Speech Processing Assignment 2

B112767

1. Introduction

In this assignment, we are focusing on building our own digit recogniser using HTK toolkit. HTK is a portable software in order to solving automatic speech recognition(ASR) task based on Hidden Markov Model(HMM) method (Woodland et al., 1994). We are aiming to figure out how the HTK works by building up our own digit recogniser. At first, we built the speaker dependent system using our own labelled data and evaluate by Word Error Rate (WER). Secondly, it is required to design and conduct some experiments based on our own hypothesis while using lots of people's data. Finally, we could have a deep understanding about how an HMM-based ASR system runs from the initial step of data collecting to the model training and evaluation.

2. Theory

The purpose of ASR task is dealing with the speech recognition task, transforming speech signal to text strings (Jurafsky & Martin, 2014). It requires concise accuracy when doing recognition task. There are many effective methods to be used as our model, such as neural networks and HMM. In this coursework, we explored HMM-based speech recognition task in the following sections. Figure 1 shows the pipeline of HMM-based model speech recognition.

2.1. Data collection and acoustic features

First of all, the first requirement of supervised learning is valid labelled data. Therefore, we should collect effective speech data to prepare for the feature engineering. In the primary experiment of speaker-dependent system for isolated digit recognition, we have recorded our speech of digits and separate them into training set and test set. Firstly, I have recorded my speech of training set and test set using Praat and labelled training set using wavesurfer. Then, we used the labelled training data to extract the acoustic features by transforming our waveform into a acoustic feature vector frames. We use mel frequency cepstral coefficients(MFCC) to represent our acoustic features and make MFCC feature vectors from our training and test file by using HCopy command with configuration file(HCopy -C 'config_file_path'). In the first step of MFCC generation process, the energy magnitude with high frequency will be pre-emphasised, then separate spectral features into small windows and do Discrete Fourier Transform (Jurafsky & Martin, 2014). After that, 39 MFCC features are transformed by HCopy command and our own MFCC files are generated as training data for the following process.

2.2. Training HMMs

The next stage is initialising and training our HMM model using HTK tool. Before training HMM model in HTK, HMM model is initialised with a prototype model. We initialise the Gaussian mixture models in each emission state with 0 mean and variance 1. Each state has 39 parameters because of 39 MFCC features. We also initialise the transition probability

Figure 1. HMM-based speech recognition process.

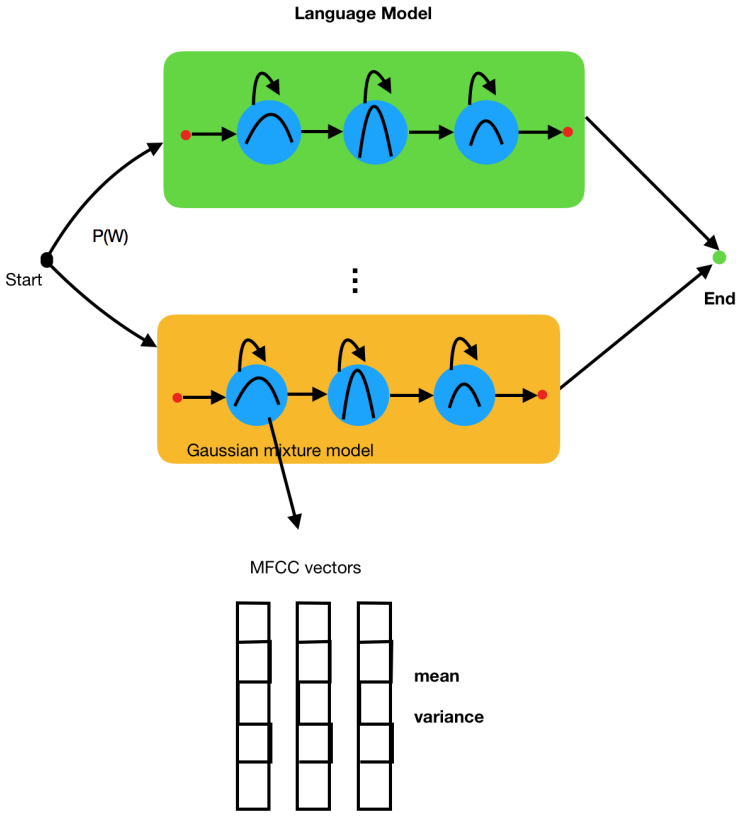

Figure 2. HMM model training process

between each states with a transition matrix in prototype model file. Table 1 shows the transition probability(from state i to j) for our initialisation. Then *HInit* command will initialise HMM models given the HMM prototype file and training data(MFCC file). Then *HInit* ran a uniform segmentation and Viterbi training for HMM model. As can be seen in figure 3, HMM model with 5 states has three emitting states, so we only need to initialise these three states with Gaussian models.

