Programming Assignment

Artificial Intelligence

Dongsuk Yook
Korea University

Programming Assignment

- A continuous speech recognizer finds the most probable word sequence $\hat{w}_{1:n}$ for a given input speech $e_{1:t}$ (vector sequence) as follows.
  - $$\hat{w}_{1:n} = \arg\max_{w_{1:n}} P(w_{1:n}|e_{1:t})$$
    $$= \arg\max_{w_{1:n}} \frac{P(e_{1:t}|w_{1:n})P(w_{1:n})}{P(e_{1:t})}$$
    $$= \arg\max_{w_{1:n}} P(e_{1:t}|w_{1:n})P(w_{1:n})$$
    $$= \arg\max_{w_{1:n}} \sum_{q_{1:t}} P(e_{1:t}, q_{1:t}|w_{1:n})\, P(w_{1:n}) \qquad ; q_{1:t} \text{ state sequence}$$
    $$\approx \arg\max_{w_{1:n}} \max_{q_{1:t}} P(e_{1:t}, q_{1:t}|w_{1:n})\, P(w_{1:n})$$
  - $P(e_{1:t}, q_{1:t}|w_{1:n})$: acoustic model probability
  - $P(w_{1:n})$: language model probability (e.g., bigram)

- Implement the above continuous speech recognizer.
  - Input: microphone input or an MFCC file.
  - Output: the most probable word sequence for input speech.
  - Submit the source code and a confusion matrix of the recognition result.
  - You may use "HResults.exe" to generate the confusion matrix.
  - Due in 2 weeks.

