孤立词识别——GMM 模型

陈洁婷　2016202134

一、实验思路

在学习了 DTW 的经典识别方法后，我们了解到了 GMM 的方法。GMM 即高斯混合模型，其基本思想是通过对训练数据建立概率模型，再将测试数据代入每一类的模型中计算出它属于这个类别的概率，最后，比较可得出概率最大的类别，即得预测值。

本次实验，我的思路是利用 10 个不同人说的同一数字的音频对每个数字进行 GMM 建模，再将测试值代入每个数字的 GMM 模型中，选择出概率最大值对应的数字作为预测值返回。

详细思路如下

（1） 对每个数字的不同音频加窗分帧、提取 MFCC
（2） 以帧为单位投入 GMM 中进行训练调参，每个数字得到一个 GMM 模型
（3） 对于测试音频，也以帧为单位代入模型中，计算一帧属于某个数字的概率，根据概率事件的乘法原理，将每一帧算出的概率相乘，得到这段音频属于某个数字的概率
（4） 对 0-9 十个数字重复这样的操作，比较选择出测试音频得到最大概率时对应的数字，即可得到预测值。

二、GMM 模型的建立

下面描述本次实验中我建立 GMM 模型时的初始化、收敛判断方法。由于对每个数字的 GMM 模型，其 Gaussian 核数未知，其每个训练样本所属 Gaussian 未知（隐含变量），每个 Gaussian 的参数也未知，我们需要采用 BIC 算法和 EM 算法明确这些参数和隐含变量。

（1）BIC 算法：使用 Bayesian Information Criterion 算法确定最佳核数 K（Gaussian 的数目）

（2）EM 算法：确定隐含变量和每个 Gaussian 的参数。

a.参数初始化

样本属于某个隐含变量的概率 P（h）：采取平均的方法，使测试值属于每种类别的概率均为 1/K

均值μ：在确定了核数 K 后，使用 kmeans 的方法聚 K 类，取质心用来初始化各个均值

协方差矩阵Σ：先用全局的协方差作为替代。

b.Expectation step

利用参数的初始化值或上一次迭代的模型参数来算出隐变量的后验概率，即隐变量的期望，作为隐变量的现估计值

c.Maximization step

将似然函数最大化，更新参数值

d.重复 bc 步，反复迭代直至收敛。这里判断收敛的标准是其对于参数的更新不再有明显提高，停止迭代阈值采用默认值 tol=1e-3

三、实验过程

实验环境：python3.6、pycharm in macOS

所用库：wave、numpy、scipy、sklearn

（1） 音频数据读入

（2）加窗分帧、特征提取

对每个音频，我们通过加窗和分帧，能得到帧数不同，但每帧向量维度相同的 MFCC 特征数据。在 GMM 中，我们把每个数字的音频样本中分出来的帧看做独立，则所有帧的 MFCC 即可直接投入 GMM 中进行训练。而且，因为训练以帧为单位进行，也不需要考虑音频长度不同带来的问题。

注意，我们不能把帧向量拼接成音频特征向量，这样做是非常粗糙没有效果的，第一次实验时，我就犯下了这样的错误。而且这种试图对每个音频利用每帧 MFCC 拼接成一整个音频特征再做训练的方法，会导致只有 8 个训练向量（每个数字取八段音频做训练），这样少的数据，不可能训练出好的 GMM 模型。

```
def enframe(wave_data, nw, inc, winfunc):   #分帧并加窗
    '''将音频信号转化为帧。
    参数含义:
    wave_data:原始音频型号
    nw:每一帧的长度(这里指采样点的长度，即采样频率乘以时间间隔)
    inc:相邻帧的间隔（同上定义）
    '''
    wlen=len(wave_data) #信号总长度
    print('信号长',wlen,'帧长',nw,'移动步长',inc)
    if wlen<=nw: #若信号长度小于一个帧的长度，则帧数定义为1
        nf=1
    else: #否则，计算帧的总长度
        nf=int(np.ceil((1.0*wlen-nw+inc)/inc))
        print('帧数',nf)
    pad_length=int((nf-1)*inc+nw) #所有帧加起来总的铺平后的长度
    zeros=np.zeros((pad_length-wlen,)) #不够的长度使用0填补，类似于FFT中的扩充数组操作
    pad_signal=np.concatenate((wave_data,zeros)) #填补后的信号记为pad_signal
    indices=np.tile(np.arange(0,nw),(nf,1))+np.tile(np.arange(0,nf*inc,inc),(nw,1)).T
    indices=np.array(indices,dtype=np.int32) #将indices转化为矩阵
    frames=pad_signal[indices] #得到帧信号
    #print(frames)
    win=np.tile(winfunc,(nf,1))  #window窗函数，这里默认取1
    return frames*win   #返回帧信号矩阵
```

