

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：3052

参赛队员（签名）：

队员 1： 向宇

队员 2： 高星

队员 3： 匡其浪

参赛队教练员（签名）：

参赛队伍组别： 本科组

参赛队号 # 3052

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）：

3052

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 # 3052

2013 年第六届“认证杯”数学中国
数学建模网络挑战赛

题 目 流行音乐时代识别

关 键 词 时代识别 $H-K$ 算法 相关系数加权 Bayes 分类器 支持向量机分类器
音乐指纹库

摘 要：

对于流行音乐的时代识别对音乐历史研究、网站电台音乐分类有现实意义。以下为步为本文解决流行音乐时代识别的步骤：

第一步，音乐时代的分类。查阅流行音乐历史相关专业书籍后，本文确定了旧上海、抗战救国、建国初期、文革时代、八九十年代、当代等 6 个流行音乐时代。

第二步，音乐特征的选取和处理。基于 $H-K$ 算法音乐主旋律特征量进行数据预处理。在 Visual Studio IDE 下用 C# 从 MIDI 文件中提取左右声道平衡度、主音量、平均音长、平均力度和音程比等 6 组基本数据，因 MIDI 文件有不同的音轨，则有好几组基本数据。我们利用 $H-K$ 算法提取出主旋律的那组基本数据，从而构建了音乐主旋律特征六维向量。

第三步，利用第第二步中求得数据，再建立判别分析模型，求解时代识别问题。

模型一，基于相关系数加权 Bayes 分类器。用模型一的音乐主旋律特征六维向量作为数据，对六个音乐特征属性加权处理，并结合本文给出 Bayes 分类算法进行时代判别。流行音乐的时代识别效果优秀，模型求解结果为对于旧上海、抗战救国、文革时代的时代效果最为明显，正确识别率在 90% 以上。

在模型二，基于支持向量机的分类器。用模型一的音乐主旋律特征六维向量作为数据，用少数样本的二次规划问题可以有解析解的形式求出六维分类超平面，最后得出的分类结果相对模型二更加明显和优秀。除了当代流行音乐外其他年代的正确识别达到 90% 以上。

在模型三，基于内容的音乐检索的方式判别年代。通过提取音乐文件的语谱图，选取其中的若干个特征位置作为音乐文件的特征点信息，形成音乐文件的指纹信息，用来标识音乐的特征，将全量的音乐指纹生成音乐指纹库，提供查询。根据这一模型和算法，论文提出了一种提取音乐指纹的算法，保证音乐指纹的唯一性。得出的分类较为理想。

最后向查阅相关资料后本文再结合历史政治环境、社会发展学以及社会心理学给出介绍各个时期流行音乐“成功要素”的变化发情况。

参赛队号： 3052

所选题目： B 题

参赛密码 _____
(由组委会填写)

Abstract

In the era of recognition of popular music has practical significance for music history research, web radio music classification. Following the step for the pop music era recognition steps:

The first step, classification of music era. Review the history of popular music related professional books, the old Shanghai, the war, the founding of the early, the cultural revolution era, eighty or ninety's, 6 times of contemporary popular music.

The second step, and select music features. Data preprocessing of H - K algorithm based on music melody feature. In the Visual Studio IDE using C# from the MIDI file extraction balance degree, master volume, average length, average intensity and interval than 6 groups of basic data, because the MIDI file has a different track, there are several groups of basic data. We use the H-K algorithm to extract the basic data of the main melody, the melody features of six dimensional vector.

The third step, the obtained data in the second step, and then the discriminant analysis model was established to solve the identification problem, age.

In model one, correlation coefficient weighted Bayes classifier based on. With music melody feature a model of six dimensional vector as the data, for the six musical attributes weighted is presented in this paper, combined with Bayes classification algorithm for age discrimination. The era of popular music good recognition effect, model results for the old Shanghai, the war, the cultural revolution era was the most obvious, the correct recognition rate is more than 90%.

In model two, a support vector machine classifier based on. With music melody feature a model of six dimensional vector as the data, using two planning problem a few samples can be analytically obtained six dimensional classification hyperplane, the classification results and finally comes to the relative model two is more obvious and outstanding. In addition to correctly identify the other's contemporary pop music reached more than 90%.

In model three, discrimination in the content-based music retrieval way. By extracting the music files spectrogram, select one of the plurality of feature location as the feature points of the music file information, form the fingerprint information of music files, used to mark the music, the music fingerprint generation music fingerprint database full volume, provides the query. According to this model and algorithm, this paper puts forward a kind of musical fingerprint extraction algorithm, to ensure the uniqueness of music fingerprint. The classification of ideal.

