音乐文件的分类

一、引入

随着时代的变化，音乐产品在不断的发生改变，如图1-1



图 1-1

从留声机->录音机->CD机->MP3->手机APP。而人的需求也在不断地发生改变，现在地人们不仅要能听音乐，还要能在不同的时间、场合，去听符合自己情绪的或者能让自己满意的音乐。人工智能里面地推荐系统就能够满足人们这一需要。

二、例子

现在我们以网易云音乐来举例，

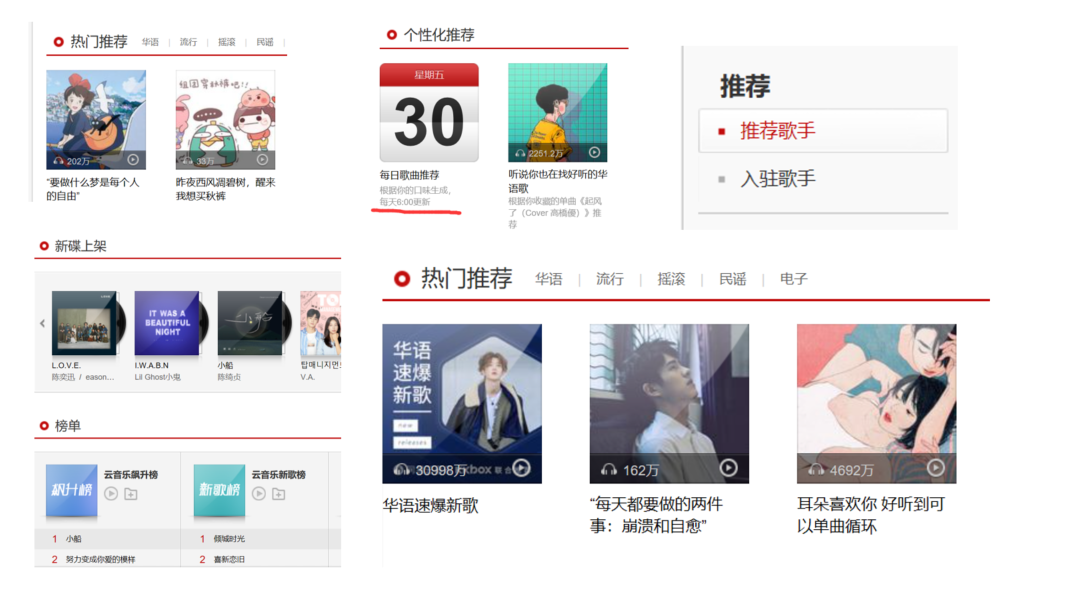


图2-1

图2-1地内容是在网易云音乐上截取过来的一部分，我们可以看到一系列的推荐，其实，还有很多，比如：朋友圈的推荐、评论的推荐等。如果从技术的角度来看待，网易云的算法也就是协同过滤算法与基于内容的推荐之间的一个配比，也就是我们个性化推荐的惊喜感与平衡性的一个配比。而个性化的推荐效果好坏，是基于我们模型的好坏，而模型的好坏，是基于我们的特征提取，所以特征的提取是非常重要的。比如：用机器学习来创建一个推荐系统。如图2-2所示：

