

$$\begin{aligned} P(\mathbf{x}, \mathbf{q} \mid \lambda) &= P(\mathbf{x} \mid \mathbf{q}, \lambda) P(\mathbf{q} \mid \lambda) \\ &= \prod_{t=1}^{T} P(\mathbf{x}_t \mid q_t, \lambda) \pi(q_1) a_{q_1 q_2} a_{q_2 q_3} \cdots a_{q_{T-1} q_T} \\ &= \prod_{t=1}^{T} P(\mathbf{x}_t \mid q_t, \lambda) \prod_{t=2}^{T} P(q_t \mid q_{t-1}) \end{aligned} \tag{16}$$

The interpretation of the above is that at the initial time, $t = 1$, we begin in state q_1 with probability $\pi(q_1)$ and generate the observation vector $\mathbf{x}_1$ with probability $p(\mathbf{x}_1 \mid q_1, \lambda)$. At the next time instance, $t = 2$, we transition to state q_2 with probability $P(q_2 \mid q_1) = a_{q_1 q_2}$ and generate observation vector $\mathbf{x}_2$ with probability $p(\mathbf{x}_2 \mid q_2, \lambda)$. This procedure is repeated until the last observation vector. Because we assume that the initial state is always state one with probability one, we get rid of the $\pi(q_1)$ in the third equality.

The maximization of a product of probabilities in (16) is simplified by applying the logarithm, which yields

$$\max_{\mathbf{q}} \log P(\mathbf{x}, \mathbf{q} \mid \lambda) = \max_{\mathbf{q}} \sum_{t=1}^{T} \log P(\mathbf{x}_t \mid q_t, \lambda) + \sum_{t=2}^{T} \log P(q_t \mid q_{t-1}) \tag{17}$$

where the objective is to determine the state sequence, $\mathbf{q}$, that maximizes the above expression. Redrawing the HMM as a trellis, as shown in Figure 4, and considering the state sequence as a path through the trellis, it becomes fairly obvious that we can use the Viterbi algorithm to obtain the single best path that will maximize the $P(\mathbf{x}, \mathbf{q} \mid \lambda)$.

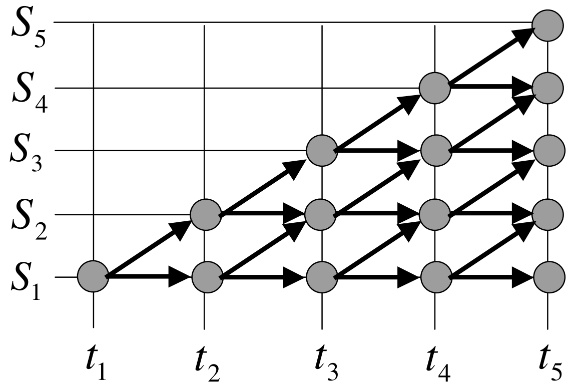

Figure 4: A trellis representing an HMM

The double maximization is repeated until convergence is achieved, where convergence is with respect to the best-path metric at the end of the trellis. Alternatively, we can stop the iteration as soon as the state sequence stops changing. The double maximization is performed for each word in the vocabulary until we obtain a model for each word. The training procedure is summarized in Figure 5.

3.4 Recognition

Now that we have a set of HMMs, one for each word, we can use them to recognize a speech signal. The objective is to determine the model that best characterizes the given speech signal. The decision rule for the recognition is based on the Maximum Likelihood (ML) detection rule, which says to choose the word model that maximizes the probability of the observation given the model. Thus, the classifier for observation $\mathbf{x}$ is

$$C(\mathbf{x}) = \arg\max_i P(\mathbf{x} \mid \lambda_i), \ 1 \leq i \leq M \tag{18}$$

where the observations are cepstral feature vectors extracted from the speech signal. The likelihood in the above maximization can be written as

$$P(\mathbf{x} \mid \lambda_i) = \sum_{\mathbf{q}} P(\mathbf{x}, \mathbf{q} \mid \lambda_i) \tag{19}$$

which is a summation over all state sequences, $\mathbf{q}$. Sometimes, it is desirable to replace the full likelihood of (19) by a Viterbi approximation in which only the most probable state sequence capable of producing $\mathbf{x}$ is taken into account. In this case, the summation in (19) is replaced by a max operator and the likelihood, $P(\mathbf{x} \mid \lambda_i)$, is approximated by

$$P(\mathbf{x} \mid \lambda_i) \approx \max_{\mathbf{q}} P(\mathbf{x}, \mathbf{q} \mid \lambda_i) \tag{20}$$

which can be calculated using the Viterbi algorithm [2] . This yields the best-path classifier

