1. Environment

MAC OS X Yosemite 10.10.5

2. Setup

a. Download packages: HTK-3.4.1.zip, HDecode-3.4.1.zip.

b. Unzip HTK-3.4.1.zip and place it into the main directory of this project.

```
$ unzip HTK-3.4.1.zip
$ mv htk/ ~/dsp/hw2/
```

c. Build HTK.

```
$ cd ~/dsp/hw2/htk
$ ./configure --without-x --disable-hslab
$ make all
$ make install
```

d. Unzip HDecode-3.4.1.zip and place it into htk/ directory.

```
$ unzip HDecode-3.4.1.zip
$ mv htk/HTKLVRec/ ~/dsp/hw2/htk/
```

e. Build HDecode.

```
$ cd ~/dsp/hw2/htk
$ make hdecode
$ make install-hdecode
```

f. Edit set_htk_path.sh.

```
$ cd ~/dsp/hw2
$ echo PATH=$PATH:"~/dsp/hw2/htk" > set_htk_path.sh
```

g. Write a new shell script all.sh to change the parameters for each training cases.

```
$ cd ~/dsp/hw2
$ vim test.sh
. . .
$ chmod 744 test.sh
```

h. Find the best result of the specified parameters.

```
$ ./test.sh min state max state min iteration max iteration
$ ./test.sh 5 20 3 10
```

3. File Edited

a. HTK Path: set_htk_path.sh

b. Nunmber of States : lib/proto

c. Iterations: 03_training.sh

d. Gaussian Mixtures: (not changed)

4. Original Parameters and Result

a. Number of States: 5

b. Iterations: 3

c. Accuracy: 74.34

5. Best Result

a. Number of States: 15

b. Iterations: 10

c. Accuracy: 96.14

6. Analysis

Accuracy		Iterations							
		3	4	5	6	7	8	9	10
States	5	74.34	75.78	75.83	76.29	76.52	76.93	76.99	76.93
	6	81.47	81.99	82.28	82.05	81.65	81.99	82.05	81.99
	7	87.80	88.20	88.32	88.49	88.49	88.43	87.97	88.15
	8	89.64	89.59	89.87	89.87	89.76	89.82	90.16	89.93
	9	90.91	91.20	91.43	91.48	91.43	91.54	91.20	91.25
	10	93.67	93.79	94.25	94.07	93.84	93.67	93.96	93.96
	11	94.25	94.36	94.59	94.71	94.59	94.65	94.53	94.48
	12	95.22	95.51	95.34	95.34	95.28	94.65	94.53	94.48
	13	95.57	95.80	95.63	95.63	95.51	95.68	95.97	96.03
	14	95.80	95.91	95.97	95.91	95.97	95.86	96.03	95.97
	15	95.91	95.80	95.91	95.80	96.03	96.09	96.03	96.14
	16	95.97	96.09	95.97	96.03	96.03	95.97	96.03	95.91
	17	95.57	95.63	95.51	95.57	95.63	95.68	95.74	95.80
	18	95.11	95.05	95.11	94.94	95.17	95.22	95.28	95.22
	19	95.05	95.05	94.94	94.88	94.94	94.88	94.88	94.88
	20	94.42	94.48	94.42	94.36	94.36	94.19	94.25	94.13

7. Conclusions

a. Iteration is not directly proportional to accuracy.

b. Number of states in not directly proportional to accuracy.

c. We can observe some local maximum (light yellow), thus 96.14 might not be the global maximum. I have tested 15 states and 15 iterations, it had 96.26 accuracy.

8. Reference

a. HTK bug: groups.google.com/forum/#!topic/fave-users/kPGS91wXUio

b. Shell Script: sites.google.com/site/tiger2000/

c. Shell Script: blog.twtnn.com/2013/12/shell-script.html

9. Testing Shell Scripts: test.sh

```
min_state=$1
max_state=$2
min_iter=$3
max_iter=$4

proto=lib/proto
proto_tmp=lib/proto_
result=result/accuracy
front=03_training_front.sh
end=03_training_end.sh

# clear result
echo -n > $result

# save partial code of 03_training.sh
head -n 17 03_training.sh > $front
tail -n 44 03_training.sh > $end

for (( i=$min_state ; i < $max_state+1 ; i=i+1 )); do

    file=$proto_tmp$i
    touch $file

    # print initial data
    echo '~o <VECSIZE> 39 <MFCC_Z_E_D_A>' >> $file
    echo '~h "proto"' >> $file
    echo '<BeginHMM>' >> $file
    echo '<NumStates>' $i >> $file

    # print states
    for (( j=2 ; j < $i ; j=j+1 )); do

        echo '<State>' $j >> $file
        echo '<Mean> 39' >> $file
        for k in {1..39}; do
            echo -n '0.0 ' >> $file
        done
        echo >> $file

        echo '<Variance> 39' >> $file
```

```
        for k in {1..39}; do
            echo -n '1.0 ' >> $file
        done
        echo >> $file

    done

    # print transition matrix
    echo '<TransP>' $i >> $file

    echo -n '0.0 1.0 ' >> $file
    for (( k=3 ; k < $i+1 ; k=k+1 )); do
        echo -n '0.0 ' >> $file
    done
    echo >> $file

    for (( j=2 ; j < $i ; j=j+1 )); do

        for (( k=1 ; k < $i+1 ; k=k+1 )); do

            if [ $k -eq $j ]; then
                echo -n '0.5 ' >> $file
            elif [ $k -eq `expr $j + 1` ]; then
                echo -n '0.5 ' >> $file
            else
                echo -n '0.0 ' >> $file
            fi

        done
        echo >> $file

    done

    for (( k=1 ; k < $i+1 ; k=k+1 )); do
        echo -n '0.0 ' >> $file
    done
    echo >> $file

    echo '<EndHMM>' >> $file

    cat $file > $proto
    rm $file

    # start testing with different iterations
    for (( j=$min_iter ; j < $max_iter+1 ; j=j+1 )); do

        k=`expr $j - 1`
        cat $front > 03_training.sh
        printf "for i in {0..%d}\n" $k >> 03_training.sh
        cat $end >> 03_training.sh

#       echo "Number of State:" $i >> $result
#       echo "Iterations:" $j >> $result

        ./00_clean_all.sh
        ./01_run_HCopy.sh
        ./02_run_HCompV.sh
        ./03_training.sh
        ./04_testing.sh

    done

done

rm -rf $front $end

# print result
cat $result
```