Finally to access relevant information in this paper, combined with the historical and political environment, social development and social psychology introduces changes of pop music "success factors" in each period of the.

目录

一、问题重述	1
二、问题分析	1
三、模型假设	1
四、符号说明	1
五、模型建立与求解	4
5.1 流行音乐年代的划分	4
5.2 音乐特征的数据预处理	4
5.3 模型二 基于相关系数的加权 <i>Bayes</i> 分类器	7
5.4 模型三:支持向量机分类器	11
5.5 模型四:基于多媒体内容的音乐检索的识别年代	14
六、模型评价	24
七、模型改进	24
八、小短文	25
九、参考文献	27
十、附录	28
附录 1	28
附录 2	28
附录 3	31

一、问题重述

近几十年来，流行音乐的风格经历了相当复杂的演变过程。每个时代都有其代表性的人物和风格，歌曲的成功要素也随时间不断改变。本文中，我们建立合理的数学模型，对曾经走红过的流行歌曲，通过其原唱录音或曲谱来推断其所处时代。

二、问题分析

第一步，音乐时代的分类

为了推断出某一首流行音乐所处年代，我们首先要划分流行音乐在历史上有哪些年代。本文就中国大陆流行音乐发展的情况，在查阅了相关资料后，根据时间和特点确定了旧上海、抗战救国、建国初期、文革时代、八九十年代、当代等 6 个流行音乐的典型代表年代。

第二步，音乐特征的选取和预处理

首先我们利用酷狗音乐软件，把 6 个不同时期的流行音乐随机下载各 100 首歌曲。因为要求根据流行音乐原唱录音或曲谱来推断一首流行音乐所处的年代，所以对歌曲特征的选取和处理成为首要任务。而对音乐特征的提取一定会利用相关的软件来提取出音乐数据信息，然后再整理这些音乐数据信息得到音乐特征向量，这便可以完成数据的预处理。

第三步，利用数据建立判别分类

在第二步建立音乐特征向量,我们来建立分类器模型。所以选择什么样的分类器，建立什么样的模型成为本文重要讨论的问题。

三、模型假设

1. 假设我们讨论得到的音乐时代是正确、合理的。
2. 假设我们在音乐库中得到的乐曲具有代表性。
3. 假设我们所收集到的数据和音乐的样本都足够准确，并且文本格式转换与压缩处理没有从本质上改变音乐的样本特征。
4. 假设每个音乐样本所截取的音频片段代表了整首歌的大部分信息。
5. 假设每一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。
6. 假设音乐风格类型只受音乐本身的物理属性的影响。
7. 假设对流行音乐的时代的分类不受到乐器的影响。

四、符号说明

表 1 模型一符号说明

符号	说明
x_1	左右声道平衡度
x_2	主音量
x_3	平均音长

x_4	平均力度
x_5	平均音高
x_6	音程比
$Balance_i$	MIDI文件中音轨 i 的平衡度
$Volume_i$	MIDI文件中音轨 i 振幅
$Velocity_{ij}$	MIDI文件中音轨 i 表征音的强弱程度
$Interval$	MIDI文件中音高的绝对值
$Pitch_j$	MIDI文件中第 j 个音符的音位
$Count_{ij}$	存放MIDI文件中音轨 i 音程统计个数
$ToneM_i$	MIDI文件中第 i 个音轨Interval ≤ 6 的音程数之和
$ToneSum_i$	MIDI文件中第 i 个音轨同一音轨音程之和
w_0	最优超平面系数矩阵
b	N 维余矢量
$\min J(W, b)$	平方标准差函数

表 2 模型二符号说明

符号	说明
A_i	n 维度特征向量第 i 个属性值
C_i	特征向量属性组中第 i 个类
X	数据样本
W_i	属性 A_i 的权重值

$Cov(X, Y)$	随机变量 X 和 Y 的协方差
$D(X)$	随机变量 X 的方差
ρ_{XY}	随机变量 X 和 Y 的相关系数

表 3 模型三符号说明

符号	说明
H	最优分类超平面
w	x_4 最优分类超平面系数矩阵
ξ	松弛变量
C	错误惩罚力度因子
a_i, b_i	拉格朗日乘子
x_6	音程比
L_d	拉格朗日函数对偶式
a^*	对最优分类超平面二次规划的解
$K(x_i, y_i)$	内核函数
$G(x, y)$	指数径向基函数

