参赛队号 # 1023

第六届“认证杯”数学中国

数学建模网络挑战赛

承 诺 书

我们仔细阅读了第六届“认证杯”数学中国数学建模网络挑战赛的竞赛规则。

我们完全明白，在竞赛开始后参赛队员不能以任何方式（包括电话、电子邮件、网上咨询等）与队外的任何人（包括指导教师）研究、讨论与赛题有关的问题。

我们知道，抄袭别人的成果是违反竞赛规则的, 如果引用别人的成果或其他公开的资料（包括网上查到的资料），必须按照规定的参考文献的表述方式在正文引用处和参考文献中明确列出。

我们郑重承诺，严格遵守竞赛规则，以保证竞赛的公正、公平性。如有违反竞赛规则的行为，我们将受到严肃处理。

我们允许数学中国网站(www.madio.net)公布论文，以供网友之间学习交流，数学中国网站以非商业目的的论文交流不需要提前取得我们的同意。

我们的参赛队号为：1023

参赛队员 (签名) ：

队员 1：余燕团

队员 2：郭 维

队员 3：康 媛

参赛队教练员 (签名)：

参赛队伍组别：研究生组

参赛队号 # 1023

第六届“认证杯”数学中国

数学建模网络挑战赛

编 号 专 用 页

参赛队伍的参赛队号：（请各个参赛队提前填写好）：1023

竞赛统一编号（由竞赛组委会送至评委团前编号）：

竞赛评阅编号（由竞赛评委团评阅前进行编号）：

参赛队号 # 1023

2013年第六届“认证杯”数学中国数学建模网络挑战赛

题　　目　基于GMM的音乐风格分类模型研究

关 键 词　EM算法；GMM；特征参数；音乐风格向量；混淆矩阵

摘　　要：

随着计算机技术和通信技术的飞速发展，各式各样的信息呈现急速增长的态势，人们也时刻能够够接触到大量多媒体形式的内容，如图形图像、音频、视频等。但是随着数据量的快速增长，特别是大数据时代的到来，如何自动对感兴趣的内容进行快速地分析管理就成为了一个亟需解决的问题。特别是浩海如烟的音乐信息，人们需要快速高效的方法对它进行分类和管理，以便更好的应用在音乐推荐系统、KTV点唱以及网络电台选歌等诸多领域中。

音乐自动分类是解决音频结构化问题和提取音频结构化信息和内容语义的关键，是当前音频分析领域中的研究热点与难点，在音频检索，视频摘要和辅助视频分析等诸多领域中有重要的应用价值。本文首先建立基于模糊分类方法的流行音乐风格类型的数序模型，然后对音乐的物理属性，即音色、旋律和和弦、节奏、语义特征，进行了初步的讨论研究，由于区别一首歌曲的重要特性是音色，而音色有多种特征参数来表征，我们选取了LPC倒谱系数和Mel频率倒谱系数作为最终参数。最后我们建立了基于GMM的音乐风格分类模型，采用K-均值聚类算法进行模型参数初始化，利用EM算法进行GMM模型的参数估计，同时，得到似然比并进行归一化，对每一首音乐进行特征参数以及似然比的计算，计算结果与所属类别的似然比进行比对分析，似然比越接近就越倾向与该风格类型。为了提高风格类型的分辨准确率，我们进行了双重音乐风格的探讨，并用音乐风格向量来表示，这样在给定阈值的情况下，有可能一首歌曲就可以划分为两类，对于所分类的性能分析，我们采用了平均准确率和 R-准确率，值越大，则分类准确度越高。

在模型建立之后，利用搜集到的数字音频数据进行实验研究，实验过程中结合MATLAB和Python软件编程并得到结果，分析结果表明本文建立的数学模型具有较好的分类精度和效率，特别在低噪声比交互环境下，对单一音乐风格类型的分类中，R-准确率都达到90%左右。本文的数学模型比较好地研究了音乐风格类型的分类问题，但在实际问题中，往往需要处理多种音频信号的问题，比如不同乐器，不同歌手的语音音质，以及音乐与噪音等的检测。因此，该课题的进一步目标为音频信号的多类检测与分类以及新音乐的检测与识别问题。

参赛队号　1023

所选题目　B

参赛密码 _____

（由组委会填写）

参赛队号 # 1023

The Study Of Music Genre Classification Model Based On GMM

Abstract: With the rapid development of computer and communication technology, a wide range of information showed a rapid growth trend, and people are always able to come into contact with a large number of multimedia contents, such as graphics, audio, video, and so forth. However, with the rapid growth of the amount of data, especially the arrival of the era of big data, how to automatically interested content analysis and management has become an urgent problem. People need it the classification and management in a fast and efficient way under the such circumstances, in order to apply in many other areas of the music recommendation system, KTV to sing as well as Internet radio to select songs.

Automatic classification of music is to solve the problem of audio-structured and extract audio structured information and content semantics key, the current audio analysis is a hot topic and become more and more difficulty, it is very important in other areas of audio retrieval, video summary and analysis of auxiliary video application value. This paper formulates a number of popular music styles order model based on fuzzy classification method, then the physical attributes of music, sounds, melodies and chords, rhythm, semantic features, a preliminary discussion of study, an important difference between a song characteristic tone, timbre parameters to characterize a variety of characteristics, we select the LPC cepstrum coefficients and Mel frequency Cepstral Coefficients as the final argument. Finally, we established a style of music based on GMM classification model, using the K-means clustering algorithm to initial the model parameters, EM algorithm for GMM model parameter estimation, at the same time, the likelihood ratio and normalized for each music feature parameters and likelihood ratio, calculated results with the category of the likelihood ratio to compare and analyze the likelihood ratio the closer the more tendency with that type of style. In order to improve the type style to distinguish accurately rate discussion of the doublet music style, and the music style vector to which means, so in the case of to a fixed threshold of, there may be a song can be divided for the two categories for classification performance analysis, we used the average precision and R-accurate rate, the larger the value, the higher the accuracy of the classification.

Having established the model, the use of the digital audio data collected experimental study, during the experiment we used the MATLAB and Python program to obtain the results we expected,and the results of the analysis showed that the mathematical model in this paper has a better classification accuracy and efficiency, particularly in the low-noise ratio of the interactive environment, a single musical style type classifications R-accurate rate reached about 90%. The mathematical model to study the classification of the type of music style, but in practical problems, often need to deal with a variety of audio signal, such as the different instruments, different singers' voice quality, as well as music and noise detection. Therefore, a further objective of the subject into an audio signal detection and classification, and detection and identification of new music.

Keywords:EM algorithm;GMM; Feature parameters;Music Style Vector; Confusion matrix

参赛队号 # 1023

目 录

参赛队号 # 1023

1 问题背景

“凡音之起，由人心生也。人心之动，物使之然也，感於物而动，故形於声。声相应，故声变，变成方，谓之音。比音而乐之，及干戚、羽旄、谓之乐。”

——《礼记·乐记》

音乐是人类最古老、最具普遍性和感染力的艺术形式之一，是人类通过各类声响的和谐有序排列组合来表达思想和感情、实现相互交流的特殊语言[1]。而随着现代社会技术的发展，尤其是多媒体技术和网络技术的发展，音乐通过各种电视频道、广播频道以及网络途径在人们生活中得以广泛传播及流行，并日渐成为人们生活中不可或缺的部分。传统的音乐信息识别多是根据名称、作曲家或歌词的关键字来搜索。随后出现了根据音乐类型, 音符节拍和旋律来检索的方法[2]。

2 问题重述

随着互联网的发展，流行音乐的主要传播媒介从传统的电台和唱片逐渐过渡到网络下载和网络电台等。网络电台需要根据收听者的已知爱好，自动推荐并播放其它音乐。由于每个人喜好的音乐可能横跨若干种风格，区别甚大，需要分别对待。这就需要探讨如何区分音乐风格的问题。

在流行音乐中，传统的风格概念包括 Pop(流行)、Country(乡村)、Jazz(爵士)、Rock(摇滚)、R&B(节奏布鲁斯)、New Page(新世纪)等若干大类，它们分别可以细分成许多小类，有些小类甚至可以做更进一步的细分。而每首歌曲只能靠人工赋予风格标签。这样的做法有许多不足：有的类别之间关系不清楚，造成混乱；有的类别过度粗略或精细；有的类别标签没有得到公认；有的音乐归属则存在争议或者难以划归。请你建立合理的数学模型，对流行音乐的风格给出一个自然、合理的分类方法，以便给网络电台的推荐功能和其他可能的用途提供支持。

