深蓝学院
shenlanxueyuan.com

《语音识别：从入门到精通》
第五章作业提示

助教 彭震东

作业一：Viterbi解码

深蓝学院 shenlanxueyuan.com

Lab2_vit.C

输入：
GMM参数、
状态转移参数、
音频数据、
单词标签

输出：
获得记录log概率和弧id的矩阵
chart

p018k7.22.20.gmm

```
% name: gmmMap
% type: matrix
% rows: 102
% columns: 1
 0
 1
 2
 3
 4
 5
 6
 7
 8
 9
```

GMM参数
GMM的ID，共102个

p018k7.22.20.gmm

```
% name: compMap
% type: matrix
% rows: 102
% columns: 1
 0
 1
 2
 3
 4
 5
 6
 7
 8
```

GMM参数
GMM的分量
单高斯一个分量，共102×1

p018k7.22.20.gmm

```
% name: compWeights
% type: matrix
% rows: 102
% columns: 1
 1
 1
 1
 1
 1
 1
```

GMM参数
由于是单分量，权重=1

p018k7.22.20.gmm

```
% name: gaussParams
% type: matrix
% rows: 102
% columns: 24
 0.446758 6.98444 0.398774 2.41593 1.71429 1.81304 0.
0.438089 -0.485581 0.663855 0.115927 0.373869 -0.5108
0.181438 -0.561005 0.885736
 0.781128 6.69152 0.728453 2.67212 2.16429 1.90951 0.
0.408175 -0.717483 0.68041 -0.0698345 0.420992 -0.676
0.236591 -0.646327 0.906887
 0.535762 5.9585 1.14746 2.36887 2.22871 1.85249 0.56
0.354747 -0.69348 0.71905 -0.120345 0.433987 -0.60472
0.295989 -0.51185 0.847266
 0.122714 4.97129 1.2665 1.99973 1.93471 1.71396 0.52
0.297161 -0.463362 0.699949 0.0166723 0.462589 -0.379
0.32588 -0.252602 0.673087
 -0.0562997 4.12557 1.10552 1.8111 1.53293 1.25533 0.
0.275735 -0.202224 0.515448 0.243153 0.420222 -0.1351
0.284157 -0.00691468 0.40911
 -0.0476727 3.49277 0.985717 1.51687 1.19257 0.723693
-0.185225 0.289705 -0.0312536 0.308871 0.403537 0.307
0.0625385 0.220004 0.130969 0.187635
```

GMM参数
gmm模型的μ和对角矩阵的值

p018k1.noloop.fsm

```
# states: 122
# input-voc: /dynamic/
# output-voc: /u/stanchen/lma/018/p018/../c018/c018m1.wdsp
1       9       0       EIGHT
1       6       6       FIVE
1       5       15      FOUR
1       10      24      NINE
1       11      33      OH
1       2       36      ONE
1       8       45      SEVEN
1       7       60      SIX
1       4       72      THREE
1       3       81      TWO
1       12      87      ZERO
1       13      99      ~SIL
2       14      36      <epsilon>
2       15      37      <epsilon>
3       16      81      <epsilon>
3       17      82      <epsilon>
4       18      72      <epsilon>
4       19      73      <epsilon>
5       20      15      <epsilon>
5       21      16      <epsilon>
6       22      6       <epsilon>
6       23      7       <epsilon>
7       24      60      <epsilon>
7       25      61      <epsilon>
8       26      45      <epsilon>
8       27      46      <epsilon>
9       28      0       <epsilon>
9       29      1       <epsilon>
10      30      24      <epsilon>
10      31      25      <epsilon>
11      32      33      <epsilon>
```

状态转移参数
初始状态、跳转状态、
对应GMM序号、对应的音素
state state gmm word

p018k7.1.dat

```
% name: utt2a
% type: matrix
% rows: 14080
% columns: 1
 3
 1
 3
 2
 2
 3
```

音频数据
一条语音14080个数字

p018k2.syms

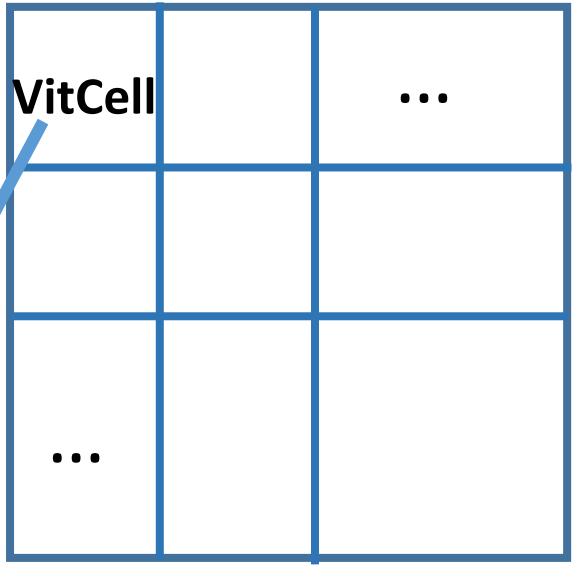
```
<epsilon> 0
ONE 1
TWO 2
THREE 3
FOUR 4
FIVE 5
SIX 6
SEVEN 7
EIGHT 8
NINE 9
```