表 4 模型四符号说明

符号	说明
F	频率
H_i	高通滤波后频率的第 i 个能量

N_i	汉明窗过滤后频率的第 <i>i</i> 个能量
$RN(N)$	滤波函数
S	音乐指纹特征点集合

五、模型建立与求解

5.1 流行音乐年代的划分

流行音乐已经发展了近一个世纪，其主要特点也在随着时代的发展而变化，不同时代的流行音乐反应出的特点如下表所示

表 5 中国大陆各个时期流行音乐

时期	特点	代表歌手	代表歌曲
旧上海	以“拜金”“色情”为主题，受欧美流行思潮影响最多【1】	白虹、龚秋霞、姚莉、李香兰	《一代妖姬》、《天涯歌女》、《得不到的爱情》、《玫瑰玫瑰我爱你》
抗战时期	革命歌曲，进步歌曲，呼吁同胞抵抗日寇侵略，其中“时代曲”具有易于传唱的通俗性和富有时代气息的新潮性	丁善德 黎英海	《黄河大合唱》《救亡进行曲》、《松花江上》、《大刀进行曲》
建国初期	配合革命宣传为目的，以群众歌曲和红军歌曲为主，保持着民间歌曲的特色	王怀大 符其贤	《歌唱祖国》 《十送红军》
文革时期	高、硬、快、响，单一乏味	张振富 叶彩华	《东方红》 《大海航行靠舵手》
八九十年代	抒情、甜美，渴望得到某种解脱和松弛，也不乏激奋情绪【2】	费翔 邓丽君	《冬天里的一把火》 《如果没有遇见你》
当代	内容浅显通俗，表意直接，抒情达意方面朗朗上口，适宜流行传唱。个性十足，融入中国元素进行许多翻唱【3】	周杰伦 李宇春	《双节棍》 《少年中国》 《吉祥三宝》 《吻别》

5.2 音乐特征的数据预处理

5.2.1 音频文件的选择

音乐信息的记录主要有三种记录格式：声音文件、MIDI 文件和模块文件【4】。

Lingand R 等人通过研究【5】，MIDI 文件具有精度高、通用性好、切编码格式简要

等特征。而且对于 MIDI 文件，许多音乐的基本特征信息都可以从中直接提取出来，这就为音乐特征的识别提供了极大地方便。所以，文本选择了 MIDI 乐曲作为提取音乐特征量数据源。

MIDI 格式音乐文件记录乐曲所有的乐谱和演奏过程，许多音乐的基本特征（比如音高、音长、音色、速度和力度等）都是可以直接提取的，并且在 MIDI 文件中每一类特征都是用 0~127Byte 类型存储的，并且 MIDI 格式的音乐文件可以有 1 至 16 个音轨【6】。

下图就是一个 MIDI 音乐文件的格式【7】

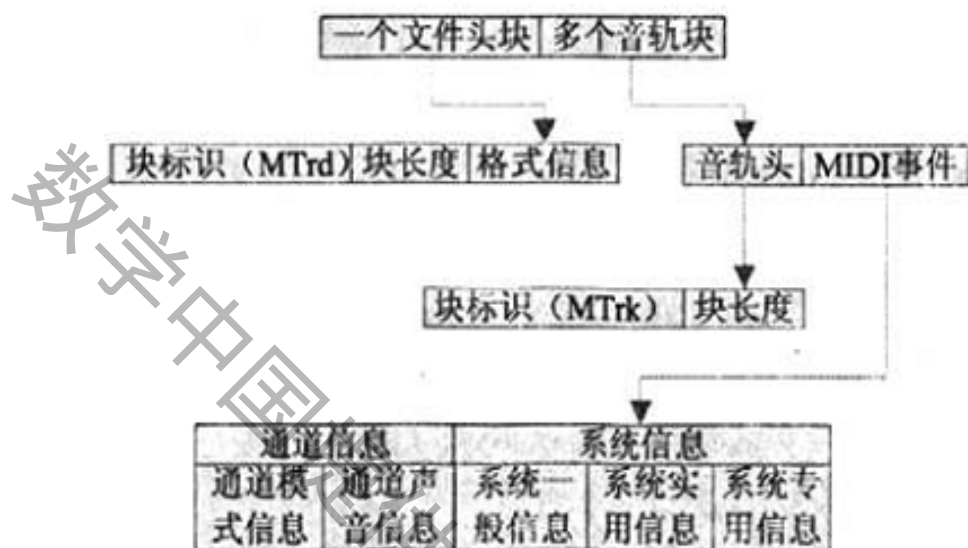


图 1 MIDI 音乐文件的基本结构

所以综合分析，文本选择 MIDI 乐曲作为提取音乐特征量数据源。

5.2.2 音乐特征量的选取

根据音乐乐理及实例数据分析。音乐的主旋律特征主要表现在左右声道平衡度、主音量、平均音长、平均力度、平均音高和音程比等几个方面。因此本文选择以上这 6 个特征作为音乐特征量。