3 模型假设

1 假设每个音乐收听者对歌曲的选择只受歌曲类型与主观因素的影响。
2 假设音乐风格类型只受音乐本身的物理属性以及乐理属性的影响。
3 假设音乐风格分类只考虑本文中所考虑涉及的几个描述项。
4 假设只考虑文章中所涉及的若干流行音乐风格分类类型。
5 假设文中所涉及的几个音乐描述项数据是易得的。
6 假设对音乐风格类型的分类不受乐器音质的影响。

4 符号系统

符号	说明
ϕ_{ij}	特征函数
$U_i(X_r)$	隶属度函数
$h(n)$	冲激响应
$X(k)$	音乐信号频域
$H_m(k)$	带通滤波器
$c(n)$	Mel 频率倒谱系数
r_i	每个风格类型似然比
CM	混淆矩阵
γ	分类有效率

注：表中没有列出的符号，文中使用时会给予说明。

5 问题的分析与模型准备

5.1 问题的分析

我们从流行音乐的物理属性、乐理属性及心里属性三个不同维度进行分析。从心理学角度来看，音乐风格是一个相当主观、模糊的概念，我们可以尝试利用模糊分类的方法进行流行音乐的风格分类初步研究，但这并不是一种普遍的风格分类标准。所以我们从音乐的物理属性入手，结合音乐理论中公认合理的音乐风格种类，利用传统的音频信号处理方法构建了一种风格分类方法，其大致流程如图 1 所示。

本文提出的基于高斯混合模型(GMM)的音乐风格识别方法具有训练与识别两个模块。在训练部分，首先对用于训练音乐进行预处理及特征提取，取到基于帧的分类特征向量。在对训练的特征帧进行聚类分析来确定基于 GMM 的风格模型参数的初值，然后利用期望最大化(EM,Expectation-Maximization)算法，对风格模型的参数进行训练，进而得到每种风格音乐的模型参数。在识别部分，将基于帧的待分类音乐特征向量与风格模型进行差异计算，求出特征帧与每类预设风格模型的似然比，把待分类音乐归入相似度最大的模型对应的风格，并将似然比利用歌曲风格向量表示到雷达图中，基于风格向量给出风格判别结果及其可视化分析。

利用*Visio* 绘制音乐风格分析方法的总体流程如下：

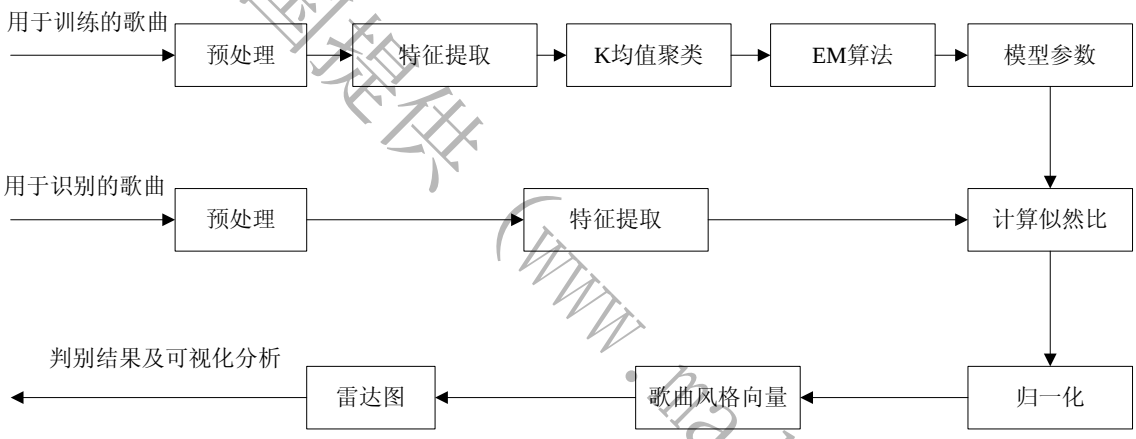

图 1 音乐风格分析方法的总体流程

5.2 模型的准备

5.2.1 音乐风格特征参数

在实际应用中，一首歌曲很难使用一个精确的符号表示。音频采样洋被是数字音乐分析中最简单的数据形式，它是通过对模拟信号采样得到的声音波形文件。由于波形文件的信息密度较低，而且音乐文件数据的数量巨大，因此不能直接应用于自动分析系统进行音乐风格分类，这需要先对音乐文件进行特征提取。

特征提取是模式识别的第一步。只要能够提取出有用的特征数据，就可以使用任何分类器对特征数据进行相应的分类。在音频信号中，应用较多的特征属性包括音乐旋律、和谐、节奏及音色等。

5.2.2 音色

音色是代表声音一种重要属性，在音乐风格分类识别中起着重要的作用。音色是声音的基本特征，即使音调和音量完全相同，但是声音音色不同时，声音听起来也是不一样的。音色的研究包括变换域和时域两个方面，当前大部分研究者致力于音色的频域特性研究，少部分学者则研究时域特性。

Peeters 较为详细地描述了音色的特征参数[2]，这些参数大部分已经获得各音乐风格研究者所认可。Peeters 所提出的这些参数被称为底层参数，它们能较好的描述声音量级，

MPEG-7 音频标准规范中就已经应用了其中部分参数[3]，底层特征参数主要包括：短时特性、能量特性、谱形态特征和直觉特征，此外还可以使用上述参数的扩展参数，如一阶梅尔顿倒谱差分系数和二阶梅尔顿倒谱差分系数等，使用这些参数可以提高分类器识别的准确度。

5.2.3 旋律和和弦

和弦即音乐音调的同步性和弦向性，而旋律则可以看作是多个连续音调连接在一起的一个实体。和弦有时被称为音乐的垂直元素，旋律则被称为音乐的水平元素。和弦和旋律分析作为基础音乐理论已经被音乐学家们所熟知，并被多位研究者应用于音乐风格分析中，并通过实验结果验证了分类效果。

Gomez[4]和 Zoia[5]等人详细描述了旋律和和弦的提取过程，Klapuri[6]介绍了多个基频同时存在情况下旋律的估计。

5.2.4 节奏

Gouyon 和 Dixon[7]论述了基于节奏的音乐风格识别系统，该系统可从不同的角度对音乐风格进行分类。然而，这种音乐风格识别系统仍然存在一定的漏洞，而使用更低层特征参数可以弥补这一缺陷，提高分类器的分类效果。

5.2.5 语义特征参数

语义特征参数也是一个重要的特征参数，在语音信号分析中经常使用这种特征量。它将整首音乐作为参考对象，不同的音乐风格使用不同的语义值表示。现在的研究中经常用截取的音乐片段来代替完整的音乐以简化计算过程，常用的做法是截取第 30~60s 范围内的音乐内容作为参考音乐片段。

Berenzweig 等人[8]利用语义信息对音乐歌手身份做识别分析。实验结果表明，歌手的声音比歌曲的背景音乐更能有效地识别出歌手的身份。

5.3 数据搜集

在实验数据库中约有 500 首歌曲，语言包括中文、英文、日语、泰语，部分英文歌曲来自 http://magnatune.com/、http://www.kuke.com/和 http://staff.aist.go.jp/m.goto ，其他音频文件来自本地硬盘。

数据存储格式如下表所示：

表 1 音乐数据文件存储格式

Piece NO.	Tr.NO.	Title	Aritist	Length	Variation
NO.1	Tr.01	Jive	Makoto Nakamura	3:22	Instrumentation 1
NO.2	Tr.02	For Two	Makoto Nakamura	6:15	Instrumentation 1
NO.3	Tr.03	Lounge Away	Takao Nagai	2:38	Instrumentation 1
……	……	……	……	……	……

6 模型的建立与求解

6.1 流行音乐模糊分类模型的建立

模糊分类是根据研究对象对类别的隶属度程度进行分类的一种方法。

假定用于识别的音乐共有 S 首，记为 $X_1, X_2, \ldots, X_s$；根据具体特征参数分成 m 类，$K_1, K_2, \ldots, K_m$。每一 $K_i\ (i=1,2,\ldots,m)$ 均为 $(X_1, X_2, \ldots, X_s)$ 上的模糊集合，每个类 K_i 的隶属度函数用 $U_i(X)(i=1,2,\cdots,m)$ 表示，隶属函数反映歌曲对分类的隶属程度。

1、求类的隶属函数矩阵

设首歌曲有 n 个特征参数，如 LPCC、MFCC 等。用 $d_1, d_2, \ldots, d_n$ 表示。设描述项 d_j 在

第i类中出现的概率为P_{ij}，从而可得矩阵

$$M_1=\begin{bmatrix} P_{11} & P_{12} & \cdots & P_{1n} \\ P_{21} & P_{22} & \cdots & P_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ P_{m1} & P_{m2} & \cdots & P_{mn} \end{bmatrix}$$