单词标签

作业一： Viterbi解码

深蓝学院 shenlanxueyuan.com

```
On exit, chart(frmIdx, stateIdx).get_log_prob()
should be set to the logarithm of the probability
of the best path from the start state to
state "stateIdx" given the
first "frmIdx" frames of observations;
and chart(frmIdx, stateIdx).get_arc_id() should be set
to the arc ID for the last arc of this best path (or -1
if the best path is of length 0).
```



矩阵chart 对应上节课的 δ 矩阵 (帧数，状态)，
每个单元是一个**VitCell（log概率，弧id）**
无法到达的单元初始化为 g_zeroLogProb 和 -1

$$\delta_{t+1}(i) = \max_{i_1,i_2,\cdots,i_t} P(i_{t+1}=i, i_t,\cdots,i_1, o_{t+1},\cdots,o_1 \mid \lambda)$$
$$= \max_{1\leqslant j\leqslant N}[\delta_t(j)a_{ji}]b_i(o_{t+1}), \quad i=1,2,\cdots,N;\ t=1,2,\cdots,T-1 \qquad (10.45)$$

作业一： Viterbi解码

用到的一些方法：

获取该状态跳转的弧总数： *graph.get_arc_count(状态id)*
获取第一个弧的id：*graph.get_arc_count(状态id)*
由弧的id 来获取对应弧参数给arc：*graph.get_arc(arcId, arc)*

获取弧转移到的目的状态：*arc.get_dst_state()*
获取弧的转移log概率aij：*arc.get_log_prob()*
获取弧对应的gmmId：*arc.get_gmm()*
从gmmProbs矩阵中获取发射log概率：*gmmProbs(frameId, gmmId)*

赋值给chart单元：*chart(row,column).assign(log概率, 当前弧Id)*

作业二：　GMM参数估计

lab2_train.C

输入：
音频数据、
对齐序列、
GMM初始参数

输出：
GMM更新后参数

p018k7.22.dat

```
% name: rn:7a
% type: matrix
% rows: 20992
% columns: 1
 2
 1
 1
 1
 1
```

音频数据

一条语音20992个数字

p018k7.22.2.align

```
% name: rn:7a
% type: matrix
% rows: 103
% columns: 1
45
45
45
45
```

viterbi解码的最优状态序列

标记每个frame对应哪个gmm
这些帧对应第45个gmm

p018k1.gmm.dat

```
% name: gaussParams
% type: matrix
% rows: 102
% columns: 24
 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1
 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1
 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1
 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1
 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1
 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1
 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1
 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1
```

GMM初始参数

作业二： GMM参数估计

深蓝学院 shenlanxueyuan.com

GmmStats::add_gmm_count(gmmid , posterior , 帧特征)

```
"m_gaussCounts" is intended to hold the total occupancy count
of each Gaussian; "m_gaussStats1" is intended for
storing some sort of first-order statistic for each
dimension of each Gaussian; and "m_gaussStats2" is intended for
storing some sort of second-order statistic for each
dimension of each Gaussian... The statistics you take
need to be sufficient for doing the reestimation step below.
```

M步：

$$N_k = \sum_{n=1}^{N} \gamma(z_{nk})$$

$$\sum_{n=1}^{N} \gamma(z_{nk}) \mathbf{x}_n$$

更新到GmmStats 中的三个属性：

m_gaussCounts采用vector容器存储语料库中对应状态出现的总数

m_gaussStats1采用Matrix容器存储总体特征数据

m_gaussStats2也采用Matrix容器存储总体平方特征数据

$$\gamma(z_{nk})(\mathbf{x}_n - \boldsymbol{\mu}_k)(\mathbf{x}_n - \boldsymbol{\mu}_k)^{\mathrm{T}}$$

用于后续计算

作业二： GMM参数估计

深蓝学院 shenlanxueyuan.com

GmmStats::reestimate()

```
·m_gmmSet.set_gaussian_mean(gaussIdx,·dimIdx,·newMean);
·m_gmmSet.set_gaussian_var(gaussIdx,·dimIdx,·newVar);
```