前面 5 个特征量都能通过 C# 直接从 MIDI 文件中提取出相关数据计算而得（编程代码见附录 1），对于音程比，本文引用了文献【8】【9】的方法，得出了如下处理步骤：

首先对音乐的音高信息处理，得到所需要的音程统计数据。

设 Interval 是相邻的两个音符 $Pitch_j$ 和 $Pitch_{j+1}$ 的音高的绝对值，那么根据如下公式：

$$Interval_j = \begin{cases} |Pitch_{j+1} - Pitch_j| & |Pitch_{j+1} - Pitch_j| < 25 \\ 25 & |Pitch_{j+1} - Pitch_j| \geq 25 \end{cases}$$

计算处理，数组 $Count_{ij}$ 存放音轨 i 的音程统计个数的结果，将 $Interval_j \leq 6$ 的音程数进行累加并记为 $ToneM_i$ ，即 $ToneM_i$ 为音轨 i 的音程 0-6 度的音程。将音轨 i 的所有音程进行累加统计得到 $ToneSum_i$ ，将音 0-6 度的音程所占的比列作为音程特征量，即：

$$\text{音程比} = \frac{\text{Tone}M_i}{\text{ToneSum}_i} = \frac{\sum_{j=0}^6 \text{Count}_{ij}}{\sum_{i=0}^6 \text{Count}_{ij}}$$

下面为对有 n 个音轨的 MIDI 文件的第 i 个音轨的音乐特征量提取计算方法【10】

表 6 音乐特征量的选取

音乐特征量	计算公式	简述
左右声道平衡度	$x_1 = 1 - \frac{ \text{Balance}_i - 64 }{64}$	音轨 i 的音符在左右声道中音量大小的比列
主音量	$x_2 = \frac{\text{Volume}_i}{127}$	音轨 i 声音振幅的大小
平均音长	$x_3 = \frac{1}{n} \sum_{j=1}^n \text{Duration}_{ij}$	音轨 i 音符持续时间的平均值
平均力度	$x_4 = \frac{1}{n} \sum_{j=1}^n \frac{\text{Velocity}_{ij}}{127}$	音轨 i 表征音的强弱程度，音轨能量的大小
平均音高	$x_5 = \frac{1}{n} \sum_{j=1}^n \frac{\text{Pitch}_{ij}}{127}$	音轨 i 不同高低的聲音
音程比	$x_6 = \frac{\text{Tone}M_i}{\text{ToneSum}_i} = \frac{\sum_{j=0}^6 \text{Count}_{ij}}{\sum_{i=0}^6 \text{Count}_{ij}}$	音轨 i 相邻两音高差的绝对值

5.2.3 H—K 算法的主旋律提取

值得我们注意的是 MIDI 音乐文件的不同音轨代表的是不同的旋律，其中主旋律音轨只有一条，但伴奏旋律音轨却有多条。所以为了建立音乐的特征向量，对主旋律音轨的提取是必须的。

基于对各音轨特征量的统计，设想采用某种算法来构建成一种模型，通过此模型可以自动地定位该特征量表征的音轨是主旋律音轨还是伴奏旋律音轨。显然，对主旋律音轨定位是一个二分类判别问题。解决该问题的主要方法就是构造一个合适的分类器。

H-K 算法针对两类或多类分类问题，在特征空间中寻找一个最优的超平面作为两类的分割[12]。H—K 算法思想很朴实，就是在最小均方误差准则下求得权矢量。它适用于线性可分和非线性可分的情况，对于线性可分的情况，给出最优权矢量，对于非线性可分得情况，能够判别出来，以退出迭代过程。因此文中将运用 H—K 算法进行主旋律音轨定位。

现由左右声道平衡度、主音量、平均音长、平均力度、平均音高和音程比构成一个 6 维特征空间，音乐主旋律特征向量为 $(x_1, x_2, x_3, x_4, x_5, x_6)^T$ 对应于特征空间中的一个点，

