1. **解决上周的疑惑**

　在使用SVM模型中要输入的有测试集数据和标签，训练集数据和标签，因为在输入之前我们都需要给它打上标签将它归为某一类，如图所示

svmtrain为训练模型函数，svmpredict为测试（预测集），每个函数分别输入相对应的标签。



而在用libsvm库进行模型训练时需要分三个步骤：1、训练 2、测试 3、预测

1、训练——就是用训练数据集，不管你采用那种寻优方式，得到相对的最优参数，训练模型。

2、测试——用刚刚得到的模型svmtrain，对测试数据进行测试，此时测试数据集的label是已知的，这主要是用来对刚刚的模型的检测，或者是对参数的检测、或者是对模型的泛化能力的检测。此时会得到一个准确率，这时的准确率是有用的，是有实际意义的，因为原来的label是已知的。

3、预测——预测就是对未知类别的样本在测试确定了模型有好的泛化能力的情况下的预测分类，这步才是真的预测能力功能的实现。此时数据的label随便给了，这样是为了满足libsvm对数据格式的要求。此时也会得到一个准确率，但是这个是没有实际意义的，因为原始的label是随便给的，没有意义，我们只是关心的是最后得到的类别号——达到分类的目的。

因此在我的程序中分有train（训练）和test（测试）两个数据，这两个数据的标签都是已知的，并且由人工给它打上标签，而只有你已经用训练集和测试集训练好一个svm模型后，即能达到最优分类的参数确定，将该参数用于未知数据进行预测是，这个预测数据的标签则是随便打的，这个时候svm中的accuracy是没有意义的，我们需要关心的只是predict\_label这个标签类别。

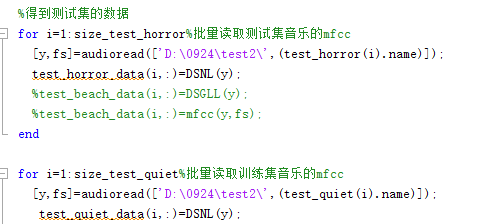
**二、训练和测试**

上一周用的是环境音进行简单的分类，而这次用的是配乐即纯音乐和有人声的音乐。

* 配乐分类

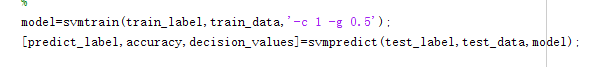
　　还是实现的二分类，有安静（quiet）和恐惧(horror)两种感情。其中horror和quiet有9首曲子，分别用其中的6首来进行训练，3首用来测试；提取的特征是短时能量和短时过量率这两个时域特征，和频域用快速傅里叶变换FFT的到的频域特征，我调整了svm的参数后的到了不同的正确率。

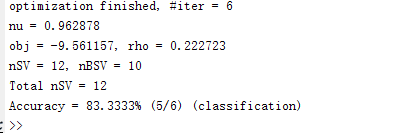
（1）提取短时能量特征，当惩罚参数-c和核函数参数-g分别为1和1的时候得到的正确率是100%



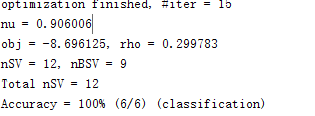


当我将参数改成1和0.5的时候就会发现正确率下降

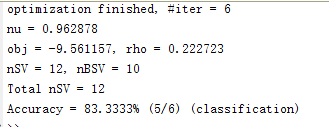




（2）在用短时过零率这个特征时，同理，当惩罚参数-c和核函数参数-g分别为1和1的时候得到的正确率是100%。

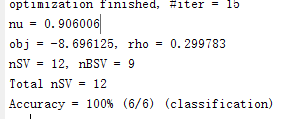


-g改成0.8是，正确率也下降。



1. FFT得到的特征





我们可以发现，在进行少数据的二分类情况下，ｓｖｍ的准确率还是较高的

* 音乐分类

　　是从音乐网站中依照标签的分类，在伤感，开心，孤单，三种情感中分别下载了14首音乐，其中10首用来作为训练集，4首用来作为测试集。根据所看的论文中要截取音频的高潮部分，即四分之一的时间到二分之一的时间段，因此在实现的过程中先把音乐的ｔ／４和ｔ／４时长截取出来，采取不同的音频特征和svm参数时，得到的是不同的准确率，而且在实现3分类的情况下分类的准确率就不像二分类那样的到100%的准确率。

　利用短时能量的平均值和方差参数无论如何调整都是大约在60%左右，即12个测试数据中大约有7个是正确的；短时过零率只有33%的正确率；快速傅里叶变换也只有50%的正确率。