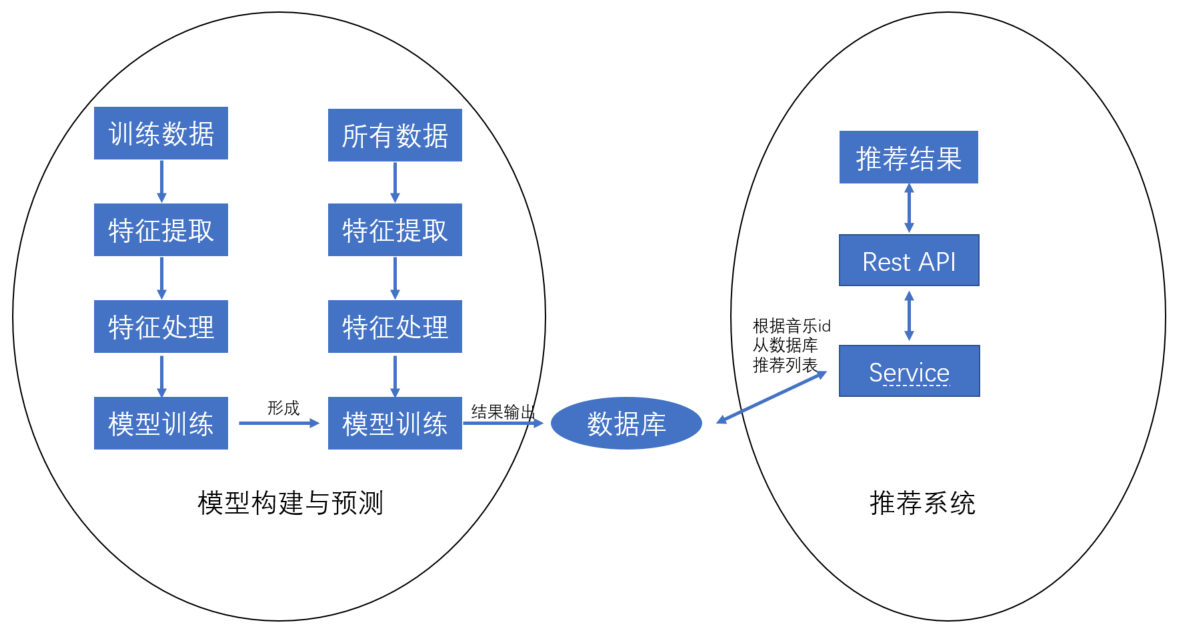


图2-2

在图2-2里面，可以很明显的看出来，

要想推荐结果的准确率高，符合用户需要，你是必须要保证数据库里面的文件分类是准确的，而数据库里面文件的分类准确性是由我们训练的模型预测得来的，而模型预测的准确性除了与我们所使用的算法有关系之外，还与我们的特征提取有很大关系。那么对于音乐文件来说，它是如何进行分类的呢？

三、音乐文件如何进行分类

网易云音乐里面对于音乐文件进行了分类,如图3-1



图3-1

在图3-1我们可以看到，音乐文件首先被分为语种、风格、场景、情感、主题，然后在每个类别里面又分为很多的小类，比如：语种这一类，分为华语、欧美、日语、韩语、粤语、小语种。音乐属于哪个语种、哪个类别等，应该是由音乐本身来决定，而不是由我们通过人为的感官来确定其类别，因为人为对音乐标注类别标签总会有不匹配的情况，也就有可能最终导致推荐效果不好的问题存在。

比如：华语的音频含蓄些，如图3-2，欧美的音频可能更加奔放些，如图 3-3

图3-2 图3-3

从图3-2和图3-3，那么是不是可以通过音频的不同来推断出是华语或者欧美，即通过音乐文件的内部信息来推断出其类别，如图3-4

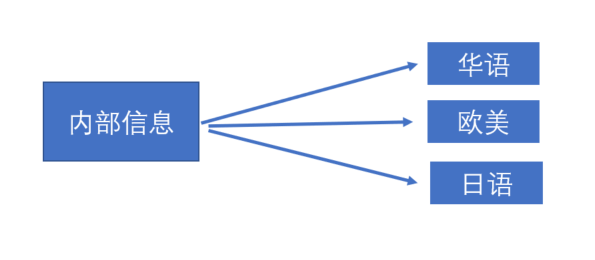


图3-4

那么现在已经知道这个音乐文件的语种，是不是可以通过同样的方式来判断其风格，如图3-5



图3-5

所以，从上面得知：音乐文件分类的关键在于获取其内部特征信息

四、音乐文件内部特征信息

对于一个音乐文件来说，我们要获得其内部特征信息，其实，只需要我们获得频谱上的包络信息，它在语音识别中被广泛用于描述特征。



图4-1

图4-1是一段语音的频率图，峰值表示语音的而主要频率成分，我们称之为共振峰，它携带了声音的辨识属性，用它来识别不同的声音。同时，频谱的包络就是一条连接这些共振峰点的平滑曲线。

综上可知：我们只需要把频谱的包络提取出来即可

五、音乐文件特征的提取

对于音乐文件特征的提取，我们需要引入一个概念MFCC(Mel Frequency Cepstral Coefficents)，中文名叫做梅尔频率倒谱系数，它能够自动识别我们语音与音乐中的有效特征。

首先，我们从名字上进行分析MFCC（梅尔频率倒谱系数），图5-1.