STATE	1(START)	2	3	4	5(END)
1(START)	0	1	0	0	0
2	0	0.5	0.5	0	0
3	0	0	0.5	0.5	0
4	0	0	0	0.5	0.5
5(END)	0	0	0	0	0

Table 1. Initialised transition probability between states.

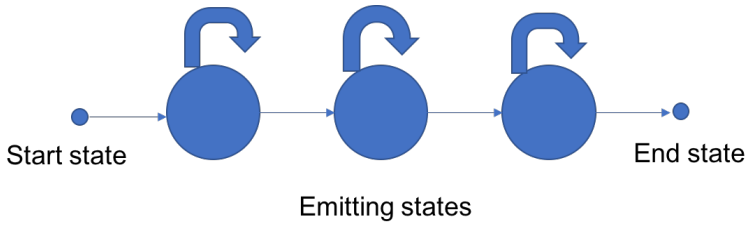

Figure 3. 5-state HMM model

After that, we began to train the HMM model. *HRest* can be used to do Baum-Welch re-estimation, which is a instance of EM algorithm by recursively compute the average mean and variance until model parameters do not change.

Figure 2 illustrates the training process(EM algorithm), which combines the language model and HMM model. After initialising the model with the mean and variance of the Gaussian mixture model in our HMM-based system, we could get a initialised model and compute the most probable output results with Viterbi algorithm. Each emission state could correspond to a frame of MFCC vectors and this MFCC sequences can be generated as our most probable recognition results.

Then we could re-compute the mean and variance based on the computed MFCC vector frames, re-compute the expectation of our HMM model and update the values of mean and variance again until converging (Bilmes et al., 1998). Finally, trained HMM can be combined with language model as the process in Figure 1.

2.3. Language modelling

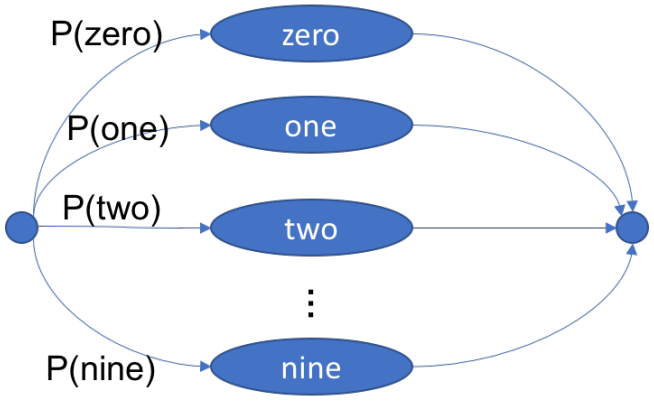

Figure 4. Isolated digit recognition language model.

In this stage we compute the probability of word sequence $P(W)$ as Figuire 4 (Singh, 2003). Maximum likelihood estimation(MLE) and be employed to create n-gram model, and smoothing methods such as add-α smoothing, Kneser-Ney smoothing can be used to deal with unseen cases. In this isolated digit recognition experiment, we only need to build a unigram language model to compute the probability of each words. If we do connected-digit recognition experiment, we also need to draw a line from the end state to the start state to make sure it can recurrently recognise the digit strings. As we can see in Figure 4, the finite state language model can be combined with HMM because the probability for each path can be computed easily and we could take the maximum as our output. We can use the probability of words $P(W)$ with the HMM model to compute the probability for each recognition string. Equation 1 computes the estimated recognition results, P(W) represents the language model of this stage, and P(O|W) means the likelihood of outputs implemented by HMM model in the previous step.

$$\hat{W} = \underset{W}{\operatorname{argmax}}\, P(O|W) \cdot P(W) \tag{1}$$

2.4. Recognition using HMMs

In this step, we use Viterbi algorithm to compute the best matching strings. We compute the maximum probability as Equation 2 (Jurafsky & Martin, 2014). $v_{i-1}(i)$ means the probability of the previous path, a_i means the transition probability(from state i to j), $b_j(o_t)$ means the emission probability of the observation o_t at state j. As Figure 5, every time Viterbi algorithm moves to next state, Viterbi algorithm compute all the probabilities of possible paths, and take the path with maximum probability. Finally, we could get the optimal output at the final state.

$$V_t(j) = max_{i=1}^{N} v_{t-1}(i) a_{ij} b_j(o_t) \tag{2}$$

HVite function implement the Viterbi recogniser, by reading dictionary file and labels and building a network of HMM model. It allows random sequence to be recognised correctly(Young et al., 2001). We use *HVite* to recognise our test MFCC files and store the recognised label files (as file with .rec extension).