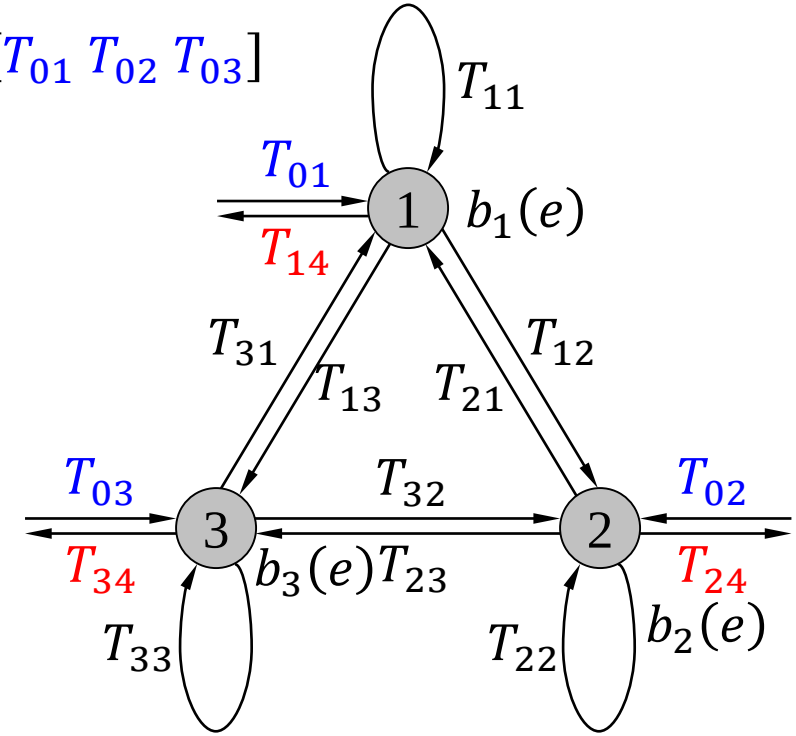
Input Vector File Format

- Input vector sequence file

```
313 39
-1.589671e+01  4.339182e+00  1.678270e+00 -4.386323e-02  1.384665e-01 ...
-1.573894e+01  2.713936e+00  2.918963e+00  1.807250e+00 -1.625646e+00 ...
-1.589687e+01  1.784740e+00 -3.876205e-03  1.939704e+00  1.013269e+00 ...
-1.686176e+01  3.179346e+00  6.970119e-01  7.169858e-01 -1.466554e+00 ...
-1.602454e+01  4.159081e+00  2.404717e+00  1.300133e+00 -1.309275e+00 ...
-1.794216e+01 -1.226994e-01 -1.229748e+00  2.328833e-02  3.530599e+00 ...
-1.572281e+01  3.731576e+00 -4.482310e-01 -1.252083e-01  2.847649e+00 ...
-1.571102e+01  6.004687e+00  1.940033e+00 -9.302789e-01  1.905544e+00 ...
-1.866060e+01 -1.945088e-01 -9.612672e-01 -6.845327e-01 -4.278716e+00 ...
-1.790727e+01 -3.463200e-01 -2.204390e-01 -6.221546e-01 -3.650035e+00 ...
-1.687654e+01  1.089474e+00 -2.015056e+00  7.445039e-01  2.003541e+00 ...
-1.630165e+01  9.615828e-01 -2.796509e+00  2.851351e-02 -2.366324e+00 ...
-1.762898e+01  3.966002e-01 -6.038963e-01  5.937940e-01  7.313928e-02 ...
-1.687426e+01  1.015894e+00 -1.440334e+00  8.511196e-01 -3.999560e+00 ...
-1.656823e+01  2.526161e+00 -1.373639e+00  2.825755e+00 -3.559372e-01 ...
-1.605652e+01  2.725700e+00  1.645913e+00  4.513128e+00  1.367162e+00 ...
-1.615862e+01  2.757725e+00 -1.037673e-01  5.169404e-01  2.256959e+00 ...
-1.697908e+01  2.430228e+00  1.174574e+00 -6.864926e-01 -2.884347e+00 ...
-1.562105e+01  4.122203e+00  6.119420e-01  2.408284e+00  1.406704e+00 ...
-1.586861e+01  2.400448e+00 -2.723778e+00 -3.281356e+00  1.186900e+00 ...
-2.964692e+01 -4.892936e+00  5.048756e+00 -7.816375e-01  9.942081e+00 ...
-3.060667e+01 -5.355003e+00  5.724719e+00  7.978249e-01  1.216068e+01 ...
-1.542544e+01  2.674652e+00  3.692956e-01 -1.053609e+00  3.725806e+00 ...
-1.660411e+01  5.190681e+00  3.267094e-01  2.324215e+00  2.873489e+00 ...
-1.603844e+01  3.882752e+00 -1.272774e-01  6.141130e+00  3.787947e+00 ...
-1.589794e+01  1.520315e+00 -6.553339e-01  2.869384e+00 -2.616245e-01 ...
...
```

Three State HMM

- HMM $M = (T, b)$
  - Transition probability
    - $P(X_0) = [P(s_1)\ P(s_2)\ P(s_3)] = [T_{01}\ T_{02}\ T_{03}]$
    - $T = \begin{bmatrix} T_{00} & T_{01} & T_{02} & T_{03} & T_{04} \\ T_{10} & T_{11} & T_{12} & T_{13} & T_{14} \\ T_{20} & T_{21} & T_{22} & T_{23} & T_{24} \\ T_{30} & T_{31} & T_{32} & T_{33} & T_{34} \\ T_{40} & T_{41} & T_{42} & T_{43} & T_{44} \end{bmatrix}$
  - Observation probability
    - $b = \begin{bmatrix} b_1(1) & b_1(2) & \cdots & b_1(v) \\ b_2(1) & b_2(2) & \cdots & b_2(v) \\ b_3(1) & b_3(2) & & b_3(v) \end{bmatrix}$



HMM File Format

- Single-Gaussian HMM

```
~h "ah"
<BEGINHMM>
<NUMSTATES> 5
<STATE> 2
<MEAN> 39
 1.898954e+000 -1.301708e+001 2.951807e-001 -8.873045e+000 -5.299952e+000 ...
<VARIANCE> 39
 1.374686e+001 2.792357e+001 3.375932e+001 3.855578e+001 5.125336e+001 ...
<GCONST> 1.185189e+002
<STATE> 3
 ...
<STATE> 4
 ...
<TRANSP> 5
 0.000000e+000 1.000000e+000 0.000000e+000 0.000000e+000 0.000000e+000
 0.000000e+000 6.985369e-001 3.014631e-001 0.000000e+000 0.000000e+000
 0.000000e+000 0.000000e+000 5.712691e-001 4.287309e-001 0.000000e+000
 0.000000e+000 0.000000e+000 0.000000e+000 5.327887e-001 4.672113e-001
 0.000000e+000 0.000000e+000 0.000000e+000 0.000000e+000 0.000000e+000
<ENDHMM>
~h "ao"
<BEGINHMM>
 ...
<ENDHMM>
...
```