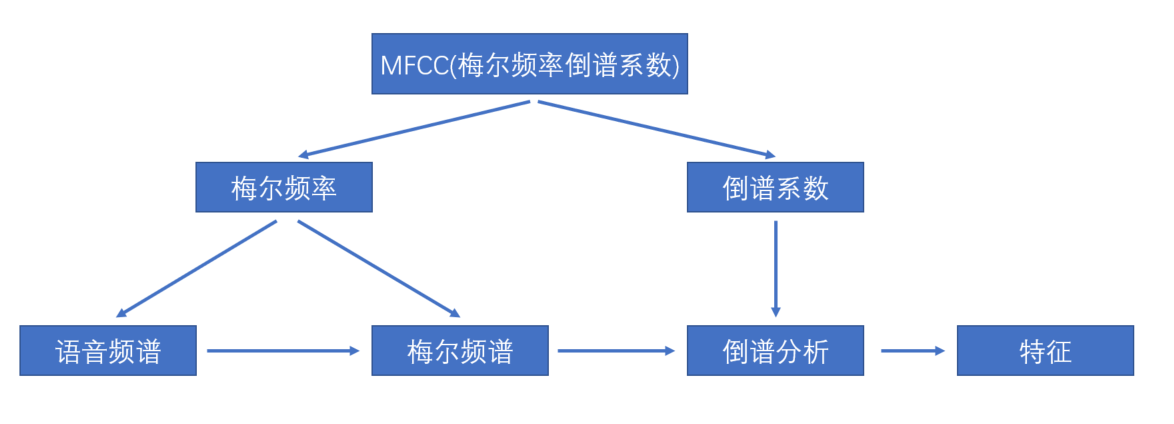


图5-1

我们是不是要先得到梅尔频率，再进行倒谱。那么再次进行分解，先得到语音的频谱，再进行梅尔频谱，最后进行倒谱分析，获取到我们的特征值。

以上是MFCC的感官流程图。现在我们一步步进行细节操作。

5.1 获取语音频谱

描述语音信号的常用方式是用声谱图(行轴为时间(Time-s),纵轴为振幅(Amplitude-dB))来表示，如图5-2所示

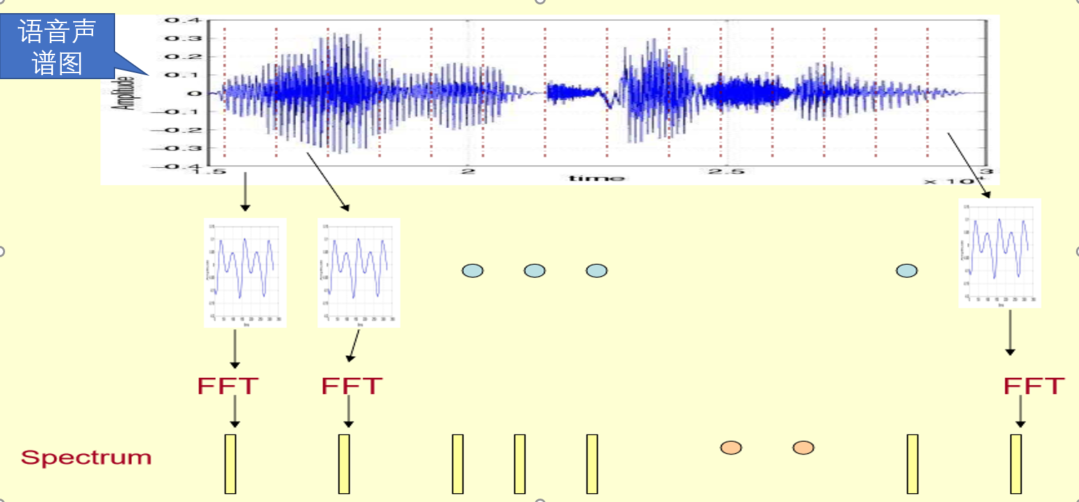


图5-2

首先，把这段语音声谱分为很多帧，把每帧语音通过短时FFT(傅里叶变换)计算得到频谱。

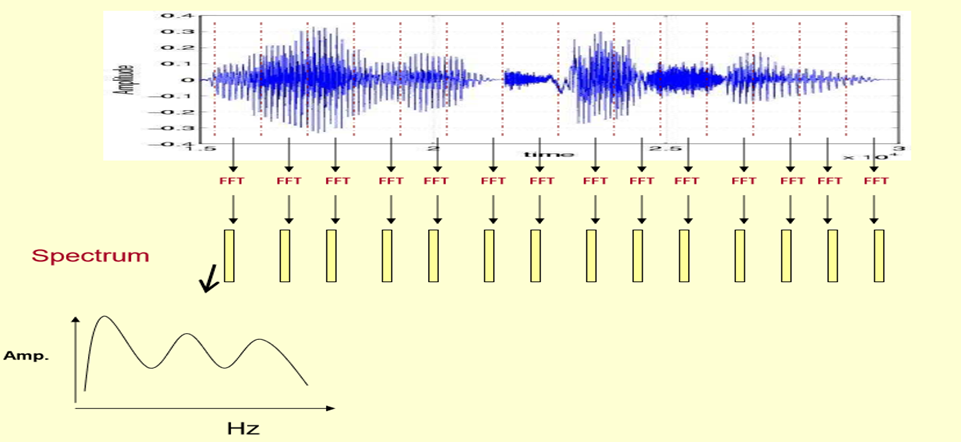