Finally, *HResult* can be used to check the performance of our HMM-based acoustic model. It will print out the accuracy

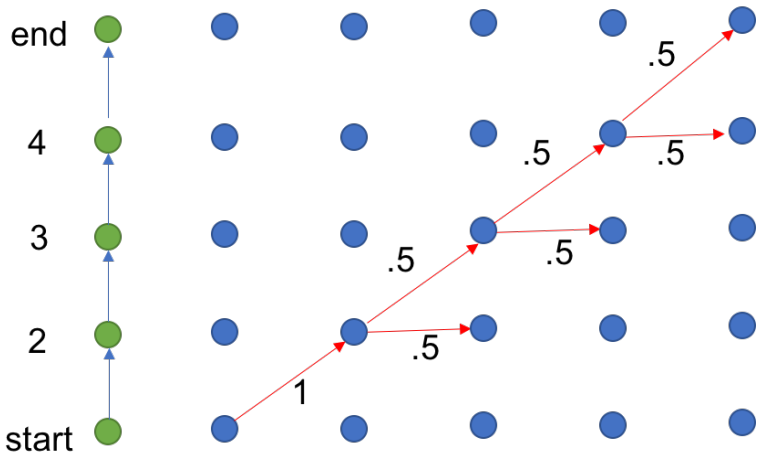

Figure 5. Viterbi algorithm

DATA SET	TRAINING SET1(M)	TRAINING SET2(F)
TEST SET1(M)	5.07%	27.03%
TEST SET2(F)	20.33%	7.67%

Table 2. The WER of test set from different genders in Experiment 3.1

for the performance of our HMM model. Also, the option $-p$ could be used to print out the phoneme statistics, which is a confusion matrix.

3. Experiments

In this section, I design three aspects of experiments based on data from multiple people in order to investigate the impact of different genders and accents on ASR task.

3.1. Impact of genders on ASR task

In the beginning, I intend to investigate whether training set and test set with different genders could increase the Word Error Rate(WER) in ASR task.

Hypothesis : HMM-based ASR system trained with single gender can not perform well when recognising test data from different gender. I hypothesised that HMM model trained and tested by data from the same gender could have much better performance when testing on data of different gender.

Experimental design :

- Training set1: training data of 30 UK male speakers with a mixture of microphone types.
- Training set2: training data of 30 UK female speakers with a mixture of microphone types.
- Test set 1: Test data of 10 UK male speakers **not** in training set 1, with a mixture of microphone types.
- Test set 2: Test data of 10 UK female speakers **not** in training set 2, with a mixture of microphone types.

As we can see in Table 2, the accuracy of HMM model trained by UK male data with mixture microphones can have 5.07% WER on test set1 from UK male, and perform around 15% lower WER than the performance on test set1 from UK female. Similarly, the the accuracy of HMM model trained by UK female data with mixture microphones performed around 20% more WER on test set1(from UK female) than test set2(from UK male).

Therefore, we can see that the model trained by data from different genders with mixture microphones performs varied when evaluating on data from different genders. This may be because male and female has the different pitch range and intonation when speaking: male have lower pitch and tend to speak with pitch from all ranges whilst female have higher pitch and speak with varied pitch from part of ranges (Jade, 2017). Abdulla et al. (2001) held the view that gender is the crucial factor when completing ASR tasks and they built up gender dependent systems after differentiating different genders according to the average pitch frequency. Also the amount of training sample is not large enough, it may lead to the fact that the data used to train HMM model do not cover the full range of female pitch. It may be worth furthering to explore them.

3.2. Impact of different accents on ASR task

Then I investigated how accent could affect the accuracy of HMM model using data from UK male and Scotland male from a mixture of microphones.

Hypothesis : Our trained model could have better performance on the test data from the same accent. Model trained on data from different accents performed variously when recognising test data from other accent.

Experimental design :

- Training set 1: training data of 30 UK male speakers with a mixture of microphone types.
- Training set 2: training data of 30 Scottish male speakers with a mixture of microphone types.
- Test set 1: Test data of 10 UK male speakers **not** in training set 1, with a mixture of microphone types.
- Test set2: Test data of 10 Scottish male speakers **not** in training data2, with a mixture of microphone types.

Table 3 shows the results of WER on test data of UK accent and Scottish accent. We can see that HMM model trained by UK male data with a mixture microphone outperformed model trained by Scottish male data when test on UK male data.