并设特征函数

$$\phi_{rj}=\begin{cases} 1, & \text{音乐}X_r\text{中有描述项}d_j \\ 0, & \text{其它} \end{cases} \qquad (r=1,2,\ldots,s;\ j=1,2,\ldots,m)$$

由M_1、ϕ_{rj}可确定类K_i的隶属函数

$$U_i(X_r)=\frac{\sum_{j=1}^{n}\phi_{rj}P_{ij}}{\sum_{j=1}^{n}\phi_{rj}}$$

用M_2表征类的隶属函数矩阵

$$M_2=\begin{bmatrix} u_1(X_1) & u_2(X_1) & \cdots & u_m(X_1) \\ u_1(X_2) & u_2(X_2) & \cdots & u_m(X_2) \\ \vdots & \vdots & \ddots & \vdots \\ u_1(X_s) & u_2(X_s) & \cdots & u_m(X_s) \end{bmatrix}$$

M_2为SX_m矩阵，即S行m列。

2、求交叉类$K_i \cap K_j$的隶属函数矩阵

由于音乐风格之间可能会相互交叉，所以某种音乐可能分到两个以上的类中，即类与类之间可能交叉。我们用M_3表示交叉类的隶属函数矩阵

$$M_3=\begin{bmatrix} u_{12}(X_1) & u_{13}(X_1) & \cdots & u_{ij}(X_1) & \cdots \\ u_{12}(X_2) & u_{13}(X_2) & \cdots & u_{ij}(X_2) & \cdots \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ u_{12}(X_s) & u_{13}(X_s) & \cdots & u_{ij}(X_s) & \cdots \end{bmatrix}$$

其中$u_{ij}(X_r)=u_i(X_r)\wedge u_j(X_r)$。

3、确定分类的阈值κ

由M_3可确定阈值κ，

$$\kappa<\underset{ij}{\wedge}\left[\underset{r}{\vee}\left(u_{ij}(X_r)\right)\right]$$

$\underset{r}{\vee}\left(u_{ij}(X_r)\right)$表示$M_3$中各列的最大值，$\underset{ij}{\wedge}\left[\underset{r}{\vee}\left(u_{ij}(X_r)\right)\right]$表示$M_3$各列最大值中的最小值。

4、将音乐归类

归类时，根据$K_i=\left\{X_r \middle| u_i(X_r)\geq\kappa\right\}$，划分音乐$X_r$。

6.2 特征分析与提取

音乐风格由不同的音乐要素构成，这些音乐要素的构成包括音律、色调等。而这些要素的体现形成即为音乐的特征参数。音乐的特征参数提取是基于音乐文件帧操作的。文件帧是指从声音波形文件中提取的一个数据组，它将一个音频文件分为多个连续的采

样点数为 n 的一组数据，一般帧长度 n 可以为 8192、4096、2048、1024 和 512 等不同的点数，前后两帧之间可以有一定的重叠。对音乐文件进行分帧处理后，可直接提取其时域特征参数，或者采用 FFT 等变换操作提取相应的变换域特征参数。

我们主要讨论研究流行音乐风格自动分析分类中常用的特征参数。

6.2.1 LPC 倒谱系数

LPC 指线性预测系数，线性预测技术广泛应用于语音编解码，从 3GPP 提出的 AMR 到 ITU-T 的 G729，LPC 均为核心组成部分。

LPC 倒谱系数(LPCC)基于 LPC 技术，目前许多成功的系统都采用 LPCC 作为其数字特征参量。由于音乐和语音有极大的相似性，因此大部分用于语音分析的参数亦可用于音乐风格分析。

LPCC 是复倒谱，求解时先将音乐信号进行 Z 变换，对 Z 变换之后的结果取对数，在对运算之后的结果进行 Z 反变换即可得到 LPCC 参数[11]。

首先假定进行分析的参考模型为全极点模型，其系统函数有如下格式：

$$H(z)=\left(1-\sum_{k=1}^{p}a_k z^{-k}\right)^{-1}$$

其中， p 为 LPC 线性预测器的阶数。假定次模型的冲激响应为 $h(n)$，则：

$$H(z)=\sum_{n=1}^{\infty}h(n)z^{-n}$$

对其取对数得：

$$\widetilde{H}(z)=\lg H(z)=\sum_{n=1}^{\infty}\tilde{h}(n)z^{-n}$$

将 $H(z)=\left(1-\sum_{k=1}^{p}a_k z^{-k}\right)^{-1}$ 代入 $H(z)=\sum_{n=1}^{\infty}h(n)z^{-n}$，并在方程两边对 z^{-1} 求偏导数，得：

$$\frac{\partial}{\partial z^{-1}}\left(1-\sum_{k=1}^{p}a_k z^{-k}\right)^{-1}=\frac{\partial}{\partial z^{-1}}\sum_{n=1}^{\infty}\tilde{h}(n)z^{-n}$$

即

$$\frac{\sum_{k=1}^{p}ka_k z^{-n+1}}{1-\sum_{k=1}^{p}a_k z^{-k}}=\sum_{n=1}^{\infty}n\tilde{h}(n)z^{-n+1}$$

故

$$\left(1-\sum_{k=1}^{p}a_k z^{-k}\right)\sum_{n=1}^{\infty}n\tilde{h}(n)z^{-n+1}=\sum_{k=1}^{p}ka_k z^{-n+1}$$

整理得：

$$z^0\left[\tilde{h}(1)-a_1\right]+z^{-1}\left[2h(2)-h(1)a_1-2a_2\right]+\cdots$$
$$+z^{-p-1}\left[(p+2)\tilde{h}(p+2)-(p+1)\tilde{h}(p+1)a_1-\cdots-12\tilde{h}(2)a_p\right]\cdots=0$$

令每一项均等于 0，可以得到 $\tilde{h}(n)$ 和 a_k 之间的地推关系，从而由 a_k 求出 $\tilde{h}(n)$：

$$\begin{cases} \tilde{h}(0)=0 & (n\le 0) \\ \tilde{h}(1)=a_1 & \\ \tilde{h}(n)=a_n+\sum_{k=1}^{n-1}(1-k/n)a_k\tilde{h}(n-k) & (1\le n\le p) \\ \tilde{h}(n)=\sum_{k=1}^{p}(1-k/n)a_k\tilde{h}(n-k) & (n>p) \end{cases}$$

其中 a_k 为 LPC 系数，而 p 为其阶数。

理论上 LPCC 技术可以取任意高的阶数，这样能保留更多的有用信息，但是这样将导致运算量急剧上升，而且随着 LPCC 阶数的增加，$\tilde{h}(n)$ 越来越小，较大阶数的 LPCC 系数实际上并没有起到应有的作用，在一般情况下 LPCC 取 8～12 阶[12]。

在音乐风格分析识别过程中，有时需要对 LPCC 做一些后续处理，通常是求解 LPCC 一阶差分或者二阶差分，这样做可以更好地保留音乐的动态特征，提高音乐风格的识别效果。

6.2.2 Mel 频率倒谱系数

Mel 频率倒谱系数(MFCC)在音乐风格识别中是一个比较重要的参数，在许多实验中其性能优于其他参数。Mel 频率倒谱系数基于 Mel 频率，完整的 Mel 频率提取过程如图 3 所示。

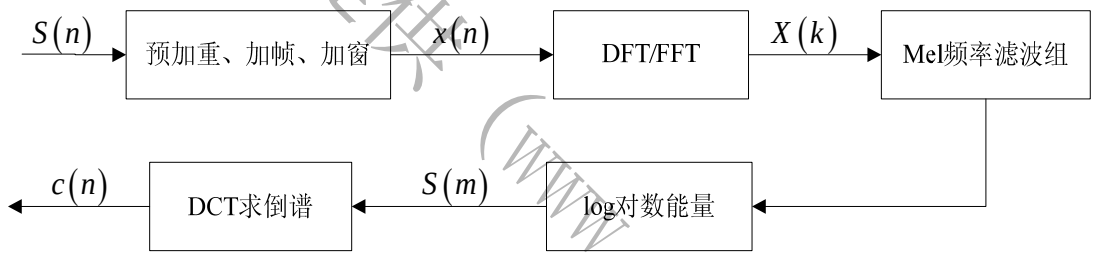

图 2 Mel 频率倒谱系数(MFCC)提取过程

MFCC 提取过程包括一下步骤：

Step 1 对原始音乐信号 $s(n)$ 进行预处理，包括预加重、分帧、加窗等，经过一些列预处理之后得到音乐帧的时域信号 $x(n)$；

Step 2 将分帧之后的时域音乐信号 $x(n)$ 后补若干 0，将帧长补为 2 的整次幂，一般取帧长 $N=512$。对分帧之后的时域音乐信号做离散傅立叶变换(DFT)，DFT 之后得到音乐信号的频域表达 $X(k)$：

$$X(k)=\sum_{n=0}^{N-1}x(n)e^{-j2n\pi k/2}\ (0\le n,k\le N-1)$$

由于在 DFT 之前已经进行了帧长补 0 处理，因此可以对原始音乐时域信号使用快速傅立叶变换(FFT)求取频谱表达；

Step 3 音乐信号进行 FFT 变换后得到频谱 $X(k)$，然后将频谱值通过 Mel 频率滤波组进行滤波，滤波之后的结果即为 Mel 频谱。得到 Mel 频谱之后需要对频谱做对数能量处理，即取其能量的对数值，可以得到音乐信号的对数频谱 $S(m)$。Mel 频率滤波器组包含若干个带通滤波器 $H_m(k),0\le m\le M$，M 个滤波器均为三角滤波器，$H_m(k)$ 的中心频率用 $f(m)$ 来表示。滤波器组的中心频率间隔与三角滤波器的序号有关，滤波器组

