An Automatic Speech Recognition System for a Robot Dog

Mark Woodward

INTRODUCTION:

This paper describes an Automatic Speech Recognition System for a Robot Dog. The system involves the training of an acoustic model and then the decoding of audio files into word strings.

Subphones are used, 3 per phone in the language. The acoustic model uses a single Gaussian per subphone. Training of the acoustic model is done using the Viterbi algorithm with a flat start; see Acoustic Model Training. Decoding of audio files is done again by the Viterbi algorithm; see Decoding.

Results show that even this simple structure is sufficient for single speaker, ~10 word vocabularies; see Results.

ACOUSTIC MODEL TRAINING:

The acoustic model gives the probability that a specific subphone (3 per phoneme in the language) generated a specific 13 audio features, called Cepstral Coefficients. In this system a single gaussian is used to represent the distribution of the 13 audio features for each of the subphones. The mean of each subphone gaussian is the most likely value for each of the 13 audio features given that the specific subphone was voiced. Training is required to set the mean and variance of the gaussians for each of the subphones.

In this system a flat start is used, meaning that all of the gaussians (1 per subphone in the language) are initialized to the mean and variance of all the audio data in the training set. The training then enters a loop in which the Viterbi algorithm processes an audio file using the current values for the acoustic model, outputting an assignment of each time step to a specific subphone; see Decoding. The gaussians for each of the subphones are then set to the mean and variance of all Cepstral vectors for timesteps assigned to that subphone. This repeats until convergence. For the results below, convergence was achieved in about 20 iterations. Figure 1 shows the flow of the training process.

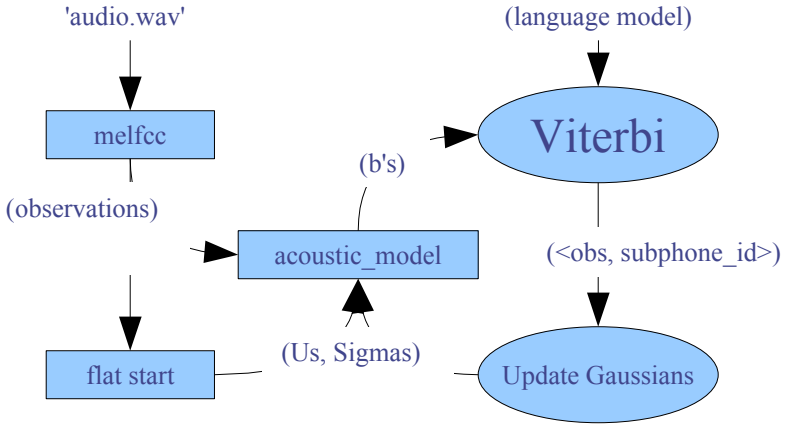

Figure 1: The work-flow for acoustic model training

DECODING:

Decoding is the process of producing a sequence of words given an audio file, an acoustic model, a language model, and a grammar. Decoding in this system is done using the Viterbi algorithm. The Viterbi algorithm is an algorithm for finding the approximate most likely sequence of states in an HMM given the probabilities of observations for each time step for each of the states. The probability of an observation given a state is computed using the acoustic model described above. The Cepstral coefficient vector corresponding to the time step is used to sample into the Gaussian of the subphone corresponding to that state. Note that a point estimate is used as an approximation to the probability. Figure 2 shows the flow of the decoding process.

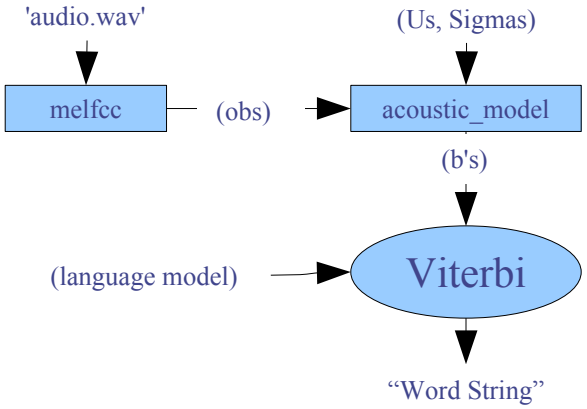

Figure 2: The work-flow for decoding an audio file into a word string.

RESULTS:

The grammar used to test out the system is shown in figure 3.

```
silence   -> (silence | sparky)
sparky    -> (around | come | fetch | free | good | heel
              | no | stay | twirl)
around    -> (silence | sparky)
come      -> (silence | sparky | stay)
fetch     -> (silence | sparky)
free      -> (silence | sparky)
good      -> (silence | sparky)
heel      -> (silence | sparky)
no        -> (silence | sparky)
stay      -> (silence | sparky)
twirl     -> (silence | sparky)
```

Figure 3: The grammar for testing out the system

Some typical sentences are shown in figure 4.

“Sparky come, stay, Sparky fetch”
“Sparky heel, Sparky good”
“Sparky twirl, Sparky around”
“Sparky no”
“Sparky stay, Sparky free”

Figure 4: Some typical sentences that can be generated by the grammar in figure 3. All of these are easily decoded by this system.

For a single speaker, for whom the acoustic model was trained, the system surpassed expectations. Given ambient conditions similar to that in training, no errors occurred in sentences containing 2 commands. e.g. "sparky sit stay, sparky fetch"; Demos available upon request. Unfortunately, due to time time constraints I was not able to test on a second speaker.

All components were implemented from scratch in Matlab in order to gain a better understanding of their inner workings. Obvious improvements include tri-phones instead of subphones, mixtures of gaussians instead of single guassians, and Baum-Welch instead of Viterbi training.

CONCLUSION:

This paper presented a system for training an acoustic model and decode an audio source into a word string. The system was tested in the context of commanding a robot dog. Sub-phones with Single Gaussians were trained using the Viterbi algorithm as a sub-routine. Decoding was also performed with the Viterbi algorithm. Results showed that even with a basic system such as this high accuracy can be obtained for single speakers given a relatively small vocabulary, ~10 words.