图5-3

得到的频谱图如图5-3所示，表示的是频谱表示频率和能量的关系。横轴为频率(Frequency-Hz)，纵轴为振幅(Amplitude-dB) 。

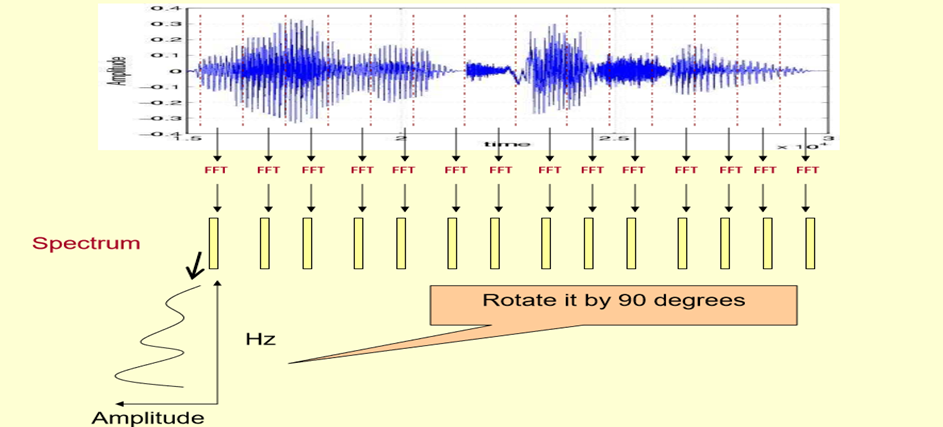


图5-4

如图5-4所示，我们将通过FFT得到的频谱图进行逆时针旋转90度，得到纵轴为频率(Frequency-Hz)，横轴为振幅(Amplitude-dB) 的频谱图。

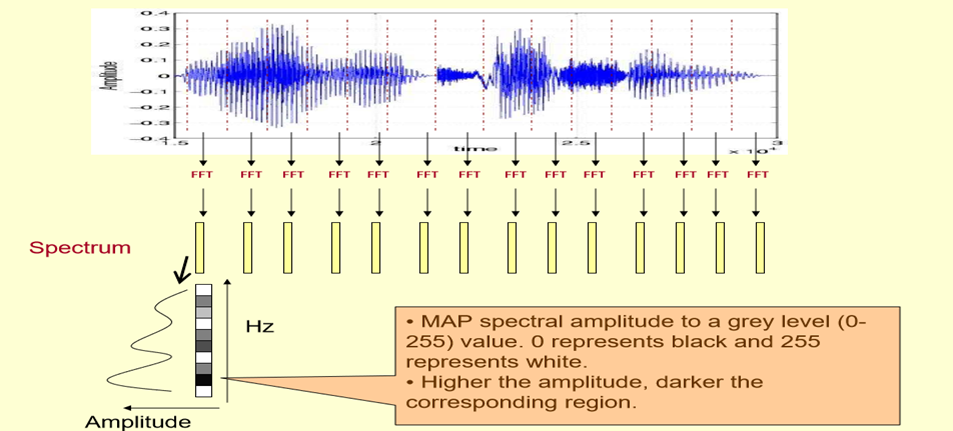


图5-5

如图5-5所示，将振幅进行映射,用一个灰度级来表示。可以理解为将连续的振幅量化为256个量化值。其中0代表黑，255为白色。振幅越大，对应区域越黑，振幅越小，对应区域越白。