Nallasamy (2016) claimed that different accents can affect the pronunciation of words, which lead to different phonemes, phonotactic distributions and lexcical distributions. Besides, people who have the same accent(speak the same language) can have the same pattern of accented pronunciation(Nallasamy, 2016). Therefore, the accent could have affects on ASR task to some degree. Kamper & Niesler (2014) believed that there are two ways to solve this: building accent-independent ASR system or combining models of different accents and identifying the accent when doing ASR task. Therefore, it may be helpful to build systems for different accents after ASR systems recognize the speech accent.

3.3. Impact of training data amount on ASR task

In this experiment, I have explored the affect of training data amount on model performance, that is, whether the more data we use, the more accurate HMM models could be.

Data set	training1(M_UK)	training2(M_SC)
test1(uk_male)	5.07%	12.37%
test2(sc_male)	10.03%	3.04%

Table 3. The WER of test set from different accents in Experiment 3.2

Data set	test set(NN_female_27)
training set1(NN_female_80)	17.35%
training set2(NN_female_50)	24.41%
training set3(Mixed_female_80)	24.41%

Table 4. The WER of different training data amount in Experiment 3.3

Hypothesis : The more training data we use, the model could learning from more data and therefore have better generalisation ability. The model could be better if we use more data to train.

Experimental design :

- Training set1: training data of 80 male non-native(NN) speakers with a mixture of microphone types.
- Training set2: training data of 50 NN male speakers with a mixture of microphone types.
- Training set3: training data of 80 mixed(NN and native speaker) male speakers with a mixture of microphone types.
- Test set: Test data of 27 NN male speakers **not** in training set 1,2 and 3, with a mixture of microphone types.

Table 4 shows the accuracy of models trained on different amount of data. It can be seen that HMM trained by more NN data outperforms the other one trained by less NN data because model could learn more information with more data amount. However, model trained by mixed data did not perform as good as model trained by NN data. This may be because mixed data has more noise information that NN data do not have. After model learn the information from noise features, our model is fitted by some noise features, which could predict the wrong output. Therefore, selecting matched training data(without noise) is more important than just adding more data blindly with noise when collecting the training set.

4. Discussion and overall conclusion

In conclusion, HMM model can complete the ASR task by four steps. Firstly, data should be collected and then MFCC features should be extracted; then HMM model can be initialised and use Baum-Welch algorithm to learn its parameters; after that, language model was built for every possible output strings and Viterbi algorithm is employed to decode HMM and find the best output.

Moreover, there are some factors which may affect the performance of HMM-based ASR systems, such as accent and gender. We could select our training data carefully before we train our HMM model. When dealing with different factors which may impact the model performance, building factor-independent systems could be a reliable method to consider(Kamper & Niesler, 2014; Abdulla et al., 2001). Besides, we should also take care about the quality of our training data and avoid model fitted by noise data.

References

Abdulla, WH, Kasabov, NK, and Zealand, Dunedin-New. Improving speech recognition performance through gender separation. *changes*, 9:10, 2001.

Bilmes, Jeff A et al. A gentle tutorial of the em algorithm and its application to parameter estimation for gaussian mixture and hidden markov models. *International Computer Science Institute*, 4(510):126, 1998.

Jade. The difference between a male and female voice over, 2017. URL https://www.matinee.co.uk/blog/difference-male-female-voice/.

Jurafsky, Dan and Martin, James H. *Speech and language processing, chapter 9*, volume 3. Pearson London, 2014.

Kamper, Herman and Niesler, Thomas R. The impact of accent identification errors on speech recognition of south african english. *South African Journal of Science*, 110(1-2):1–6, 2014.

Nallasamy, Udhyakumar. *Adaptation techniques to improve ASR performance on accented speakers*. PhD thesis, Carnegie Mellon University, 2016.

Singh, Rita. Designing hmm-based asr systems, 2003. URL https://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-345-automatic-speech-recognition-spring-2003/lecture-notes/lecture15.pdf.

Woodland, C, Philip, Odell, J, Julian, Valtchev, Valtcho, Young, and J, Steve. Large vocabulary continuous speech recognition using htk. In *Acoustics, Speech, and Signal Processing, 1994. ICASSP-94., 1994 IEEE International Conference on*, volume 2, pp. II–125. Ieee, 1994.

Young, Steve, Evermann, Gunnar, Kershaw, Dan, Moore, Gareth, Odell, Julian, Ollason, Dave, Valtchev, Valtcho, and Woodland, Phil. The htk book, 2001. URL http://www1.icsi.berkeley.edu/Speech/docs/HTKBook/node279_mn.html.