（3）对每个数字，直接调用 sklearn 库中的 mixture.GaussianMixture 库训练模型并保存，这里后期还需要调整主要参数，寻找最优值。在这里，我使用了 bic 算法，计算不同 K 值下的惩罚，选择惩罚最小的 K 值，从而确定每个数字最优的高斯核数。这里说到的 K 个核，可以形象地理解成，每一个核代表着数字单词中的一种音素，音素就是每个数字 GMM 模型的隐含变量。

```
        T=np.vstack((T,template))
lowest_bic = np.infty
bic = []
#先随意初始化一个gmm
gmm = GaussianMixture(n_components=1, covariance_type="diag")  # ASR中一般采用diag对角线的协方差矩阵
n_components_range = range(1, 9)  #遍历不同个数的gaussian
kernel=1  #用于记录每个数字的GMM模型的核数，这里先随意初始化一个值
for n_components in n_components_range:
    avg=float(1/n_components)
    w=[avg]*n_components
    print(w)
    tempgmm = GaussianMixture(n_components=n_components, covariance_type="diag",weights_init=w)
    tempgmm.fit(T)
    bic.append(tempgmm.bic(T))
    if bic[-1] < lowest_bic:
        lowest_bic = bic[-1]
        gmm=tempgmm
        kernel=n_components
print("the number of number",i,"'s Gaussian is:",kernel)
save_path="model"+str(i)+".m"
joblib.dump(gmm,save_path)
```

（4）对每个测试数据，分别调出十个 GMM 模型，计算出概率的 log 值的相反数，挑出概率最大时对应的数字，作为预测值返回。经过以帧为单位的训练，我们得到的 GMM 模型是一个输入为测试帧，输出为测试帧属于该 GMM 模型对应数字的概率。而对于一整个测试音频，它的每一帧属于某一数字的概率之积，就是这个音频属于某个数字的概率。

注意，这里如果直接使用概率值，可能会出现由于概率值太小，贴近于 0 而出 bug 的问题，因此我们可以直接对概率取 log 再取相反数进行比较，为了防止

溢出，可以再除以 10。这些操作对最后的乘积影响不大。

$$w^* = \arg\min_{w \in \text{vocab}} -\log P(A'_{\text{test}}|w)$$

```
        test = np.array(test, dtype=np.float)
        for i in range(10):
            Pt=1.0  #the probability of a wave file
            model_path = "model" +str(i) + ".m"
            gmm=joblib.load(model_path)
            for frameMFCC in test:
                P = gmm.score_samples([frameMFCC])[0]
                P=-1*(1*P)/10   #寻找-log(P)的最小值,除以10是为了方便计算,防止数据溢出
                Pt=Pt*P  #由事件概率可知整个音频属于某个数字的概率由其每帧属于某个数字的概率相乘得到
            # print("P is here:",P)
            if Pt<=Pmin:
                result=i
                Pmin=Pt
        print('the number is:', t)
        print('the predict number is:', result)
        #计算命中率
        if result == t:
            hit = hit + 1
hitrate = hit / test_size
print('the hit rate is:', hitrate)
```

（5）测试、修正与调整

1)关于 covariance_type 协方差阵类型

上课时我们了解到，在 ASR 中为了避免参数过多开销过大，我们一般采用对角协方差阵，因此，我们选择 covariance_type=diag。对于该参数，我也选用过 full 的尝试，比如，采用 diag 的命中率为 65%，而采用 full 的命中率也为 65%，多次试验可以发现开销过大且效果提高不大，遂放弃，确定选用对角协方差阵 diag

2)关于特征向量的处理

最初，我考虑将每个音频作为一条数据投入 GMM 中训练，于是我希望通过每帧的 MFCC 特征得到一段音频的特征。而由于音频长度不一定，故我采用了上面提到的补 0 和截取的方法统一长度。但显然，这样的处理方式会对特征向量造成影响。于是，我尝试采用 pca 降维的方法统一向量特征，试用了调库、手写函数等多种方法，但发现 pca 都只能用于样本数>=样本特征维度的情况，本实验中，每个数字的训练样本只有 8 个，特征却有 2000 多个，显然不适用。关于 pca 降维以及帧特征合成为音频特征的问题，后续我将向老师或助教请教。

请教之后，我发现这样的思路是错误的，我应当将每帧的 MFCC 直接作为独立的点投入 GMM 中训练。将每帧的 MFCC 拼接起来的想法是不正确的，这样确实没法得到有代表性的特征，现实中不会有人这么做。