中序号靠前的相邻三角滤波器中心频率之差比序号靠后的相邻三角滤波器频率之差要大。Mel 频率滤波器表达为：

$$H_m(k)=\begin{cases}0 & (k<f(m-1)) \\ \dfrac{k-f(m-1)}{f(m)-f(m-1)} & (f(m-1)\le k\le f(m)) \\ \dfrac{f(m+1)-k}{f(m+1)-f(m)} & (f(m)\le k\le f(m+1)) \\ 0 & (k>f(m+1))\end{cases} \quad (0\le m\le M)$$

定义三角滤波器中心频率 $f(m)$ 为

$$f(m)=\left(\frac{N}{F_s}\right)B^{-1}\left(B\left(f_l+m\frac{B(f_h)-B(f_l)}{M+1}\right)\right)$$

其中 f_l、f_h 分别为当前三角滤波器频谱的最低频率值和最高频率值，N 为 FFT 序列的长度，即音乐信号补 0 后的帧长度。F_s 表示音乐信号的采样频率，而 B^{-1} 表达式为

$$B^{-1}(b)=700\left(e^{b/1125}-1\right)$$

求出 Mel 频率之后，对 Mel 频谱取对数，这样在利用 Mel 频率做谱误差估计时会有更好的鲁棒性，将音乐信号取对数之后得到的输出对数频率谱数据(传递函数)为：

$$S(m)=\ln\left(\sum_{k=0}^{N-1}|X(k)|^2\right)(0\le m\le M)$$

在求出音乐信号的对数频谱之后，还要对对数频谱 $S(m)$ 进行离散余弦变换，离散余弦变换可以将对数频谱变换到频谱域，经过离散余弦变换之后的输出值即为 Mel 频率倒谱系数 $c(n)$：

$$c(n)=\sum_{m=1}^{M-1}S(m)\cos\left(\frac{\pi n(1+1/2)}{M}\right)(0\le m\le M)$$

MFCC 特征提取过程的 MATLAB 程序代码见附录。

6.3 GMM 模型及其参数估计

6.3.1 GMM 模型描述

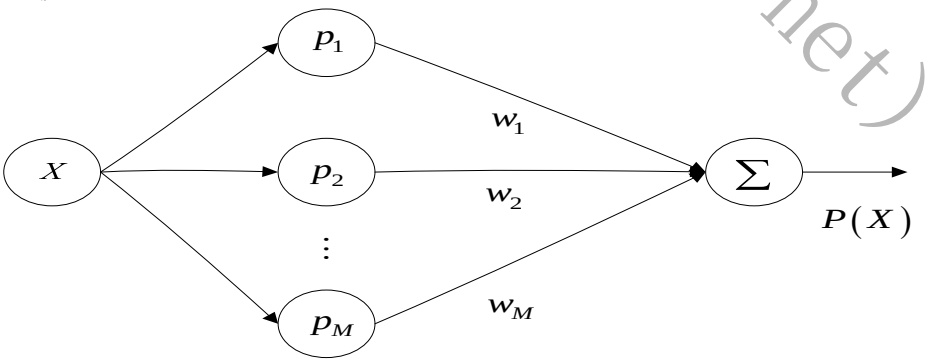

图 3 高斯混合模型示意图

高斯混合模型如图 2 所示，它由 M 个多维高斯分布加权叠加得到。一个 M 阶的混合高斯模型的概率密度函数是由 M 个单高斯概率密度函数加权和得到的，表示如下：

$$P(X|\lambda)=\sum_{i=1}^{i=M} w_i p_i(X)$$

其中 M 是混合模型的阶数， X 是一个 D 维随机向量， $w_i(i=1,2,\ldots,M)$ 是混合权重，且满足：

$$\sum_{i=1}^{M} w_i = 1$$

$p_i(X), i=1,2,\ldots,M$ 是子分布，每个子分布是 D 维的联合高斯概率分布，可表示如下：

$$p_i(X)=(2\pi)^{-\frac{D}{2}}\left|\sum_i\right|^{-\frac{1}{2}}\exp\left\{-\frac{1}{2}(X-\mu_i)^T\sum_i^{-1}(X-\mu_i)\right\}$$

其中 μ_i 是均值向量， $\sum_i$ 是协方差矩阵。

整个的高斯混合模型便可由参数均值向量，协方差矩阵和混合权重来描述。因此一个模型 λ 可以表示为如下一个三元组：

$$\lambda=\left\{w_i,\mu_i,\sum_i\right\},\quad i=1,2,\ldots,M$$

6.3.2 K-均值聚类的模型参数初始化

在对 GMM 进行训练之前，首先要对其初始化。常用的参数初始化方法有两种，一种是随机选择法，即从训练歌曲特征矢量集中随机选择 M 个特征矢量作为 GMM 的均值矢量 $\mu_i(i=1,2,\ldots,M)$ 的初始值。由于这种初始化方法没有用到特征矢量分布的先验信息，因此迭代所需次数较多。另一种是聚类选择法，这种参数初始化方法通常有 LBG 算法和 K-均值聚类算法，此参数初始化方法结合了特征矢量的分布函数由多个高斯分布函数的线性组合原理，故人为地将特征矢量划分为 M 个聚类，利用了样本先验分布知识，显然比前一种方法更好。

使用 K-均值聚类算法来确定初始化混合权重、均值矢量和协方差矩阵，具体步骤如下：

Step 1 任意选取 M 个矢量作为初始聚类的中心，假设初始选取的 M 个聚类中心为 $\left(c_1^{(1)},c_2^{(1)},\ldots,c_M^{(1)}\right)$；

Step 2 按最小距离准则对输入样本分类，若

$$\left\|x_k-c_i^{(m)}\right\|\le\left\|x_k-c_j^{(m)}\right\| \qquad \forall i\ne j; i,j=1,2,\ldots,M$$

则 x_k 属于第 i 类；

Step 3 计算各个新的聚类中心

$$c_i^{(m+1)}=\frac{1}{N}\sum_{x_k\in c_i^{(m)}} x_k \qquad i=1,2,\ldots,M$$

其中 N_i 表示第 i 类(记为 C_i)中的样本个数；

Step 4 若 $\left\|c_i^{(m+1)}-c_i^{(m)}\right\|\ge\delta$，则转到 *Step* 2 继续，否则转 *Step* 5；

Step 5 初始 GMM 参数的计算为：

$$w_i=N_i/T$$

$$\mu_i=\frac{1}{N_i}\sum_{x_k\in C_i}x_k,\sigma_{ik}^2\frac{1}{N_i}\sum_{x_k\in C_i}(x_{ik}-\mu_{ik})^2 \qquad k=0,1,\ldots,D-1$$

其中 T 为样本总数， D 为协方差矩阵的维数。

6.3.3 EM 算法的原理[9]

EM 算法是一种通用的方法，它能够最大似然地估计非完成数据集的概率分布模型参数。它是一种迭代算法，每一次迭代包括两个步骤：计算期望值和最大化期望值，所以成为 EM 算法。

假设 X 是观测到的样本集，服从某种概率分布 λ，称 X 为非完整样本集。这里假设一个完全样本集 $Z(X,Y)$ 存在，则有联合密度函数

$$p(Z|\lambda)=p(X,Y|\lambda)=p(Y|X,\lambda)\,p(X|\lambda)$$

对这个新的概率密度函数，可以定义一个新的似然函数

$$L(\lambda|Z)=L(\lambda|X,Y)=p(X,Y|\lambda)$$

上式中的 X 和 λ 是常量，假设 Y 是隐藏信息，是未知、随机的，服从概率分布 λ 的。

EM 算法首先要做的是在已知 X 和 λ 的情况下，找到完全样本集 Z 的对数似然函数 $\log p(X,Y|\lambda)$ 的数学期望，定义为

$$Q(\lambda,\lambda^{i-1})=E\left[\log p(X,Y|\lambda)\,\middle|\,X,\lambda^{(i-1)}\right]$$