运用已知的样本进行学习产生二个最优超平面 $d(x) = w_0x + w_7$ ，其中 $w_0 = (w_1, w_2, w_3, w_4, w_5, w_6)$ ，将特征空间分成 2 个子区域，使不同模式分布在不同的子区域中。将特征量符号规范化，即将特征矢量加上一维分量 1，并将伴奏音轨的所有特征分量都乘以 -1，得到特征量增广矩阵 X ，为了使其解更可靠，引入 N 维余矢量 $b > 0$ ，于是不等式方程组变成：

$$w_0 X \geq b \geq 0$$

$$\text{式中 } w_0 = (w_1, w_2, w_3, w_4, w_5, w_6, w_7)$$

使用平方误差标准函数：

$$\min J(w, b) = \|w_0 X - b\| = \sum_{i=1}^n (w_i x_i - b_i)^2$$

$H-K$ 算法将准则函数 $J(\cdot)$ 视作 W 和 b 的函数，在迭代过程中修正 w 的同时，也对矢量 b 进行调整，运用最优化技术求得准则函数 J 关于 w 和 b 的极小值点。在迭代调整过程中，应满足 b 的各分量均为正值的约束，同时也为 J 使 w 更趋向解区的中心， $b(k)$ 的各分量只能向增大的方向调整。

5.2.4 模型求解结果

本文中“后来.mid”为实例运用 $H-K$ 分类算法，所得结果如表 2 所示。由表 2 可知，音轨 2 是音乐“后来. mid”的主旋律音轨。

表 7 “后来,mid”的主旋律特征抽取实例

音轨号	音轨 1	音轨 2	音轨 3
发音时间	00:42:300	00:42:300	00:42:300
左右声道平衡度	0.6	0.8	0.9
主音量	3.1542	0.2325	1.1234
平均音长	86.41936	62.45614	65.41936
平均力度	0.9851	0.9085	0.62992
平均音高	0.7815	0.9732	0.8321
音程比	0.984848	0.7	1
音轨判别值	-0.035	0.037	-0.022

到此本文就完成了音乐主旋律音轨的特征向量的建立。接下来着手解决如何根据音乐主旋律音轨的特征向量去判断音乐的年代，这实质上是一个数据分类问题。

数据分类是信息化时代数据挖掘的主要内容之一，主要是通过分析训练数据样本，产生关于类别的精确描述。分类的目的是构造一个分类模型(也常常称作分类器)，该模型能把数据库中的数据项映射到给定类别中的某一个。分类技术解决问题的关键在于如何构造分类器。

5.3 模型二 基于相关系数的加权 Bayes 分类器

相关系数是用来测定变量间相关关系程度及方向的统计指标【11】。对分类问题而言，条件属性和决策属性之间的相关程度越高，条件属性对分类的重要性应越大，因此，

本文针对加权 Bayes 分类模型提出了基于相关系数的权重求解方法，并对算法原理进行了论证。

5.3.1 朴素 Bayes 文本分类【12】

设每个数据样本用一个 n 维特征向量 $X = [x_1, x_2, \dots, x_n]$ 表示,分别描述 n 个属性 A_1, A_2, \dots, A_n , 上的值, 假定分为 m 个类, 分别用 C_1, C_2, \dots, C_m 表示。对于给定的一个未知的数据样本 X (即没有类标号), 根据 Bayes 定理:

$$P(C_i|X) = \frac{P(X|C_i)P(C_i)}{P(X)}$$

由于 $P(X)$ 对于所有类为常数, 最大化后验概率 $P(C_i|X)$ 可转化为最大化先验概率 $P(X|C_i) \cdot P(C_i)$ 如果训练数据集有许多属性和元组, 计算 $P(X|C_i)$ 的开销可能非常大, 为此, 通常假设各属性的取值互相独立, 这样就有:

$$P(X|C_i) = \prod_{k=1}^n p(x_k|C_i)$$

(注: 先验概率 $p(x_1/C_i), p(x_2/C_i), \dots, p(x_n/C_i)$ 可以从训练数据集求得)