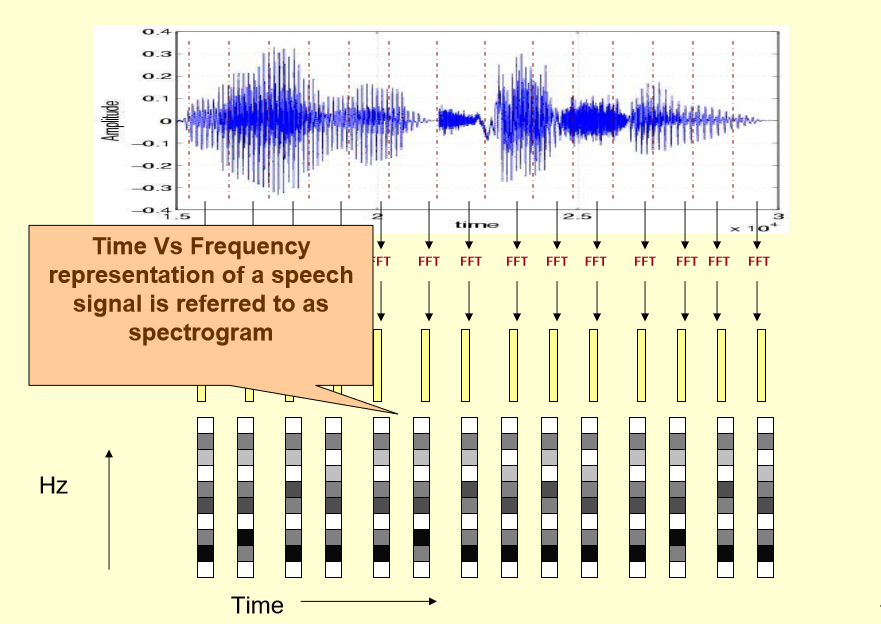


图5-6

现在我们把每一帧的语音都通过FFT的变换，然后把它们结合起来，这样就可以得到一个随着时间变化语音的频谱图(以时间为行轴，以频率为纵轴)，如图5-6所示。

为什么我们要把语音变成一个光谱图呢？那是因为在光谱图中能更好的观察音素及其性能。如图5-7所示

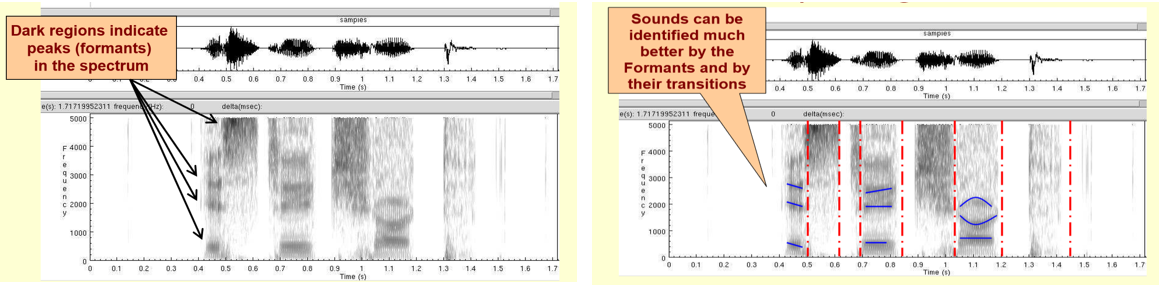


图5-7

总结：通过对语音进行分帧进行时频变换，得到每一帧的FFT频谱再将各帧频谱按照时间顺序排列起来，得到时间-频率-能量分布图。

5.2 倒谱分析

从第四小节：音乐文件内部特征信息中，我们可以得知只需要把频谱的包络提取出来即可。而倒谱分析就是为了这一目标而设置的。

如图5-8所示，原始的频谱由两部分组成：包络和频谱的细节，因此只需要把这两部分分离开，就可以得到包络了。

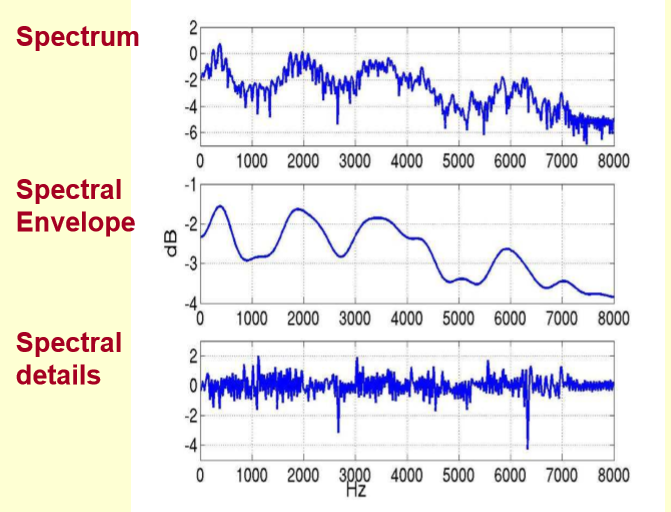


图5-8

同时，在图5-8中，我们也可以看到包络主要是低频成分，而高频主要是频谱的细节。为了能更好的得到包络信息，我们可以使那些振幅较低的成分相对高振幅成分得以拉高，以便观察掩盖在低幅噪声中的周期信号。所以我们将频谱转换成对数频谱（就是将各谱线的振幅都作了对数计算（纵坐标的单位是dB（分贝））。

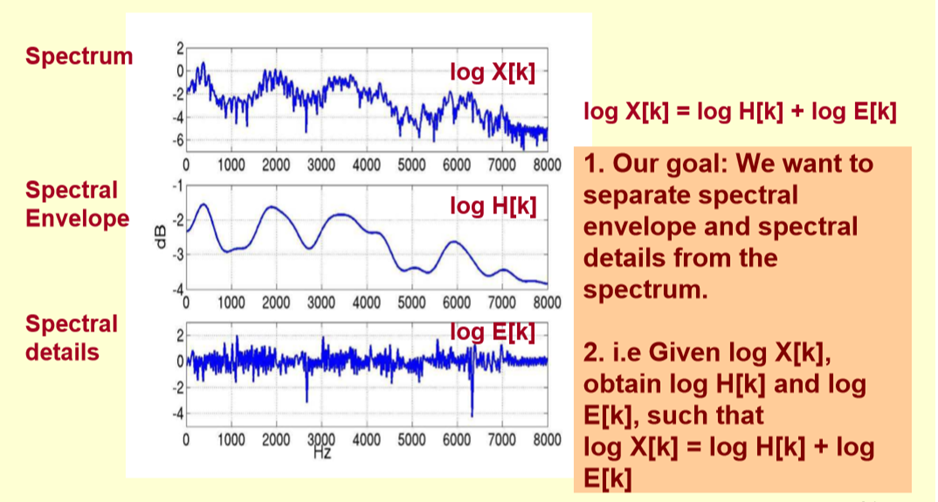


图5-9

图5-9所示，在给定logX[k]的基础上,使logH[k]和logE[k]满足logX[k]=logH[k]+logE[k].为了达到这个目标，我们需要Play a Mathematical Trick。这个Trick是什么呢？就是对频谱做FFT。在频谱上做傅里叶变换就相当于逆傅里叶变换Inverse FFT (IFFT)。需要注意的一点是，我们是在频谱的对数域上面处理的，这也属于Trick的一部分。这时候，在对数频谱上面做IFFT就相当于在一个伪频率（pseudo-frequency）坐标轴上面描述信号。