这里，λ^{i-1} 是当前参数的估计，用于计算期望。λ 是新的参数值，有 λ^{i-1} 增加到 Q 得到，上式中，X 和 λ^{i-1} 是常量。

上面对数学期望的估计叫做 EM 算法的 E 步，注意到函数 $Q(\lambda,\lambda')$ 有两个参数，第一个参数 λ 最终被最大似然的优化，第二个参数用来计算数学期望。

EM 算法的第二步称为 M 步，用来最大化 E 步得到的数学期望。如下式

$$\lambda^{i}=\arg Q(\lambda,\lambda^{i-1})$$

以上两个步骤根据需要将不断重复，可以证明[10]，每一次重复都使似然值增大，并且算法能保证收敛于似然函数的局部最大点。

表 2 EM 算法步骤

The EM Algorithm
STEP 0 Pick a starting value θ^0.Now for $j=1,2,\ldots$, repeat steps 1 and 2 below： (The E-step) Calculate：
STEP 1 $J(\theta|\theta^j)=E_{\theta^j}\left(\log\dfrac{f(Y^n,Z^n;\theta)}{f(Y^n,Z^n;\theta^j)}\,\middle|\,Y^n=y^n\right)$ The expectation is over the missing data Z^n treating θ^j and the observed data Y^n as Fixed
STEP 2 Find θ^{j+1} to maximize $J(\theta|\theta^j)$

6.3.4 用 EM 算法估计 GMM 的参数

GMM 的概率密度函数为

$$P(X|\lambda)=\sum_{i=1}^{M}w_i\,p_i(X)$$

其中 $\lambda=(w_1,w_2,\ldots,w_M;\lambda_1,\lambda_2,\ldots,\lambda_M)$，且有 $\sum_{i=1}^{M}w_i=1$，并且每个高斯概率密度模型的参数为 λ_i。$P(X|\lambda)$ 由 M 个高斯混合模型按照混合系数 w_i 混合得到。

非完整样本集 X 的对数似然函数表示为

$$\log\left(L\left(\lambda|X\right)\right)=\log\prod_{i=1}^{T}p\left(x_i|\lambda\right)=\sum_{i=1}^{t}\log\left(\sum_{j=1}^{M}w_j p_j\left(x_i|\lambda_j\right)\right)$$

其中，T 为 X 中样本的个数，M 为混合模型中的高斯模型混合数。在上式中包含对数的和，该似然函数难以求极值。然而，如果考虑 X 是非完整的情况，假设存在未观测到的数据项 $Y\approx\{y_t\}_{t=1}^{T}$，$Y$ 的值表明每个 X 集合中的样本是由混合模型的哪个模型产生的，这时，似然函数的表达式将简化。假设 $y_t\in 1,2,\ldots,M$ 。如果第 t 个样本由混合模型中的第 k 个模型产生，则取 $y_t=k$ 。

引入 Y 后，完整样本集的似然函数可以写为

$$\log\left(L\left(\lambda|X,Y\right)\right)=\log\left(P\left(X,Y|\lambda\right)\right)=\sum_{i=1}^{T}\log\left(w_{y_t}p_{y_t}\left(x_t|\lambda_{y_t}\right)\right)$$

对于高斯混合模型，根据 EM 算法的 E 步计算上式似然函数的数学期望

$$Q\left(\lambda,\lambda^E\right)=E\left[\log\left(L\left(\theta|X,Y\right)\right)\right]$$

展开得

$$Q\left(\lambda,\lambda^E\right)=\sum_{i=1}^{M}\sum_{t=1}^{T}\log\left(w_i\right)p\left(l|x_t,\lambda^E\right)+\sum_{i=1}^{M}\sum_{t=1}^{T}\log\left(p_l\left(x_t|\lambda_l\right)\right)p\left(l|x_t,\lambda^E\right)$$

其中，$l=1,2,\ldots,M$，$p\left(l|x_t,\lambda^E\right)$ 表示在已知样本集和 X 和模型参数 λ^E 情况下，某样本 x_t 属于混合模型中第 l 个模型的后验概率。由下式计算，k 代表第 k 次循环。

$$p^{(k)}\left(l|x_t,\lambda^E\right)=\frac{w_t^{(k)}p_l\left(x_t|\lambda_l^{(k)}\right)}{\sum_{j=1}^{M}C_j^{(k)}p_j\left(x_t|\lambda_j^{(k)}\right)}$$

在 EM 算法的 M 步，需要找到在使似然函数最大化的模型参数。由上述似然函数的展开式可以看出，该式有两项相加得到，前一项只跟 C_l 有关，后一项只跟 λ_l 有关。最大化前一项得到新的混合系数估计，最大化后一项，可得到新的均值和协方差矩阵估计。

特征矢量落入隐状态 i 的概率为

$$p\left(i|x_t,\lambda\right)=\frac{w_t p_t\left(x_t\right)}{\sum_{k=1}^{M}w_k p_k\left(x_l\right)}$$

由此，可以得到下面三个 GMM 参数重估公式。其中，加权系数重估公式为

$$w_i^{(k+1)}=\frac{1}{T}\sum_{t=1}^{T}p^{(k)}\left(i|x_t,\lambda\right)$$

均值重估公式为

$$\mu_i^{(k+1)}=\frac{\sum_{t=1}^{T}p^{(k)}\left(i|x_t,\lambda\right)x_t}{\sum_{t=1}^{T}p^{(k)}\left(i|x_t,\lambda\right)}$$

方差重估公式为

$$\sigma_{ik}^2=\frac{\sum_{t=1}^{T}p^{(k)}\left(i|x_t,\lambda\right)\left(x_{ik}-\mu_{ik}^{(k+1)}\right)^2}{\sum_{t=1}^{T}p^{(k)}\left(i|x_t,\lambda\right)}$$

可以看出，上述公式同时执行了求期望值和最大化。对上述公式的迭代重复，就是对 EM 算法中 E 步、M 步的重复迭代。当找到似然函数的极大值时停止迭代。对最大似然的估计训练样本的高斯混合模型参数来说，EM 算法是一个极佳的算法，算法流程图如下，利用 EM 算法估计 GMM 模型参数的 MATLAB 程序见附录。

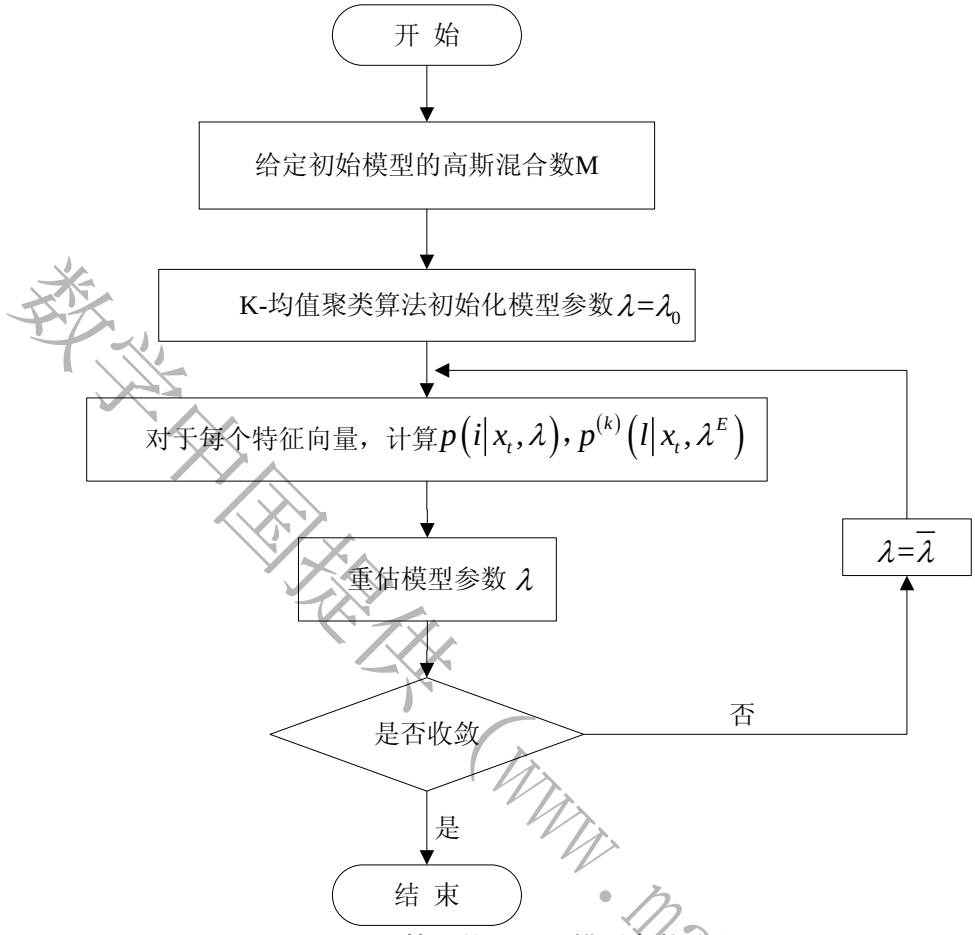

图 4 EM 算法估计 GMM 模型参数流程图

6.4 音乐风格向量以及雷达图表示

6.4.1 音乐风格向量的表示

对流行音乐风格进行分类的时候，必然会研究那些风格之间容易混淆，并且对于歌曲风格，一般我们所说的歌曲都是单一风格的歌曲，但是由于歌曲风格的模糊化定义，类别与类别之间没有明确的界限，一首歌曲可能会同时具有几种风格。为了对这样同时兼有几种风格的歌曲进行研究，我们提出一种基于高斯混合模型的研究类别之间混识程度和双重风格歌曲的方法，根据 GMM 的似然比建立音乐风格向量，根据音乐风格向量绘制雷达图，使用雷达图提供可视化分析。