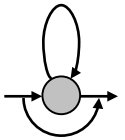
HMM File Format

- Two-Gaussian HMM

```
~h "ah"
<BEGINHMM>
<NUMSTATES> 5
<STATE> 2
<NUMMIXES> 2
<MIXTURE> 1 4.817315e-001
<MEAN> 39
 4.137055e+000 -1.180742e+001 1.235130e+000 -6.246143e+000 -5.400127e+000 ...
<VARIANCE> 39
 9.940362e+000 2.234269e+001 3.181495e+001 3.140755e+001 3.038879e+001 ...
<GCONST> 1.134534e+002
<MIXTURE> 2 5.182614e-001
<MEAN> 39
 7.230198e-002 -1.516407e+001 -2.030157e+000 -1.170948e+001 -3.230822e+000 ...
<VARIANCE> 39
 9.100752e+000 2.617574e+001 3.306291e+001 3.100306e+001 7.574311e+001 ...
<GCONST> 1.088633e+002
<STATE> 3
 ...
<STATE> 4
 ...
<TRANSP> 5
 ...
<ENDHMM>
...
```

HMM File Format

- Optional silence HMM



```
~h "sp"
<BEGINHMM>
<NUMSTATES> 3
<STATE> 2
<NUMMIXES> 2
<MIXTURE> 1 5.687151e-001
<MEAN> 39
 -1.528916e+001 1.884770e+000 -1.786322e-001 9.084788e-001 -2.541062e-001 ...
<VARIANCE> 39
 3.127717e+000 3.337751e+000 4.364497e+000 6.843961e+000 9.882758e+000 ...
<GCONST> 6.342905e+001
<MIXTURE> 2 4.312517e-001
<MEAN> 39
 -1.353393e+001 5.515828e-001 -1.442452e+000 3.601370e-001 -1.042004e+000 ...
<VARIANCE> 39
 9.201511e+000 1.160456e+001 1.037773e+001 9.865545e+000 1.413276e+001 ...
<GCONST> 8.848967e+001
<TRANSP> 3
 0.000000e+000 8.050888e-002 9.194912e-001
 0.000000e+000 9.276201e-001 7.237989e-002
 0.000000e+000 0.000000e+000 0.000000e+000
<ENDHMM>
```

HMM in Header File Format

- HMM in header file format for C programming

```
#define N_STATE          3
#define N_PDF            10
#define N_DIMENSION      39

typedef struct {
  float weight;
  float mean[N_DIMENSION];
  float var[N_DIMENSION];
} pdfType;

typedef struct {
  pdfType pdf[N_PDF];
} stateType;

typedef struct {
  char *name;
  float tp[N_STATE+2][N_STATE+2];
  stateType state[N_STATE];
} hmmType;
```

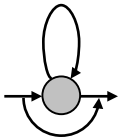
HMM in Header File Format

- HMM in header file format for C programming

```
hmmType phones[] = {
  { "f", // HMM
    { // transition probability
      { 0.000000e+000, 1.000000e+000, 0.000000e+000, 0.000000e+000, 0.000000e+000 },
      { 0.000000e+000, 8.519424e-001, 1.480576e-001, 0.000000e+000, 0.000000e+000 },
      { 0.000000e+000, 0.000000e+000, 7.039050e-001, 2.960950e-001, 0.000000e+000 },
      { 0.000000e+000, 0.000000e+000, 0.000000e+000, 5.744837e-001, 4.255163e-001 },
      { 0.000000e+000, 0.000000e+000, 0.000000e+000, 0.000000e+000, 0.000000e+000 }
    },
    {
      {{// state 1
        { // pdf 1
          8.379531e-002,
          { -1.100132e+001, -1.507629e+000, 5.286411e+000, 5.901514e+000, ... },
          { 2.583579e+001, 1.714888e+001, 1.768794e+001, 1.732637e+001, ... }
        },
        { // pdf 2
          ...
        },
        ...
      }},
      {{// state 2
        ...
      }}
      ...
    }
  },
  { "k", // HMM
    ...
  }
  ...
```

HMM in Header File Format

- HMM in header file format for C programming



```
  ...
  { "sp", // HMM
    { // transition probability
      { 0.000000e+000, 2.385641e-001, 7.614358e-001 },
      { 0.000000e+000, 9.152609e-001, 8.473914e-002 },
      { 0.000000e+000, 0.000000e+000, 0.000000e+000 }
    },
    {
      {{// state 1
        { // pdf 1
          1.120568e-001,
          { -1.508647e+001, 1.690120e+000, -3.829488e-001, 6.419236e-001, ... },
          { 3.735557e+000, 4.400073e+000, 6.065806e+000, 7.459801e+000, ... }
        },
        { // pdf 2
          ...
        },
        ...
      }}
    }
  },
};
```