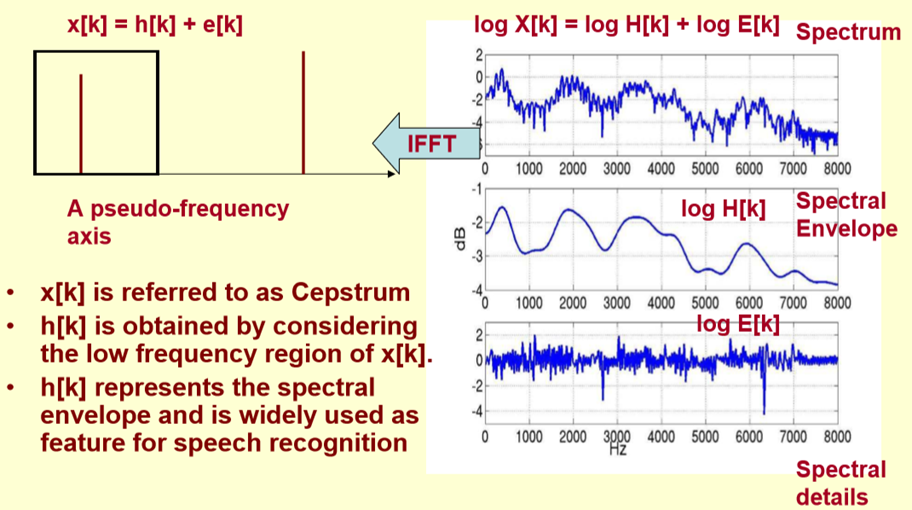


图5-10

如图5-10所示，我们把x[k]称为倒谱。h[k]是通过考虑x[k]的低频区域获得的。h[k]表示谱包络，被广泛用作语音识别的特征。

即，h[k]是x[k]的低频部分，因此将x[k]通过一个低通滤波器就可以得到h[k]了，也就是频谱的包络（连接所有共振峰值点的平滑曲线），即所谓的特征。

那么现在总结下倒谱分析的过程：

1、将原语音信号经过傅里叶变换得到频谱：X[k]=H[k]E[k]；若只考虑幅度就是：|X[k] |=|H[k]||E[k] |；

2、在两边取对数：

log||X[k] ||= log ||H[k] ||+ log ||E[k] ||

3、再在两边取逆傅里叶变换得到：x[k]=h[k]+e[k]。

5.3 获得Mel频谱

要想获得Mel频谱，只需要将得到的语音频谱通过过一组Mel滤波器就得到Mel频谱，也就是说我们只需要进行Mel频率分析即可获得。

Mel频率分析是基于人类听觉感知实验的。实验观测发现人耳就像一个滤波器组一样，只关注某些特定的频率分量（人的听觉对频率是有选择性的）。也就说，它只让某些频率的信号通过，而压根就直接无视它不想感知的某些频率信号。这些滤波器在频率坐标轴上却不是统一分布的，在低频区域有很多的滤波器，分布比较密集，在高频区域，滤波器的数目就变得比较少，分布很稀疏。如图5-11所示

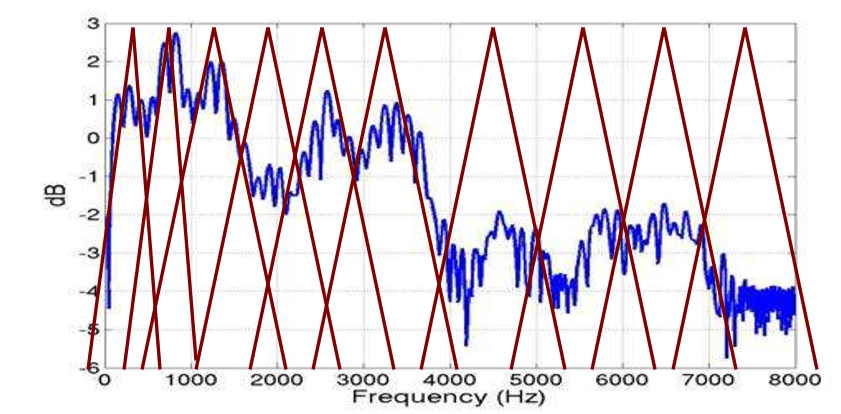


图5-11

将普通的频率转化成Mel频率的公式：

它可以将不统一的频率转化为统一的频率，也就是统一的滤波器组。

5.4 Mel频率倒谱系数

综合5.1节—5.3节中的内容，我们可以得知，要获得MFCC，只需要语音通过FFT转化成频谱，然后通过一组Mel过滤器就得到Mel频谱，公式为：log X[k] = log (Mel-Spectrum)。

然后取对数：

log X[k] = log H[k] + log E[k]