4)关于每个高斯的权重初始化

我采取了平均的方法，使测试值属于每种类别的概率均为 1/K，在 gmm 中通过设置参数 weight_init 确定。但这种方法和默认的 KMeans 初始化方法相比似乎差距并不大。

```
avg=float(1/n_components)
w=[avg]*n_components
print(w)
tempgmm = GaussianMixture(n_components=n_components, covariance_type="diag",weights_init=w)
```

5）关于每个数字 gmm 模型的高斯个数选择

为选择出每个数字 gmm 模型的最佳高斯个数，我在其中加入了 BIC 算法。BIC 算法对不同的高斯个数选择所带来的惩罚进行了比较，选择了惩罚最小的作为该数字的 GMM 模型选用的高斯核数。对于每个数字，我都对 1-50 个核进行了遍历，发现 bic 算法显示的最优核数在 18 左右。然而，bic 算法的最优似乎并不代表着算法最终的命中率最优。经过对数据集大小、对计算开销、bic 最优结果、算法命中率的权衡，我将高斯核数目限制在了 15 以下。经试验发现，每个数字最优的高斯核数可能是不同的，但差异不会非常大。

```
#先随意初始化一个gmm
gmm = GaussianMixture(n_components=1, covariance_type="diag")  # ASR中一般采用diag对角线的协方差矩阵
n_components_range = range(1, 15)  #遍历不同个数的gaussian
kernel=1  #用于记录每个数字的GMM模型的最优核数，这里先随意初始化一个值
for n_components in n_components_range:
    avg=float(1/n_components)
    w=[avg]*n_components
    tempgmm = GaussianMixture(n_components=n_components, covariance_type="diag",weights_init=w)
    tempgmm.fit(T)
    bic.append(tempgmm.bic(T))
    if bic[-1] < lowest_bic:
        lowest_bic = bic[-1]
        gmm=tempgmm
        kernel=n_components
print("the number of number",i,"'s Gaussian is:",kernel)
save_path="model"+str(i)+".m"
joblib.dump(gmm,save_path)
```

bic 算法验证核数在 1-14 时

每个数字的核数选择

```
Is training now……
the number of number 0 's Gaussian is: 14
the number of number 1 's Gaussian is: 11
the number of number 2 's Gaussian is: 14
the number of number 3 's Gaussian is: 13
the number of number 4 's Gaussian is: 12
the number of number 5 's Gaussian is: 13
the number of number 6 's Gaussian is: 13
the number of number 7 's Gaussian is: 14
the number of number 8 's Gaussian is: 13
the number of number 9 's Gaussian is: 14
```

命中率

```
the number is: 9
the predict number is: 9
the number is: 9
the predict number is: 9
the hit rate is: 0.7
```

bic 算法验证核数在 1-50 时

每个数字的核数选择

```
Is training now……
the number of number 0 's Gaussian is: 17
the number of number 1 's Gaussian is: 19
the number of number 2 's Gaussian is: 13
the number of number 3 's Gaussian is: 18
the number of number 4 's Gaussian is: 20
the number of number 5 's Gaussian is: 15
the number of number 6 's Gaussian is: 18
the number of number 7 's Gaussian is: 20
the number of number 8 's Gaussian is: 18
the number of number 9 's Gaussian is: 18
```

命中率

```
the number is: 9
the predict number is: 2
the number is: 9
the predict number is: 9
the hit rate is: 0.55
```

六、结果分析

取前两位 speaker 说的数字 0-9 作为测试数据，其余作为训练数据时，命中率在 65%-80%之间浮动

test1 命中率 0.75

```
the number is: 0
the predict number is: 0
the number is: 0
the predict number is: 0
the number is: 1
the predict number is: 1
the number is: 1
the predict number is: 1
the number is: 2
the predict number is: 2
the number is: 2
the predict number is: 6
the number is: 3
the predict number is: 3
the number is: 3
the predict number is: 8
the number is: 4
the predict number is: 4
the number is: 4
the predict number is: 4
the number is: 5
the predict number is: 5
the number is: 5
the predict number is: 0
the number is: 6
the predict number is: 7
the number is: 6
the predict number is: 6
the number is: 7
the predict number is: 7
the number is: 7
the predict number is: 7
the number is: 8
the predict number is: 8
the number is: 8
the predict number is: 3
the number is: 9
the predict number is: 9
the number is: 9
the predict number is: 9
the hit rate is: 0.75
```