实验中，我们对六种单一风格的歌曲进行了分类，假如已经训练出六种风格音乐对应的模型。对于待识别歌曲，我们经计算它与每个风格模型的似然比 $r_i\left(i=1,2,\ldots,6\right)$，计算方法如下：

$$r=\sum_{j=1}^{N}\log\sum_{k=1}^{M}w_k p_k\left(X_j\right)$$

其中，X_j 是特征向量，N 是该首用于识别歌曲所包含的特征向量总数，$p_k\left(X_j\right)$ 为单高斯概率密度函数。

然后，我们利用极差标准化的方法对似然比进行归一化：

$$r = \frac{r_i - \min(r)}{\max(r) - \min(r)}$$

其中，$\max(r)$，$\min(r)$分别为似然比$r_i(i=1,2,\ldots,6)$中的最大、最小值，求得的结果记为$r_i(cu, genre_i)\ i=1,2,\ldots,6$，代表当前待识别歌曲$cu$与音乐风格类别$genre_i$之间的相似程度，这样就形成了一个歌曲风格向量:

$$Vec = (r_1(cu, Pop), r_2(cu, Cou), r_3(cu, Jaz), r_4(cu, Roc), r_5(cu, RB), r_6(cu, NA))$$

6.4.2 音乐风格向量的雷达图表示

我们可以把音乐风格向量中六个归一化的似然比值表征在雷达图中，雷达图每条圆心到圆周的连线代表一种风格，圆心代表 0，圆周代表 1。这样表示之后，可以求出某一种风格所有待识别歌曲平均似然比的雷达图，有助于我们清楚地看出那些音乐风格之间容易误判，那些风格之间差异明显，为我们对风格之间的关系提供可视化分析。

我们还可以利用音乐风格向量与雷达图对双重风格歌曲进行研究。对于双重风格歌曲，给出两个候选结果，即音乐风格向量中最大值和次大值对应的风格类型。对于一首双重风格歌曲，假如风格类型为$genre_1$和$genre_2$，如果在音乐风格向量中满足：

$$\arg(\max(r_i)) = genre_1 \vee genre_2$$

$$\arg(sec\text{-}\max(r_i)) = genre_1 \vee genre_2$$

即音乐风格向量中最大和次大值对应的风格正好是$genre_1$和$genre_2$或者$genre_2$和$genre_1$，则我们人为识别正确，否则，都认为识别错误。图 5 是一首新世纪摇滚音乐的雷达图表示，其音乐风格向量为$Vec = (0, 0.42395, 0.70346, 1, 0.53893, 0.95391)$。

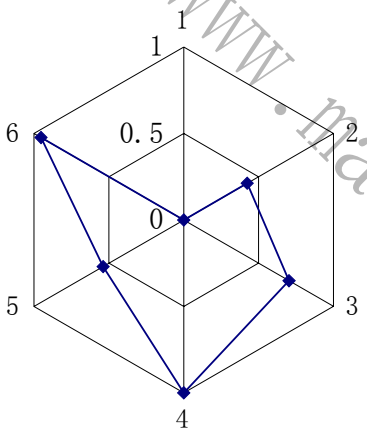

图 5 一首新世纪摇滚歌曲的雷达图表示

从图中可以看出，摇滚得分最高，新世纪得分次之，兼有两种风格，识别正确。

6.5 分类结果评价

为了对音乐风格分类结果进行客观评价，参考目前国际上通用的信息检索测试标准——美国 TREC 会议(文本检索会议，Text Retrieval Conference)标准。该标准是 20 实际 90 年代美国国防部高级研究计划署(DARPA)出资推行的一个文本处理计划。根据音乐风格分类的特点，主要选择了如下指标，

(1) 平均准确率(Mean Average Precision，即 MAP)：单个主题的 MAP 是指检索出每篇相关文档的准确率的平均值。主题集合的 MAP 是每个主题的 MAP 的平均值，MAP 反映了系统对于全部相关文档的检索性能。

(2) R-准确率(R Precision)：单个主题的 R-准确率是指检索出 R 篇文档时的准确率。其中 R 是测试集中与主题相关的文档数目。主题集合的 R-准确率是每个主题的 R-准确率的平均值。具体来说，如表 3 所示，对于指定的主题，所有文件集合可以分为四类文件的数目：R、S、M、N，R 是检索到的与主题相关的 R 个文件，S 则为检索到的与主题无关的 S 个文件，此外还有与主题相关的 M 个文件和不相关的 N 个文件未被检索到。此时，R 准确率 $\hat{R}$ 定义为

$$\hat{R}=\frac{R}{R+S}$$

表 3 R-准确率中各类文件的数目

	相关文件	非相关文件
获取值	R	S
非获取值	M	N

6.6 实验结果的分析

运用实际数据，对流行音乐风格进行模糊划分，得到表 4。实验结果表明，模糊分类方法具有较好的歌曲风格分类能力，平均准确率达到 85%，R-准确率达到 90%。

表 4 模糊分类最终划分结果

类别 / 音乐	Pop	Country	Jazz	Rock	R&B	New Age
NO.1	1	0	0	1	0	0
NO.2	0	1	0	0	0	0
NO.3	0	0	0	0	0	1
……	……	……	……	……	……	……

基于 GMM 模型的音乐风格自动分类结果如下：单一音乐风格共 310 首，双重音乐风格共 189 首，原音乐数据库单一音乐风格共 315 首，双重音乐风格共 193 首，因此单一音乐风格识别正确率为 98.413%，双重音乐风格识别正确率为 97.927%。由此可得高斯混合模型的识别正确率和识别效率都明显好于模糊分类模型。

由 GMM 模型得到音乐风格分类的识别混淆矩阵，

$$CM=\begin{bmatrix}112 & 1 & 2 & 0 & 3 & 16\\ 1 & 84 & 1 & 2 & 9 & 8\\ 0 & 0 & 21 & 0 & 0 & 1\\ 3 & 2 & 0 & 32 & 5 & 0\\ 2 & 10 & 1 & 9 & 68 & 10\\ 3 & 9 & 4 & 0 & 5 & 84\end{bmatrix}$$

则总体音乐风格分类有效率为 $\gamma=\frac{trace(CM)}{508}=0.7894$，每一类分类有效率为 0.9492,0.7925,0.8077,0.7442,0.7556,0.7059。有结果分析，利用 GMM 模型进行音乐风格判别分析与分类研究的有效率都比较高，运算和收敛速度都达到预期要求，移植性和健壮性也较好，这是一个算法优劣的主要评价参数。

7 模型评价与改进

7.1 模型评价

针对实际中提供的单一音乐风格、双重音乐风格音频数据，模糊分类模型对风格类型具有较好的分类能力，但由于该方法不具有普适性。因此，我们在对构成音乐主要特征参数研究的基础上，选择了 LPC 倒谱系数和 Mel 频率倒谱系数，它们具有较好地反映音乐风格类型的特性。然后建立了 GMM 模型，并采用 K-均值聚类算法和 EM 算法进行模型参数的初始化和参数估计，在实验验证过程中，利用 MATLAB 和 Python 编程实现。

考虑到一首歌曲可能是几种风格组合而成，所以我们进行了基于音乐风格向量的研究，并用雷达图表示音乐风格向量，呈现出一种可视化分析，实验结果表明，该模型具有比较好的识别效率与准确率，且得到了较为满意的结果，此外，这种方法具有可操作性强，易于实现，可移植性强等特点。