Vocabulary

- Vocabulary

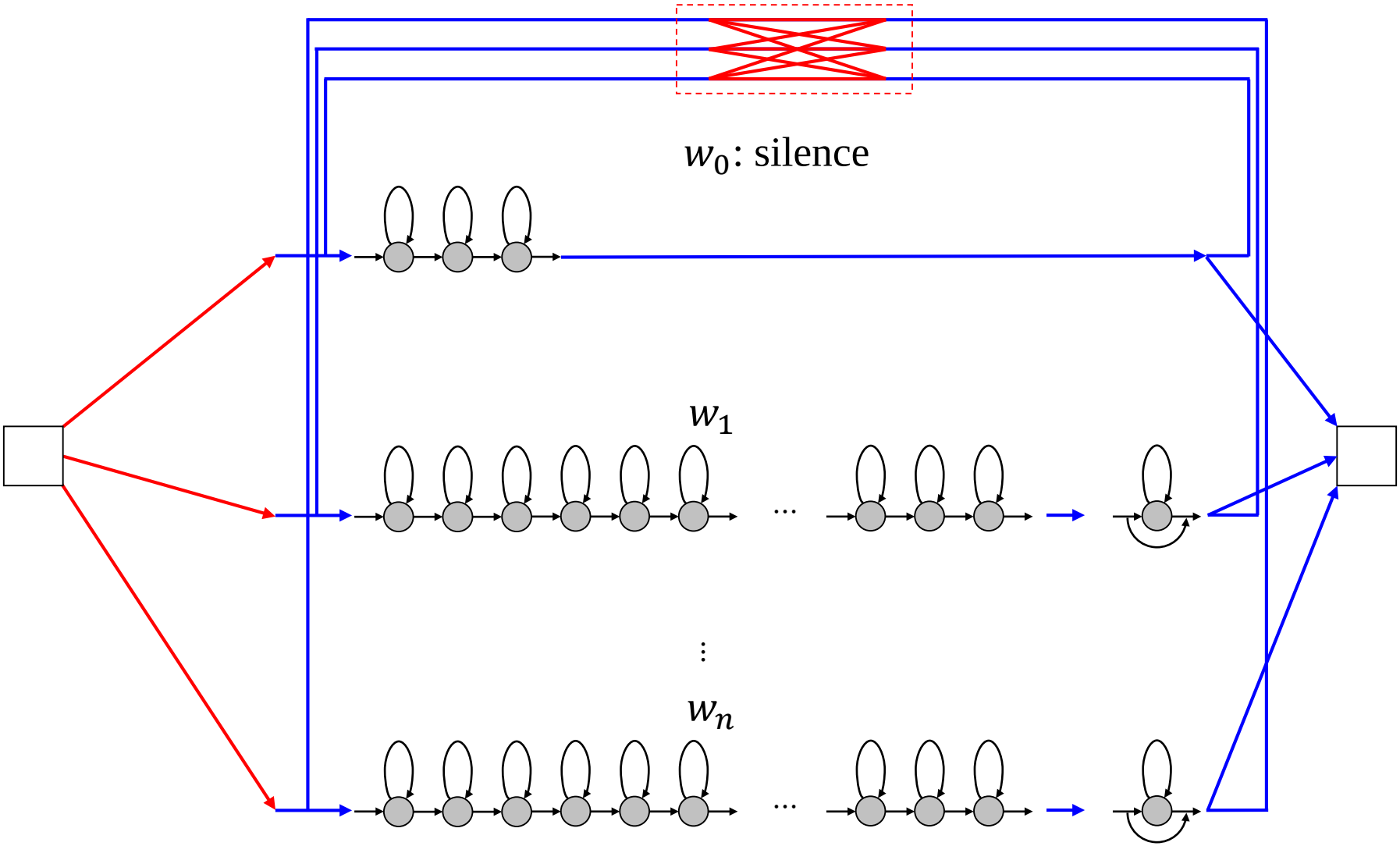
```
zero
oh
one
two
three
four
five
six
seven
eight
nine
```

Pronunciation Dictionary

- Pronunciation dictionary

```
<s>     sil
eight   ey t
five    f ay v
four    f ao r
nine    n ay n
oh      ow
one     w ah n
seven   s eh v ah n
six     s ih k s
three   th r iy
two     t uw
zero    z ih r ow
zero    z iy r ow
```

Utterance HMM Construction (Bigram)