根据此方法, 对一个未知类别的样本 X , 可以先分别计算出 X 属于每一个类别 C_i 的概率 $P(X|C_i) \cdot P(C_i)$, 然后选择其中概率最大的类别作其类别。即朴素 Bayes 分类模型为:

$$V_{NB}(X) = \arg \max_c \left(P(C) \prod_{k=1}^n P(a_k|c) \right)$$

朴素 Bayes 分类器的 Matlab 代码见附录 2。

5.3.2 加权朴素 Bayes 分类

由于在实际中难于满足朴素 Bayes 条件独立性的假设, 可给不同的属性赋不同的权值使朴素 Bayes 得以扩展, 则加权朴素 Bayes 模型为

$$V_{WNB}(X) = \arg \max_c \left(P(C) \prod_{k=1}^n P^{W_k}(a_k|c) \right)$$

其中 W_k 代表属性 A_k 的权值, 属性的权值越大, 则该属性对分类的影响就越大。所以, 加权朴素 Bayes 的关键问题就在于如何确定不同属性的权值。

5.3.3 相关系数

训练样本集由多个条件属性和一个决策属性描述, 可将各条件属性字段视为随机变量 $X_i (i=1, 2, \dots, n)$, 决策属性视为随机变量 Y 。各属性的属性值分布即为各随机变量的取

值分布，若任一随机变量 X_i 的所有可能的取值为 $X_{ij}(j=1,2,\dots,s)$ ，则 X_i 取各个可能值的概率 $P(X_i = X_{ij}) = P_j(j=1,2,\dots,s)$ 称为随机变量 X_i 的概率分布。这样，整个训练样本集即为若干随机变量的集合，各个随机变量的取值按一定的概率分布。于是就可以度量每个条件属性与决策属性之间的相关程度。设 s, m 分别为随机变量 X_i 与 Y 的取值个数，分别计算随机变量 $X_i(i=1,2,\dots,n)$ 与 Y 的协方差：

$$\text{Cov}(X_i, Y) = \sum_{j=1}^s \sum_{k=1}^m [x_{ij} - E(X_i)][y_k - E(Y)] p_{jk} \quad (1)$$

其中， $P(j=1,2,\dots,s; k=1,2,\dots,m)$ 为 (X_i, Y) 的联合分布率。随机变量 X_i 与 Y 的方差为：

$$D(X_i) = \sum_{j=1}^s [x_{ij} - E(X_i)]^2 p_j \quad (2)$$

其中， $P(X_i = X_{ij}) = P_j(j=1,2,\dots,s)$ ，是 X 的分布率。

$$D(Y) = \sum_{k=1}^m [y_k - E(Y)]^2 p_k \quad (3)$$

其中 $P_k = P\{Y = y_k\} \{k=1,2,\dots,m\}$ 是 Y 的分布率，随机变量 X_i 与 Y 的相关系数为：

$$\rho_{X_i Y} = \frac{\text{Cov}(X_i, Y)}{\sqrt{D(X_i)D(Y)}} \quad (4)$$

由式 (1) ~ 式 (4) (未标出) 可知，相关系数 $\rho_{X_i Y} \in [-1, 1]$ ，它是两个变量 X 和 Y 之间线性关系强弱的一种描述性测量，或反映相关关系密切程度的重要指标。

5.3.4 相关系数的加权 Bayes 分类算法原理

5.3.4.1 定理 设分类 T 具有 n 个条件属性和 1 个决策属性，可分别用随机变量 $X_i(i=1,2,\dots,n)$ 和 Y 表示，则在加权朴素 Bayes 分类模型中，可将

$$w_i = |\rho_{X_i Y}| = \left| \frac{\text{Cov}(X_i, Y)}{\sqrt{D(X_i)D(Y)}} \right| \text{ 作为分类表 } T \text{ 第 } i \text{ 个条件属性的权重系数。}$$

5.3.4.2 证明 如果考虑以随机变量 X 的线性函数 $a + bX$ 来近似表示随机变量 Y ，以均方差：

$$e = E[Y - (a + bX)]^2 = E(Y^2) + 2bE(X^2) + a^2 - 2bE(XY) + 2abE(X) - 2aE(Y)$$

来衡量以 $a + bX$ 近似表达 Y 的好坏程度，那么 e 的值越小表示 $a + bX$ 与 Y 的近似程度越好。取 a, b 使 e 最小，将 e 关于 a, b 求偏导数，并令它们等于零，得：

$$\begin{aligned}\frac{\partial e}{\partial a} &= 2a + 2bE(X) - 2E(X) = 0 \\ \frac{\partial e}{\partial b} &= 2bE(X^2) - 2E(XY) + 2aE(X) = 0\end{aligned}$$

由上面两式可得：

$$\begin{aligned}b_0 &= \frac{Cov(X, Y)}{D(X)} \\ a_0 &= E(Y) - b_0E(X) = E(Y) - \frac{b_0Cov(X, Y)}{D(X)}\end{aligned}$$