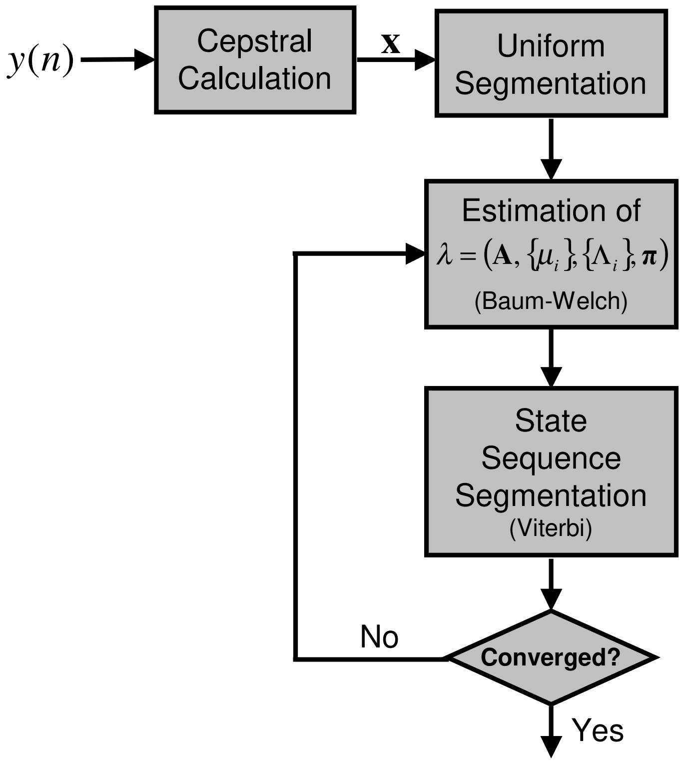

Figure 5: Block Diagram of the Training Procedure

$$C'(\mathbf{x}) = \arg\max_i \left[\max_{\mathbf{q}} \log P(\mathbf{x}, \mathbf{q} \mid \lambda_i) \right], \; 1 \leq i \leq M \tag{21}$$

that selects the recognized word as the one whose model has the highest best-path metric. A block diagram of the recognition system is illustrated in Figure 6.

4 Conclusion

In this summary, we introduced hidden Markov models and described one method of using them isolated-word speech recognition of a small vocabulary. This theory can also be extended to connected-word speech recognition, where the vocabulary size is also small. Once the appropriate word models have been derived, the objective becomes finding the optimum sequence of word models that best matches the connected-word string [3].

Although HMMs for speech recognition are popular, other techniques have been the attention of recent research on this topic. Artificial neural networks (ANN), as well as hybrid HMM/ANN approaches, have been proposed [2]. Some people also suggest that the maximum-likelihood approach may be mismatched to the desired objective of HMM design because the ultimate objective is not to accurately model the words, but to minimize the number of errors in the recognition of the words. Rao and Rose, [4], propose an HMM design aimed at minimizing classification error.

The area of speech recognition has been a target of extensive research the past few years. Although successful products, such as speech-to-text converters, already exist, the research is nowhere

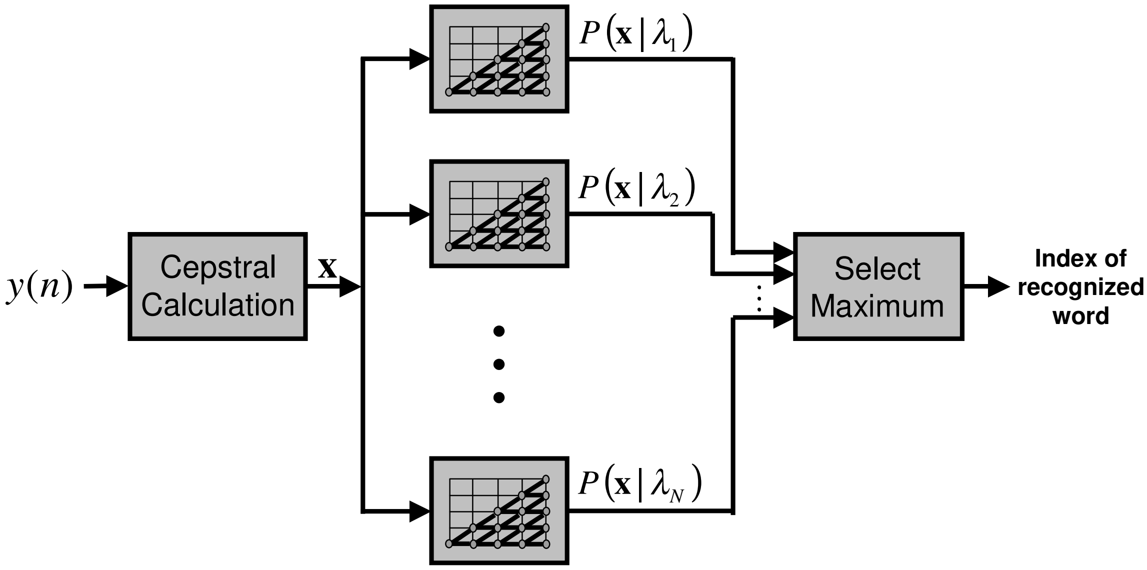

Figure 6: Block Diagram of an Isolated-Word Speech Recognition System

near completion. At the rate things are going, humans may be holding full conversations with their computers, and before long, the computers may just start talking to each other.

References

[1] Y. Ephraim and M. Rahim. "On Second Order Statistics and Linear Estimation of Cepstral Coefficients". *IEEE Trans. Speech and Audio Processing*, 7:162–176, March 1999.

[2] N. Morgan and H. Bourlard. "Neural Networks for Statistical Recognition of Continuous Speech". *Proceedings of the IEEE*, 83(5):742–770, May 1995.

[3] L. R. Rabiner. "A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition". *Proceedings of the IEEE*, 77(2):257–286, February 1989.

[4] A. V. Rao and K. Rose. "Deterministically Annealed Design of Hidden Markov Model Speech Recognizers". *IEEE Trans. Speech and Audio Processing*, 9:111–126, February 2001.

[5] D. B. Roe and J. G. Wilpon. "Whither Speech Recognition: The Next 25 Years". *IEEE Communications Magazine*, 31(11):54–62, November 1993.

[6] A. S. Willsky, G. W. Wornell, and J. H. Shapiro. Stochastic processes, detection and estimation course notes, 1999.