进行逆变换：x[k] = h[k] + e[k]。

如图5-12中的左图所示，然后把每一帧语音的特征得到综合起来，得到

如图5-12中的右图所示

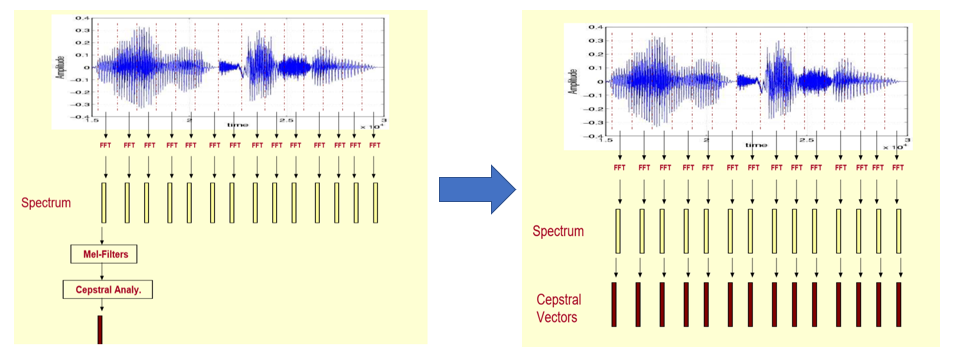


图5-12

六、 获得特征之后的处理

我们通过MFCC获取到了音频的特征，特征的形式，如图6-1中左图所示，将其进行旋转，是以Hz为特征，以时间观测值的形式。如图6-1中右图所示。

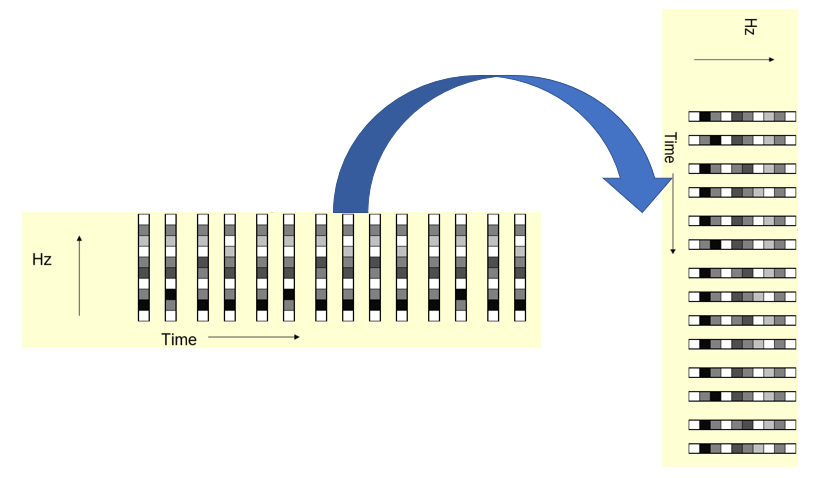


图6-1

若我们假设一段语音有m帧，有 10个特征值，那么我们的数据集将会是

m\*10。如果说有1000个文件，那么整个数据集对应的shape是1000\*m\*10,考虑到机器学习当中，一般希望数据集的维度是2维的，那么可以把它们进行合并，转变成（1000，m\*10）的2维，数据矩阵的特征从10个特征转变成了m\*10个特征。矩阵的每一行代表的是一个音频文件。

因为每一个音频文件的时长是不一样的，即每个音频的m是不一致。这样使得我们处理音频的维度可能是不一致的。同时对特征的处理是基于对音频文件的了解，考虑到很多人对音频文件掌握得理论不够，我们在这里讲解两种简单的不同处理方式：

方案一、使用AudioSegment当中的相关API，对音频文件进行一个预处理。比如：1、使得每首歌的音频数据的长度一致、各个特征的信息一致；2、把每个音频的信息从矩阵变成向量；3、对音频文件的长度进行截取或者填充。这样保证了每个音频文件m的一致性。

方案二、在MFCC指标上，提取更高层次的信息特征。可以使用简单的特征与特征之间的关联程度的标准来进行处理。比如：使用均值与协方差的方式来进行。



图6-2

如图6-2.把一个音频的数据，求取它的每一个特征的均值，得到一个向量。虽然达到了我们的维度一致性的目标，但是未必太简单了，而且会使得我们处理之后的特征与原始特征相差比较大。所以这个时候，我们需要将其维度进行扩充。考虑用协方差的方式来反映特征与特征之间的关系，因为协方差矩阵是一个对称矩阵，所以我们只要获得它的上三角或者下三角对应的信息。如图6-3，将特征维度从4维扩充至10维。

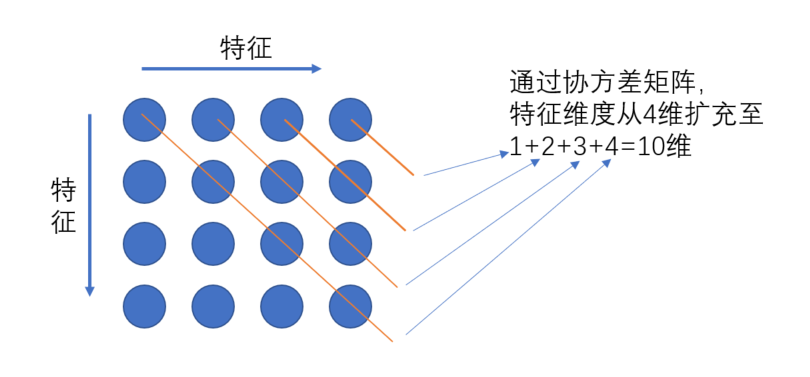


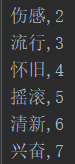
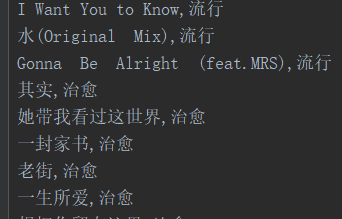
图6-3