当我将第三个标签去掉的时候即只剩下伤感和开心两种情感，发现正确率高达８７.５％，只有一个出现预测错误。由此可见在进行多分类的情况下应该采取不同的方法，不能直接用一个分类器。

正确率低的原因可能有

1. 有可能我的样本区别不是很明显；
2. 或者是提取的特征不具有代表性；
3. ｓｖｍ模型的参数调整不正确。
4. **下次要研究的问题**

**（１）ＳＶＭ实现多分类**

其实我们可以发现SVM是一种典型的两类分类器，当分类的数据很多时，使用svm的多分类方法。而不能简单的使用一个分类器，是通过组合多个二分类器来实现多分类器的构造。即有一对多和一对一两种方法。

Ａ．一对多（one-versus-rest,简称OVR 　SVMs）：训练时依次把某个类别的样本归为一类,其他剩余的样本归为另一类，这样k个类别的样本就构造出了k个SVM。分类时将未知样本分类为具有最大分类函数值的那类。

假如有四类要划分（也就是4个Label），他们是A、B、C、D。

于是在抽取训练集的时候，分别抽取

（1）A所对应的向量作为正集，B，C，D所对应的向量作为负集；

（2）B所对应的向量作为正集，A，C，D所对应的向量作为负集；

（3）C所对应的向量作为正集，A，B，D所对应的向量作为负集；

（4）D所对应的向量作为正集，A，B，C所对应的向量作为负集；

使用这四个训练集分别进行训练，然后的得到四个训练结果文件。

在测试的时候，把对应的测试向量分别利用这四个训练结果文件进行测试。

最后每个测试都有一个准确率的结果f1(x),f2(x),f3(x),f4(x)，在哪一个分类器的准确率最高就将它归在该类。

Ｂ．一对一（one-versus-one,简称OVO　SVMs）：其做法是在任意两类样本之间设计一个SVM，因此k个类别的样本就需要设计k(k-1)/2个SVM。当对一个未知样本进行分类时，最后得票最多的类别即为该未知样本的类别。Libsvm中的多类分类就是根据这个方法实现的。

假设有四类A,B,C,D四类。

在训练的时候选择A,B; A,C; A,D; B,C; B,D;C,D所对应的向量作为训练集，然后得到六个训练结果，在测试的时候，把对应的向量分别对六个结果进行测试，然后采取投票形式，最后得到一组结果。

(A,B)-classifier 如果是A win,则A=A+1;otherwise,B=B+1;

(A,C)-classifier 如果是A win,则A=A+1;otherwise, C=C+1;

(A,D)-classifier 如果是A win,则A=A+1;otherwise, C=C+1;

(B,C)-classifier 如果是A win,则B=B+1;otherwise, C=C+1;

(B,Ｄ-classifier 如果是A win,则B=B+1;otherwise, D=D+1;

(C,D)-classifier 如果是Ｃwin,则C=C+1;otherwise,D=D+1;

The decision is the Max(A,B,C,D)，即看看在逐个相互比较的过程中，它作为某一类得到的票数最高它就属于哪一类。

**（２）特行提取的另一种方法**

在读论文中还发现另一种特征提取的方法，就是解析ＭＩＤＩ文件，ＭＩＤＩ文件以字节的形式对乐曲进行描述，使用数字编码描述乐谱的基本特征。多数MIDI文件都是多音轨文件，其中的主旋律和伴奏旋律都分布在不同的音轨上，通常将主旋律所在的音轨称之为主音轨，其包含了音乐的主要情感信息。因此通过分析主音轨的情感就可以识别出该段音乐所表达的感情。通过对MIDI文件的解析，可以得到的基本数据信息包括音高、音长、音强和节拍等信息（具体的解析方法还没有去研究）。

**（３）ＳＶＭ主动学习**

　　从程序中可以发现，我们所用的标签都是人工分类的，这种属于被动监督学习方法，是通过用户对训练样本进行人工标注，然后进行训练得到分类器，最后对未知样本数据进行分类，这种被动学习方法的分类效果需要大量标注好的训练样本参与训练才能有所保证，但是标注这些训练样本所耗费的人力和时间是巨大的。因此，如何有效地利用未标注样本来训练分类器具有十分重要的应用价值。在利用未标注样本进行学习的众多方法中，主动学习是最行之有效的方法之一。主动分类就是要求分类器可以自动地从给定的未标注的样本中选择出最有价值的样本，提供给用户进行标注，然后将新标注的样本加入到训练集再次训练学习器，如此循环往复，从而在多次迭代中不断改善学习器性能。