test2　命中率 0.8

```
the number is: 0
the predict number is: 0
the number is: 0
the predict number is: 0
the number is: 1
the predict number is: 1
the number is: 1
the predict number is: 1
the number is: 2
the predict number is: 2
the number is: 2
the predict number is: 2
the number is: 3
the predict number is: 3
the number is: 3
the predict number is: 2
the number is: 4
the predict number is: 4
the number is: 4
the predict number is: 4
the number is: 5
the predict number is: 5
the number is: 5
the predict number is: 7
the number is: 6
the predict number is: 3
the number is: 6
the predict number is: 6
the number is: 7
the predict number is: 7
the number is: 7
the predict number is: 7
the number is: 8
the predict number is: 8
the number is: 8
the predict number is: 3
the number is: 9
the predict number is: 9
the number is: 9
the predict number is: 9
the hit rate is: 0.8
```

取前 8 组人说的数字为训练数据，后 2 组人说的数字 0-9 为测试数据时，命中率为 45%-65%

test1　命中率 0.5

```
the number is: 0
the predict number is: 3
the number is: 0
the predict number is: 0
the number is: 1
the predict number is: 3
the number is: 1
the predict number is: 1
the number is: 2
the predict number is: 2
the number is: 2
the predict number is: 8
the number is: 3
the predict number is: 3
the number is: 3
the predict number is: 3
the number is: 4
the predict number is: 4
the number is: 4
the predict number is: 4
the number is: 5
the predict number is: 7
the number is: 5
the predict number is: 3
the number is: 6
the predict number is: 8
the number is: 6
the predict number is: 8
the number is: 7
the predict number is: 3
the number is: 7
the predict number is: 7
the number is: 8
the predict number is: 8
the number is: 8
the predict number is: 8
the number is: 9
the predict number is: 3
the number is: 9
the predict number is: 3
the hit rate is: 0.5
```

Test2　命中率 0.6

```
the number is: 0
the predict number is: 0
the number is: 0
the predict number is: 0
the number is: 1
the predict number is: 7
the number is: 1
the predict number is: 1
the number is: 2
the predict number is: 2
the number is: 2
the predict number is: 8
the number is: 3
the predict number is: 3
the number is: 3
the predict number is: 8
the number is: 4
the predict number is: 4
the number is: 4
the predict number is: 4
the number is: 5
the predict number is: 7
the number is: 5
the predict number is: 6
the number is: 6
the predict number is: 8
the number is: 6
the predict number is: 8
the number is: 7
the predict number is: 7
the number is: 7
the predict number is: 7
the number is: 8
the predict number is: 8
the number is: 8
the predict number is: 8
the number is: 9
the predict number is: 7
the number is: 9
the predict number is: 9
the hit rate is: 0.6
```

经参数调整和多次测试，本次实验的识别准确率最终大致落在 50%-80%之间，GMM 方法相比于最简单的 DTW 方法约 60%的命中率，效果提升明显。

在做这次实验之前，我对 GMM 模型一直理解模糊，许多细节的由来并不清楚。并且，对于一段音频的特征处理，也十分迷茫，查了很多资料，都找不到清晰明白的解释。一开始，我按照自己的理解，对每个训练音频 flatten 拼接出一个

MFCC 大特征，并将其作为这个音频的特征向量用于训练和测试，准确率在 45%左右。经过和老师、同学的交流我才知道，这样拼接 MFCC 获得的特征是没有意义的，非常粗糙。同时，这样会导致数据过少，根本无法建立出合理的 GMM 模型。

此外，把每段音频的一个特征向量投入 gmm 中，也无法充分发挥 gmm 的精妙之处。抽象地理解，那样最多只能以同一数字的不同 speaker 音频为 components 的 GMM，而无法反映一个数字发音的音素构成特点。因此，我们应当以帧为单位，进行 GMM 模型的训练。至于测试，我也以帧为单位进行，获得每个帧属于某个数字的概率，再由概率事件的乘法原理，可知它们的乘积就是整个音频属于某个数字的概率。最后，对每个数字的 GMM 上得到的概率作比较，即可挑出预测值。

经过这次实验，将一段音频的我对 GMM 模型有了更深入的认识，对它的每一个参数的功能都有了清楚的理解。但需要注意的是，目前我们仅仅将每一帧视为独立，并未考虑其前后的序列特征，但很多论文资料表明，人们更多地是将 GMM 与 HMM 搭配，以进行孤立词识别的。事实上，每个词不仅仅是音素上有差别，序列特征也是很重要的一个方面。因此，在后面的实验中，我将尝试搭配 HMM 模型，实现更好的孤立词识别系统。