将 a_0, b_0 代入式 (10) 得

$$\min E[(Y - (a + bX))^2] = E[(Y - (a_0 + b_0X))^2] = (1 - \rho_{XY}^2)D(Y)$$

由上式可知均方差 e 是 $|\rho_{xy}|$ 的严格单调减函数，这样 ρ_{xy} 的含义就很明显了。当 $|\rho_{xy}|$ 较大时， e 较小，表明 X, Y 联系较紧密；特别当 $|\rho_{xy}| = 1$ 时， X, Y 之间以概率 1 存在着线性关系。 ρ_{xy} 是一个可以用来表征 X, Y 之间线性关系紧密程度的量。当 $|\rho_{xy}|$ 较大时， X 和 Y 线性相关的程度较好；当 $|\rho_{xy}|$ 较小时， X 和 Y 线性相关的程度较差；当 $|\rho_{xy}| = 0$ 时， X 和 Y 不相关。因此， $|\rho_{xy}|$ 表示了随机变量 X 和 Y 的相关性程度，可作为加权 Bayes 分类模型中条件属性的权重系数。

5.3.4 算法实现

实现关键在于求解各条件属性与决策属性间的相关系数，查阅了关于相关系数的加权 Bayes 的相关资料【13】，找到并且改进了基于相关系数的加权 Bayes 分类算法，以下为本文采用的算法实现。