用上一步得到的m_gaussCounts、m_gaussStats1、 m_gaussStats2 中存储的数值来计算**每一个高斯均值和方差并更新到m_gmmSet**

$$\boldsymbol{\mu}_k^{new} = \frac{1}{N_k}\sum_{n=1}^{N}\gamma(z_{nk})\mathbf{x}_n$$

$$\boldsymbol{\Sigma}_k^{new} = \frac{1}{N_k}\sum_{n=1}^{N}\gamma(z_{nk})(\mathbf{x}_n-\boldsymbol{\mu}_k^{new})(\mathbf{x}_n-\boldsymbol{\mu}_k^{new})^T$$

$$\pi_k^{new} = \frac{N_k}{N}, \qquad N_k = \sum_{n=1}^{N}\gamma(z_{nk})$$

作业三： 前向后向算法

深蓝学院 shenlanxueyuan.com

p018k7.1.dat

音频数据

一条语音14080个数字

```
% name: utt2a
% type: matrix
% rows: 14080
% columns: 1
 3
 1
 3
 2
 2
 3
```

p018k7.1.fsm

状态转移参数

state state gmm word

```
# name: utt2a
1       2       81      TWO
1       3       99      -SIL
2       4       81      <epsilon>
2       5       82      <epsilon>
3       6       99      <epsilon>
3       7       100     <epsilon>
4       4       81      <epsilon>
4       5       82      <epsilon>
5       5       82      <epsilon>
5       8       83      <epsilon>
6       6       99      <epsilon>
6       7       100     <epsilon>
7       7       100     <epsilon>
```

p018k7.22.2.gmm

GMM初始参数

```
% name: gaussParams
% type: matrix
% rows: 102
% columns: 24
 1.78483 9.07439 -1.37606 0.770062 1.2342 1.24608 1.1
0.778223
 1.86406 5.8347 0.324437 1.72267 2.92862 1.04291 0.40
 0.627198 3.72614 2.5177 0.380028 3.71711 0.377435 -0
0.370245
 0.842158 2.86277 2.78368 0.640238 2.36568 1.81553 1.
0.087329
 -0.614734 2.07299 2.63426 0.404085 1.47429 0.118752
-0.101324 0.0442475
 -0.313438 3.29428 0.795127 1.33528 1.04143 0.365529
```

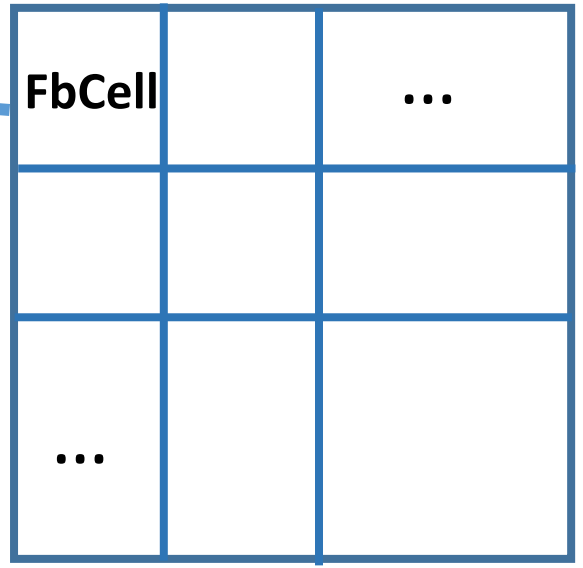
lab2_fb.C

输入： 音频数据、状态转移参数、GMM初始参数

输出： GMM更新后参数

作业三： 前向后向算法

矩阵chart 的每个单元FbCell用于存储
前向概率$\alpha_t(i)$ 和后向概率 $\beta_t(i)$



前向递推：

$$\alpha_{t+1}(i)=\left[\sum_{j=1}^{N}\alpha_t(j)a_{ji}\right]b_i(o_{t+1}),$$

后向递推：

$$\beta_t(i)=\sum_{j=1}^{N}a_{ij}b_j(o_{t+1})\beta_{t+1}(j)$$

弧上概率：

$$\xi_t(i,j)=P(i_t=q_i,i_{t+1}=q_j|O,\lambda)$$

$$=\frac{P(i_t=q_i,i_{t+1}=q_j,O|\lambda)}{P(O|\lambda)}$$

$$=\frac{\alpha_t(i)a_{ij}b_j(o_{t+1})\beta_{t+1}(j)}{\sum_{i=1}^{N}\alpha_T(i)}$$

作业三： 前向后向算法

深蓝学院 shenlanxueyuan.com

用到的其它方法：

存储前向概率：*chart(frmIdx, stateIdx).set_forw_log_prob(logProb)*
存储后向概率：*chart(frmIdx, stateIdx).set_back_log_prob(logProb)*
获取前向概率：*chart(frmIdx, stateIdx).get_forw_log_prob()*
获取后向概率：*chart(frmIdx, stateIdx).get_back_log_prob()*
将一组log概率转正常概率求和再取log：*add_log_probs(vector<double>)*
总前向概率和 $\sum_{i=1}^{N} \alpha_T(i)$ ： *uttLogProb = init_backward_pass(graph, chart)*

存储结果后验概率到 gmmCountList 中
gmmCountList.push_back(GmmCount(arc.get_gmm(), frmIdx - 1, exp(logProb - uttLogProb)))

深蓝学院
shenlanxueyuan.com

感谢各位聆听 !

Thanks for Listening