同时，也可以把均值获取的特征与西方差矩阵获取到特征进行组合。

七、建模之前准备工作

7.1 准备音乐文件

训练文件和测试文件，以及每首歌的类型和每个类型对应的编码编号。最后，需要建立一个空文件来存储我们提取的每一个音乐文件的特征

7.2 安装库

使用scipy库中的方法对wav格式音乐文件的读取，然后使用python\_speech\_features中的MFCC相关方法对语音数据进行特征提取



对于MP3或者非WAV格式的文件无法通过scipy库进行读取操作，所以需要先将其转化成WAV格式文件。

一般可以通过pydub库中的AudioSegment进行操作。（需要安装ffmpeg服务或者libav服务）



图7-1是我们对音乐文件进行特征的提取。最后我们把每一个音乐文件的特征保留在一个文档中，如图7-2所示，每一行代表的是一首歌曲。

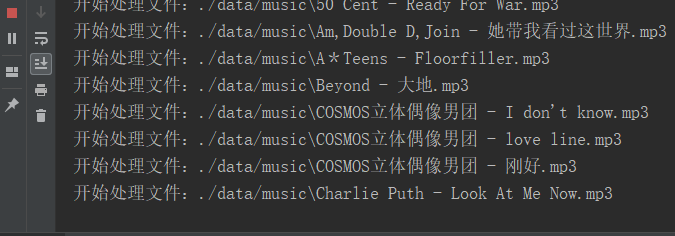


图7-1

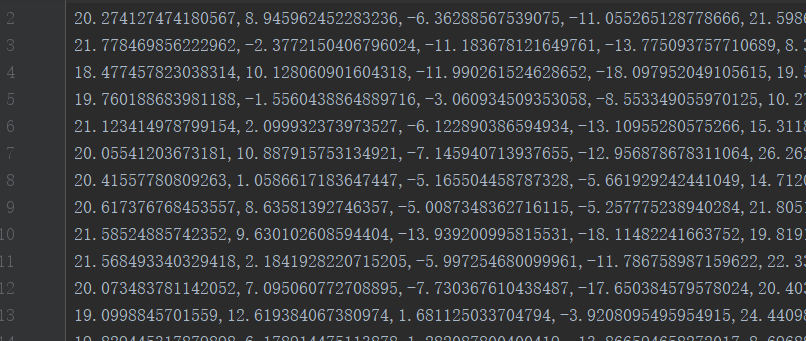


图7-2

下载ffmpeg安装包地址：

http://ffmpeg.org/download.html

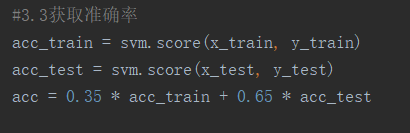
八、建立模型并进行预测

在这里使用svm算法实现模型的构建，用模型交叉验证的API，来对模型最优参数进行选择。然后我们进行多次模型训练，最终将模型效果最好的那个模型使用joblib.dump()函数保存起来，最后我们使用joblib.load()函数对保存的模型进行加载，得出我们的预测结果，并输出评分。

注意：如果你使用的样本数据有点少，所以模型的衡量指标采用了训练集和测试集加权准确率作为衡量指标。

比如：

score=0.35\*train\_score+0.65\*test\_score



同时：

经过多次训练，在训练上可以达到90%以上的准确率，在测试集上至少可以达到60%以上的准确率。如图8-1

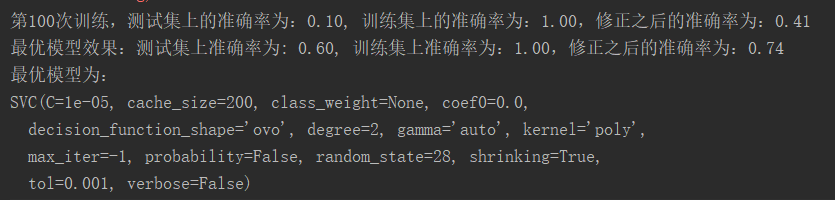


图8-1

然后对音乐文件进行预测，四首歌里面只有最后一首歌预测错误，把类型为怀旧的预测成清新类型。如图8-2所示，这主要的还是样本训练的数据太少了。

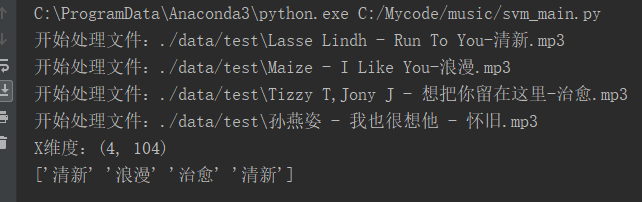


图8-2

大家要清楚明白一个概念，机器学习的模型效果是基于大量的数据基础之上进行训练的。

九、总结

以上就是我们用机器学习的svm对于音乐文件进行分类的过程。我们要掌握的知识有：为什么要用内部信息进行分类？使用MFCC来进行一个特征的提取及MFCC的原理，提取特征之后，我们该怎么去处理特征？svm的原理，如何使用GridSearchCV进行交叉验证，获取最优的模型参数。最后，是保存模型，以及调用模型来进行预测。