Language Models

- Bigram

```
<s>     eight   0.012084
<s>     five    0.011881
<s>     four    0.009139
<s>     nine    0.011474
<s>     oh      0.012591
<s>     one     0.010967
<s>     seven   0.010967
<s>     six     0.011779
<s>     three   0.010865
<s>     two     0.013201
<s>     zero    0.010053
eight   <s>     0.012287
eight   eight   0.005991
eight   five    0.005788
eight   four    0.006600
eight   nine    0.007616
eight   oh      0.006397
eight   one     0.005585
eight   seven   0.005483
eight   six     0.005991
eight   three   0.005890
eight   two     0.006803
eight   zero    0.006499
five    <s>     0.013708
five    eight   0.005788
five    five    0.005686
...
zero    zero    0.013911
```

Language Models

- Unigram

```
<s>     0.990000
eight   0.000925
five    0.000890
four    0.000886
nine    0.000905
oh      0.000968
one     0.000905
seven   0.000869
six     0.000939
three   0.000883
two     0.000941
zero    0.000889
```

Label Format

- Label format (reference)

```
#!MLF!#
"tst/f/ak/1237743.lab"
one
two
three
seven
seven
four
three
.
"tst/f/ak/1393387.lab"
one
three
nine
three
three
eight
seven
.
"tst/f/ak/276317o.lab"
two
seven
six
three
one
seven
oh
.
```

Label Format

- Label format (recognized)

```
#!MLF!#
"tst/f/ak/1237743.rec"
one
two
three
seven
seven
four
three
.
"tst/f/ak/1393387.rec"
one
three
nine
three
three
eight
seven
.
"tst/f/ak/276317o.rec"
two
seven
six
three
one
seven
oh
.
```

Confusion Matrix

- Confusion matrix

```
HResults -p -I reference.txt vocabulary.txt recognized.txt

====================== HTK Results Analysis =======================
  Date: Mon Jan 1 00:00:00 2014
  Ref : reference
  Rec : recognized
------------------------ Overall Results --------------------------
SENT: %Correct=87.52 [H=1087, S=155, N=1242]
WORD: %Corr=99.82, Acc=97.98 [H=8678, D=4, S=12, I=160, N=8694]
------------------------ Confusion Matrix -------------------------
       z   o   o   t   t   f   f   s   s   e   n
       e   h   n   w   h   o   i   i   e   i   i
       r       e   o   r   u   v   x   v   g   n
       o               e   r   e       e   h   e
                       e               n   t       Del [ %c / %e]
zero  815  0   0   0   0   0   0   0   0   0   0    0
  oh    0  744  0   1   0   1   0   0   0   0   2    2 [99.5/0.0]
 one    0   0  809  0   0   0   0   0   0   0   1    0 [99.9/0.0]
 two    0   0   0  803  1   0   0   0   0   0   0    1 [99.9/0.0]
thre    0   0   0   2  812  0   0   0   0   0   0    0 [99.8/0.0]
four    0   0   0   0   0  783  1   0   0   0   0    0 [99.9/0.0]
five    0   0   0   0   0   0  784  0   0   0   0    0
 six    0   0   0   0   0   0   0  800  1   0   0    0 [99.9/0.0]
seve    0   0   1   0   0   0   0   0  791  0   0    0 [99.9/0.0]
eigh    0   1   0   0   0   0   0   0   0  824  0    1 [99.9/0.0]
nine    0   0   0   0   0   0   0   0   0   0  713   0
Ins     0  97   5   8   0   0   1   0   0  44   5
===================================================================
```

Language Model Weights

- Word transition penalty and language model weight
  - $\hat{w}_{1:n} \approx \arg\max_{w_{1:n}} \max_{q_{1:t}} \log P(e_{1:t}, q_{1:t}|w_{1:n})P(w_{1:n})$

    $\approx \arg\max_{w_{1:n}} \max_{q_{1:t}} [\log P(e_{1:t}, q_{1:t}|w_{1:n}) + \log P(w_{1:n}) - \lambda|w_{1:n}|]$

    $\approx \arg\max_{w_{1:n}} \max_{q_{1:t}} [\log P(e_{1:t}, q_{1:t}|w_{1:n}) + \lambda \log P(w_{1:n})]$

Summary

1. Read `hmm.txt` or include `hmm.h` in your source code.
2. Read `dictionary.txt`.
3. Construct word HMMs.
4. Read `unigram.txt` and `bigram.txt`.
5. Construct an universal utterance HMM.
6. Implement the Viterbi algorithm for the universal utterance HMM.
7. For each test file in `tst.zip`,
   ① Read a test file.
   ② Run the Viterbi algorithm and find the most likely state sequence.
   ③ Convert the state sequence to the corresponding word sequence.
   ④ Output the word sequence.
8. Run `HResult.exe` to produce a confusion matrix.
9. Repeat the step 7 with various values of language model weight and find the best recognition accuracy by roughly balancing deletion and insertion errors.