7.2 改进技术路线

大多数对音乐风格分类模型进行改进的研究都使用平均谱(如本文的 MFCC)包络表现音乐的频谱特性。这种特性子频段的频谱平均化，而且反映平均频谱的特性，但是它不能表现每个子频段频谱的相对特性，这一点在区别不同类型的音乐时也许更有参考性。因此，本文模型的改进方向是，多种语言音乐风格类型的分析与研究，并尝试利用基于八音阶频谱对比度特征表现音乐的频谱相对特征[12]，即分别考虑两个子频段频谱的峰值和谷值的强度，以便使它可以表现频谱的相对特性，然后粗略地反映谐波和非谐波的分布。新类别检测，即分类器能够识别一种新的音乐风格，可以自动地将这首音乐标记为新的音乐风格；多类标记，即将一首音乐标记为多种风格。随着音乐的发展与融合，一首音乐可能融合多种风格，对音乐进行多标签分类，可以有效地解决音乐风格模糊、不明确问题，同时也更加符合实际。对与多类分类问题可以采用纠错编码 SVMs[13,14]进行音乐风格类型分类，它是基于纠错编码理论，并通过汉明距离测度准则进行判断，从而最终得到识别结果。

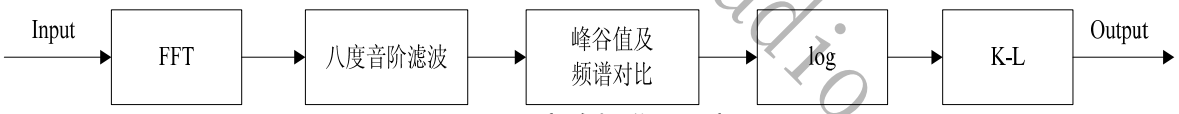

图 6 八音阶频谱对比度

8 模型的推广与应用

本文所建立的数学模型可运用于多类音频分类与识别，音频数字信号处理，音频信息检索研究，以及基于音乐风格类型的歌手音质的识别与评价等场合。对于流行音乐的风格的分类方法，可以给网络电台的推荐功能、流行音乐的市场分析和基于流行音乐的大众审美研究等提供理论依据。

可以利用本文的模型进行相似音乐的检索，歌唱者可以根据自己的声音和风格，利用检索系统检索与自己相似的歌曲，同时，用户可以从网络电台歌曲库中自动选择自己喜爱的歌曲，满足个性化音乐推荐和服务。最后，对算法进行优化，增加多线程和并行计算技术以及将数据库进行预分类，还可以较大幅度地提高系统的运行速度，不仅可以用于非纯音乐的检索，还可以满足基于网络的大音乐库检索和推荐功能，为用户自动选择喜爱的歌曲和自动推荐歌曲提供可能。

参考文献

[1]Scruton R.The Aesthetics of Music[M].Oxford:Oxford University Press,1997:15.
[2]张飞弦，张伟，谢湘. 基于 GMM 的流行音乐情感识别研究[M]. NCMMSC，2009.
[3]MPEG-7.Information Technology-Multimedia Content Description Interface-Part 4:Audio[S].ISO/IEC JTC 1/SC29,ISO/IEC FDIS 15938-4:2002,2002.
[4]K West and S Cox.Finding an optimal segmentation for audio genre classification,6th International Symposium Music Information Retrieval[C].London,UK,2005.
[5]G Zoia,R Zhou and D Mlynek.A multi-timbre chord/harmony analyzer based on signal processing and neural networks,IEEE International Workshop Multimedia Signal Processing[C].Siena,Italy,2004.
[6]A Klapuri.Multiple fundamental frequency estimation based of harmonicity and spectral smoothness[J].IEEE Transaction Speech Audio Processing,2003,11(6):804-816.
[7]F Gouyon and Dixon.A review of automatic rhythm description system[J].Computer Music Journal,2005,29(1):34-54.
[8]A Berenzweig,D Ellis and S Lawrence.Using voice segments to improve artist classification of music,AES 25th Internal Conference Virtual,Synthetic Entertainment Audio[C].London,England,2004.
[9]Cardoso J F.Infomax and maximum likelihood for source separation.IEEE Letters on Signal Processing.1997,4(5):112-114.
[10]L.A.Wasserman.All of Statistics:A Concise Course in Statistical Inference.[M].Springer-Verlag New Youk Inc.,2004,143-145.
[11]蔡莲红，黄德智，蔡锐. 现代语音技术基础与应用[M]. 北京：清华大学出版社，2003.
[12]项慨. 基于频谱对比度特征的音乐风格分类[J]. Microcomputer Application,21(3),2005.
[13]夏建涛，何明一. 支持向量机与纠错编码相结合的多类分类算法[J].JOURNAL OF NORTHWESTERN POLYTECHNICAL UNIVERSITY,21(4),2003.
[14]刘志刚，李德仁等. 支持向量机在多类分类问题中的推广[J].COMPUTER ENGINEERING AND APPLICATIONS,40(7),2004.
[15] Python 官方网站：http://www.python.org/
[16] Chun,W.J.，宋吉广(译).Python 核心编程[M].北京：人民邮电出版社，2008.7.

