Estimating speech from lip movement

Jithin D. George, Ronan Keane, Connor Zellmer

March 7, 2017

Abstract

1 Introduction and Overview

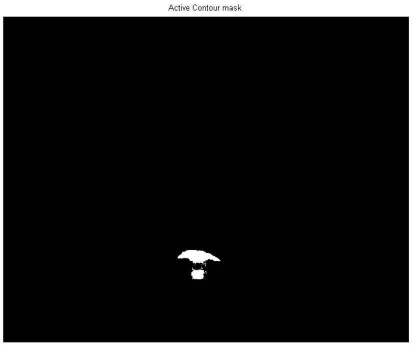

Figure 1: Active Contour

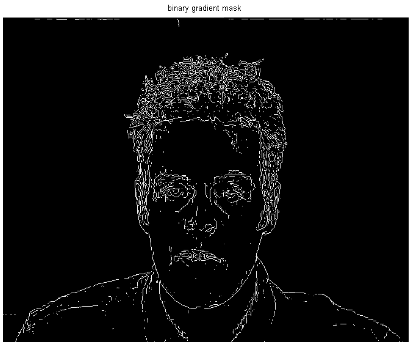

Figure 2: Binary Gradient

2 Theoretical Background

3 Hidden Markov Models

A Markov model involves the transition of a particular state to other states based on transition probabilities. A future state is only depends on the current state and not the states before it. Now, consider that at every state, there would be real world observations. These observations are controlled by the emission probabilities at each state.

For example, if we were to represent the ever changing weather, the states would be sunny, rainy or snowing and the observations would be summer clothes, rain boots or snow shoes. We can see that the emission probabilities for each observation is different depending on the state. To be more clear, the emission probabilities depend on the states.

We look at Hidden Markov Models(HMM). We decide that this is a relevant model because the words spoken are those defined by language and thus occur in specific pattern and not randomly. For example, given the first letter of word 'k', the probability that the next letter is a vowel is much higher than it being a consonant. A machine learning algorithm without this would be as inefficient as the initial Enigma machine in the movie "The Imitation Game". HMMs are very popular in the fields of speech [2] and gesture recognition [4] [5].

Although HMMs have fascinating problems related to evaluation and learning, our interests are in decoding. We have a sequence of observations and our aim is to estimate the states that created that. The Viterbi algorithm [3] gives us the states that maximize the occurrence of the observations.

So, given the features from the videos, we find the states. The states are the units of words, here chosen to be phonemes.

4 Implementation and Development

4.1 Extracting Phonemes

Using the nltk library in Python, we convert every word to its constitutive arpabet phonetics. It gives the following output for the words - f', 'see', 'sea', 'compute', 'comput', 'cat'. Only 'comput' fails because it isn't a real word

```
['EH1', 'F']
['S', 'IY1']
['S', 'IY1']
['K', 'AH0', 'M', 'P', 'Y', 'UW1', 'T']
'comput'
['K', 'AE1', 'T']
```

From the words spoken in our videos, we get a set of 36 unique phonemes. The code for this is shown in using B.1 The 36 unique phonemes are

```
'AA1', 'AE1', 'AH0', 'AH1', 'AO1', 'AW1', 'AY1', 'B', 'CH', 'D', 'DH', 'EH1', 'EY1', 'F',
 'G', 'IH0', 'IH1', 'IY1', 'JH', 'K', 'L', 'M', 'N', 'OW0', 'OW1', 'P', 'R', 'S', 'T',
  'TH', 'UW0', 'UW1', 'V', 'W', 'Y', 'Z
```

4.2 Extracting Transcripts

From the transcripts, we extract all the data into a csv file using B.2

5 Jobs

This section is where we see the stuff to do.

- lip tracking using color classification (Ronan)
- figures of edge tracking for the report
- Getting the transition probabilities between phonemes (Jithin)
- Getting a feature space for each phoneme.

  Code to extract all the output from the subtitles file. (Extract words and times)

  Code to guess how to split phonemes for each word

  Code to align and crop lips

  Code to split aligned lips to match phonemes
- Something with nnets.

6 Computational Results

7 Summary and Conclusions

Further step and modifications.

References

[1] J. Proctor, S. Brunton and J. N. Kutz, Dynamic mode decomposition with control, arXiv:1409.6358.

[2] Rabiner, Lawrence R. "A tutorial on hidden Markov models and selected applications in speech recognition." Proceedings of the IEEE 77.2 (1989): 257-286.

[3] Forney, G. David. "The viterbi algorithm." Proceedings of the IEEE 61.3 (1973): 268-278.

[4] Yang, Jie, and Yangsheng Xu. Hidden markov model for gesture recognition. No. CMU-RI-TR-94-10. CARNEGIE-MELLON UNIV PITTSBURGH PA ROBOTICS INST, 1994.

[5] Starner, Thad E. Visual Recognition of American Sign Language Using Hidden Markov Models. MASSACHUSETTS INST OF TECH CAMBRIDGE DEPT OF BRAIN AND COGNITIVE SCIENCES, 1995.

A MATLAB Code

A.1 Contours.m

```matlab
obj=VideoReader('vid1.mpg');
vidFrames = read(obj);
numFrames = get(obj,'numberOfFrames');
[mov]= getmovout(vidFrames,numFrames-1);
X=frame2im(mov(50));
A=rgb2gray(X);
mask = zeros(size(A));
mask(400:450,320:400) = 1;
bw = activecontour(A,mask,300);
figure, imshow(bw), title('Active Contour mask');
[~, threshold] = edge(A, 'sobel');
fudgeFactor = .5;
BWs = edge(A,'sobel', threshold * fudgeFactor);
figure, imshow(BWs), title('binary gradient mask');
```

B Python Code

B.1 Phonemes.py

```python
import nltk

arpabet = nltk.corpus.cmudict.dict()

for word in ('f', 'see', 'sea', 'compute', 'comput', 'cat'):
                try:
                        print(arpabet[word][0])
                except Exception as e:
                        print(e)
```

B.2 Transcripts.py

```python
import csv
import os
os.chdir("align")
beach = os.listdir()
with open("tes.csv", "w") as f:
        for sand in beach[:-1]:
                text_file = open(sand, "r")
                lines = text_file.read().split(',')
                k = lines[0]
                g = k.split()
                writer = csv.writer(f)
                writer.writerow(g)
```