附 录

1、用 EM 算法估计 GMM 模型参数。

```
%%=========================================================================%%
function [label, model, llh] = emgm(X, init)
% EM algorithm for Gaussian mixture model
%% initialization
fprintf('EM for Gaussian mixture: running ... ');
R = initialization(X,init);

tol = 1e-6;
maxiter = 500;
llh = -inf(1,maxiter);
converged = false;
t = 1;
while ~converged && t < maxiter
    t = t+1;
    model = maximization(X,R);
    [R, llh(t)] = expectation(X,model);
    converged = llh(t)-llh(t-1) < tol*abs(llh(t));
end
[~,label(1,:)] = max(R,[],2);
llh = llh(2:t);
if converged
    fprintf('converged in %d steps.\n',t);
else
    fprintf('not converged in %d steps.\n',maxiter);
end
%%=========================================================================%%
%%=========================================================================%%
function R = initialization(X, init)
[d,n] = size(X);
if isstruct(init)  % initialize with model
    R  = expectation(X,init);
elseif length(init) == 1  % random initialization
    k = init;
    idx = randsample(n,k);
    m = X(:,idx);
    [~,label] = max(bsxfun(@minus,m'*X,sum(m.^2,1)'/2));
    while k ~= unique(label)
        idx = randsample(n,k);
        m = X(:,idx);
        [~,label] = max(bsxfun(@minus,m'*X,sum(m.^2,1)'/2));
    end
    R = full(sparse(1:n,label,1,n,k,n));
elseif size(init,1) == 1 && size(init,2) == n  % initialize with labels
    label = init;
    k = max(label);
    R = full(sparse(1:n,label,1,n,k,n));
elseif size(init,1) == d && size(init,2) > 1  %initialize with only centers
    k = size(init,2);
    m = init;
    [~,label] = max(bsxfun(@minus,m'*X,sum(m.^2,1)'/2));
    R = full(sparse(1:n,label,1,n,k,n));
else
    error('ERROR: init is not valid.');
end
```

```
%%=====================================================================%%
%%=====================================================================%%
function [R, llh] = expectation(X, model)
mu = model.mu;
Sigma = model.Sigma;
w = model.weight;

n = size(X,2);
k = size(mu,2);
R = zeros(n,k);

for i = 1:k
    R(:,i) = loggausspdf(X,mu(:,i),Sigma(:,:,i));
end
R = bsxfun(@plus,R,log(w));
T = logsumexp(R,2);
llh = sum(T)/n; % loglikelihood
R = bsxfun(@minus,R,T);
R = exp(R);
%%=====================================================================%%
2、MFCC 参数提取过程
%%=====================================================================%%
function model = maximization(X, R)
[d,n] = size(X);
k = size(R,2);
sigma0 = eye(d)*(1e-6); % regularization factor for covariance

s = sum(R,1);
w = s/n;
mu = bsxfun(@rdivide, X*R, s);
Sigma = zeros(d,d,k);
for i = 1:k
    Xo = bsxfun(@minus,X,mu(:,i));
    Xo = bsxfun(@times,Xo,sqrt(R(:,i)'));
    Sigma(:,:,i) = (Xo*Xo'+sigma0)/s(i);
end

model.mu = mu;
model.Sigma = Sigma;
model.weight = w;
%%=====================================================================%%
%%=====================================================================%%
function ccc = mfcc(x)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%function ccc=mfcc(x);
%对输入的语音序列 x 进行 MFCC 参数的提取，返回 MFCC 参数和一阶
%差分 MFCC 参数，Mel 滤波器的阶数为 24
%fft 变换的长度为 256，采样频率为 8000Hz，对 x 256 点分为一帧
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
bank=melbankm(24,256,8000,0,0.5,'m');
% 归一化 mel 滤波器组系数
bank=full(bank);
bank=bank/max(bank(:));
% DCT 系数,12*24
for k=1:12
  n=0:23;
  dctcoef(k,:)=cos((2*n+1)*k*pi/(2*24));
```

```
end
% 归一化倒谱提升窗口
w = 1 + 6 * sin(pi * [1:12] ./ 12);
w = w/max(w);
% 预加重滤波器
xx=double(x);
xx=filter([1 -0.9375],1,xx);
% 语音信号分帧
xx=enframe(xx,256,80);
% 计算每帧的 MFCC 参数
for i=1:size(xx,1)%size(xx,1)返回 xx 的维数
  y = xx(i,:);
  s = y'.* hamming(256);%乘窗
  t = abs(fft(s));
  t = t.^2;%计算能量
  c1=dctcoef * log(bank * t(1:129));% dctcoef 为 DCT 系数，bank 归一化 mel 滤波
器组系数
  c2 = c1.*w'; % w 为归一化倒谱提升窗口
  m(i,:)=c2';
end
%差分系数
dtm = zeros(size(m));
for i=3:size(m,1)-2
  dtm(i,:) = -2*m(i-2,:) - m(i-1,:) + m(i+1,:) + 2*m(i+2,:);
end
dtm = dtm / 3;
%合并 mfcc 参数和一阶差分 mfcc 参数
ccc = [m dtm];
%去除首尾两帧，因为这两帧的一阶差分参数为 0
ccc = ccc(3:size(m,1)-2,:);
%%=====================================================================%%
3、特征提取过程
%%=====================================================================%%
%%Feature_Extract.m
function [FileName,mean_value,variance]=Feature_Extract(FileName)
% [mean_value,variance]=Feature_Extract(FileName)
% FileName 是需要分析的波形文件的路径
% mean_value 是计算得出的特征值的平均值
% variance 是计算得出的特征值的方差
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
[F,Fs,NBITS] = wavread(FileName,20*44100);   % 读入波形数据
time = 20;                                   % 采样时间 60 秒
T = 1:time*Fs;                                 % 采样时间轴
Wave = F(T);                                   % 采样段数据
Wave = Wave/max(abs(Wave)); % 数据归一化处理
WLen = length(T); % 统计采样数据点的数量

winlen = 2^nextpow2(Fs*20/1000); % 窗长为 10ms~30ms,这里取 20ms
dupwin = 2^nextpow2(Fs*5/1000); % 为保持连续性，窗口有重叠，重叠 5ms
stepwin = winlen-dupwin; % 窗每次移动 stepwin 个采样点
E = zeros(WLen-stepwin,1); %` 初始化能量矩阵
for i = 1:stepwin:WLen-stepwin % 计算帧能量 FE
    xm = Wave(i:i+stepwin);
    E(i) = sum(xm.*xm);
```

```
end
E0 = [E zeros(length(E),1)]; % 为记录帧的位置准备，E0 第一维是 E,
                             % 第二维是相应的位置
E0 = setxor(E0(:,1),0); % 删除末尾零记录
j=1;
for i = 1:length(E) % 记录帧的位置
    if E(i)>0
        E0(j,1)=E(i);
        E0(j,2)=i;
        j = j+1;
    end
end
Emin = min(E0(:,1)); % 计算帧能量的最小值
Emax = max(E0(:,1)); % 计算帧能量的最大值
Emean = mean(E0(:,1)); % 计算帧能量的平均值
lamda = 0.5; % 设定静音阈值
Ttfe = Emin + lamda * (Emean - Emin);

for i = 1:length(E0(:,1)) % 屏蔽 E0 中对饮帧能量小于静音阈值的值
    if E0(i,1) < Ttfe
        E0(i,1) = 0;
    end
end
% 寻找特征片段
FER = ones(length(E0(:,1)),2); % 初始化帧能量比矩阵
for i = 1:(length(E0(:,1))-1) % 计算帧能量比
    if(and(E0(i,1),E0(i+1,1))) % 若当前帧与后一帧都不为零
        FERa = E0(i+1,1)/E0(i,1);
        FERb = E0(i,1)/E0(i+1,1);
        FER(i,1)=max(FERa,FERb);
        FER(i,2)=E0(i,2);
    end
end
level = mean(FER(:,1)); % 设定高潮端点阈值
result0 = zeros(length(FER(:,2)),1); % 初始化结果矩阵
j = 2;
if FER(1,1)-level >0
    result(1) = FER(1,2);
end            %过滤出高潮端点
for i = 2:length(FER(:,2))-1
    if FER(i,2)-level >0
        if FER(i-1,2)-level <0
            result0(j) = FER(i,2);
            j = j+1;
        end
    end
end
result0 = setxor(result0,0); % 删除多余的零元素
result = zeros(length(result0)-1,1);
for i = 1:length(result0)-1
    result(i) = result0(i+1)-result(i);
end
characion = zeros(size(result));
for i = 1:length(result)-1
    characion(i) = result(i+1)-result(i);
end
```

```
result = charaction;
FileName;                %输出特征向量
u = mean(result);
d = var(result);
disp([FileName])
disp([ '均值：' num2str(u) '方差：' num2str(d)]);
mean_value = u; % 函数返回特征向量
variance = d;
end
%%=====================================================================%%
%%=====================================================================%%
```

4、计算实验过程中的 Python 部分关键代码（详细代码见附件）

```
#########################################################################
#!/usr/bin/python

import distances
import fnmatch
import os
import ranking
import sys

if len(sys.argv) != 2:
    print 'usage: ', sys.argv[0], ' querydirectory'
    sys.exit(1)

querydir = sys.argv[1]


print 'Computing ranking...'

ranking.compute(querydir)
#########################################################################
#########################################################################
#!/usr/bin/python

import glob
import os
import sys

if len(sys.argv) != 3:
    print 'usage: ', sys.argv[0], ' modelfile querydirectory'
    sys.exit(1)

configfile = open('config', 'r')
config = configfile.readline().strip()
audiopath = configfile.readline().strip()
featuresext = configfile.readline().strip()

model = os.path.join('..', sys.argv[1])
querydir = os.path.join('..', sys.argv[2])

# specific processing

extractor = ''
evaluator = ''

if config == 'shell':
    extractor = './DescriptorExtractor.sh'
```

```
    evaluator = './EvaluateModel.sh'

elif config == 'python':
    extractor = './DescriptorExtractor.py'
    evaluator = './EvaluateModel.py'

os.chdir('bin')

# compute all queries first (extract descriptors + evaluate model)
for f in glob.glob1(querydir, '*.query'):
    queryfilename = os.path.join(querydir, f)
    queryfile = open(queryfilename, 'r')
    wavfile = os.path.join(audiopath, queryfile.readline().strip())
    queryfile.close()

    # compute descriptors && evaluate model
    if config == 'matlab':
        os.system('echo  "DescriptorExtractor(\'' + f + '\',\'' + f +
featuresext + '\')" | matlab -nosplash -nodesktop')
        os.system('echo "EvaluateModel(\'' + f + featuresext + '\',\'' + model
+ '\',\'' +
                  queryfilename + '\')" | matlab -nosplash -nodesktop')
    else:
        os.system(extractor + ' ' + wavfile + ' ' + wavfile + featuresext)
        os.system(evaluator + ' ' + model + ' ' + queryfilename + '.result
' + wavfile + featuresext)


# check how many results are correct
queries, good = 0, 0
names, goodl, totall = [], [], []

total = len(glob.glob1(querydir, '*.query'))
current = 0
for f in glob.glob1(querydir, '*.query'):
    current += 1
    print '\r[' + str(current) + '/' + str(total) + ']',
    sys.stdout.flush()
    queryfilename = os.path.join(querydir, f)
    queryfile = open(queryfilename, 'r')
    queryfile.readline() # skip song name
    expected = queryfile.readline().strip()
    # update data structure
    try:
        queries += 1
        idx = names.index(expected)
        totall[idx] += 1
    except:
        names.append(expected)
        goodl.append(0)
        totall.append(1)

    queryfile.close()
    try:
        resultfile = open(queryfilename + '.result', 'r')
        result = resultfile.readline().strip()
        if result == expected:
            idx = names.index(expected)
```

```
            goodl[idx] += 1
            good += 1
    except:
        pass

normalized_score = 0.0

for i in range(len(totall)):
    normalized_score += float(goodl[i]) / (totall[i]*len(totall))

print
print good, 'good identifications out of', queries, 'queries'
print 'averaging', good * 100 / queries, '% correct answers'
print 'result normalized with respect to the probability of each class:',
int(normalized_score * 100), '% correct answers'
################################################################################
################################################################################
```