具体算法如下：

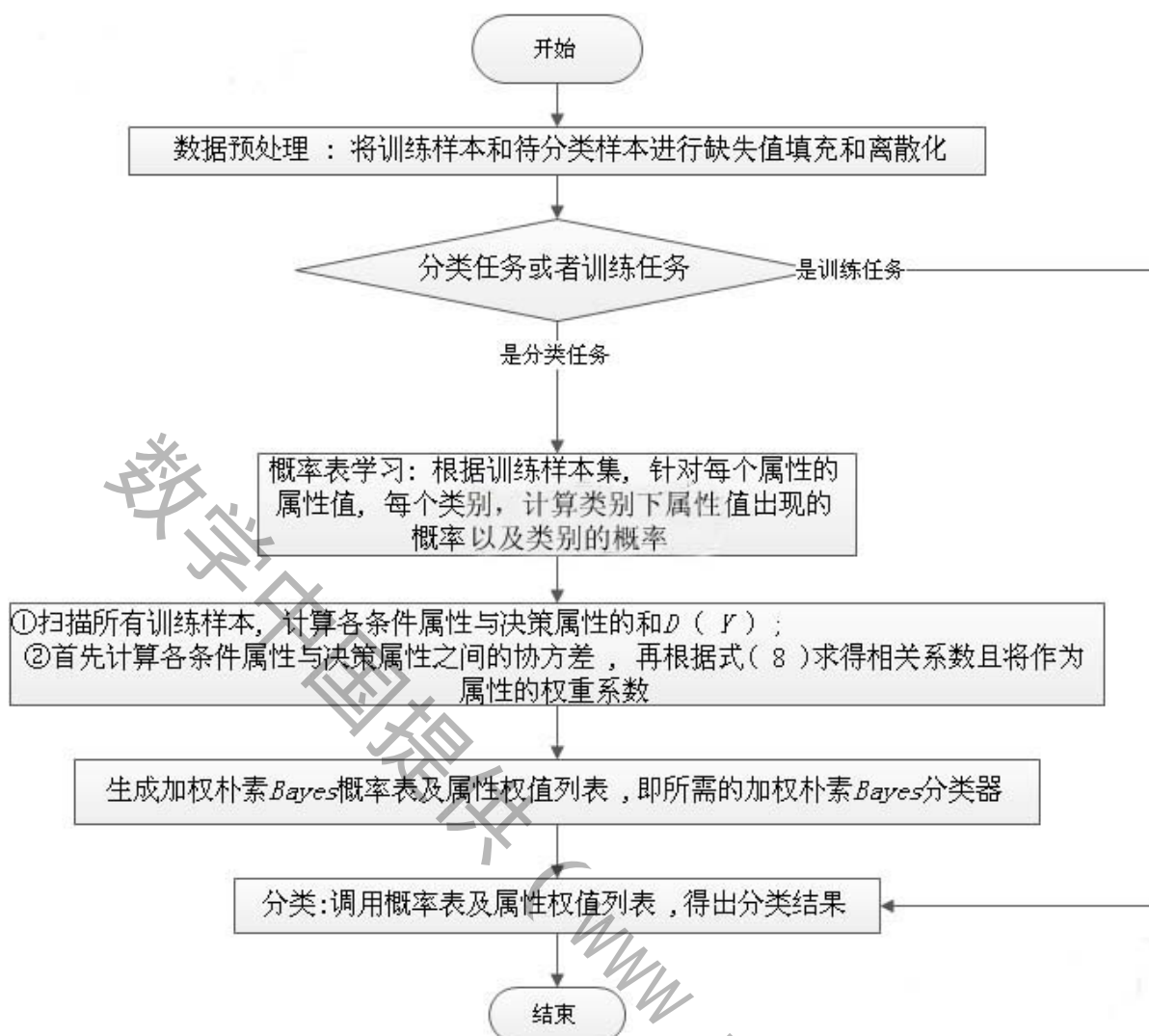


图2 基于相关系数的加权 Bayes 分类器算法实现

5.3.5 模型求解结果

然后，将全体样本按照 4:1 的比例随机划分为训练集和测试集，用我们建立的分类器做训练和测试。确定了旧上海、抗战和解放战争、建国初、文革时期、八九十年代、当代这 6 流行音乐时期，通过实验验证上述基于相关系数的加权朴素 Bayes 分类的可行性和优越性，按照下图的神经网络结构方案，分类器以每个时代流派各 80 首乐曲训练，另外 20 首乐曲样本进行测试。实验结果见下表 1：

表 8 基于相关系数的加权 Bayes 分类算法时代分的实结果

所处时代	正确识别数	错误识别的其编号	正确率
旧上海	18	1003,1025	90%
抗战和解放战争	19	2023	95%
建国初	17	3012,3064,3086	85%
文革时期	17	4033,3075,3078	85%
八九十年代	19	5024	95%
当代	14	6002,6043,6056,6076,6080,6081	70%

5.4 模型三:支持向量机分类器

对于音乐的时代分类,可以视为对音乐主旋律音轨的特征向量六维空间的线性不可分问题,即需要寻找高维空间的超平面对高维空间数据点进行区域划分。

5.4.1 支持向量机基本理论【14】

SVM 从线性可分情况下的最优分类面发展而来。

最优分类面就是要求分类线不但能将两类正确分开(训练错误率为 0),且使分类间隔最大。

SVM 考虑寻找一个满足分类要求的超平面,并且使训练集中的点距离分类面尽可能的远,也就是寻找一个分类面使它两侧的空白区域(margin)最大。

定义 过两类样本中离分类面最近的点且平行于最优分类面的超平面上 H_1, H_2 的训练样本就叫做支持向量

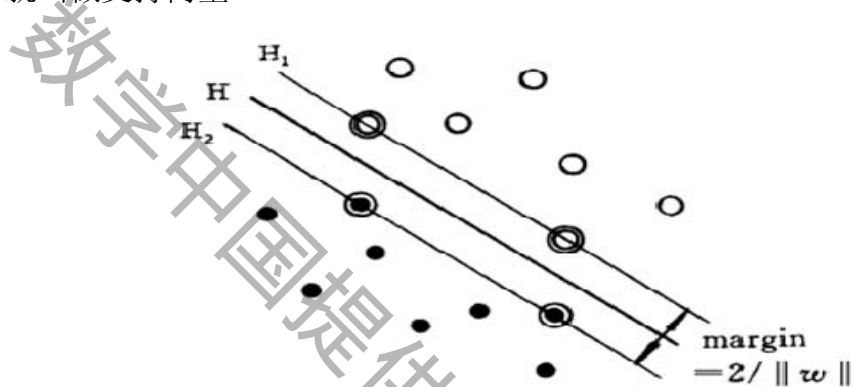


图3 利用 SVM 分类器在二维空间中关于寻找优秀分类面

5.4.2 支持向量机高维空间中的最优分类面理论

因为超平面分类函数只涉及到训练样本之间的内积运算($x_i \cdot x_j$)。因此,在高维空间中只需进行内积运算,这种内积运算可通过定义在原空间中的函数来实现,甚至不必知道变换的形式。

所以在最优分类面中采用适当的内积函数就可以实现某一非线性变换后的线性分类,而计算复杂度却没有增加。

5.3.3 支持向量机分类

支持向量机训练分类器的目的就是找到最优分类超平面 H , H 要最大化两类凸包到分类超平面的垂直距离,并使训练的误差最小。又因为本文分类对象是音乐主旋律音轨的特征向量,是一个六维特征向量,所以选择的是六维空间。

$$\frac{1}{2} \|w\|^2 + C \sum_{i=1}^6 \xi_i$$

其中 ξ 是松弛变量,作为对误分的修正。 C 是错误惩罚力度因子。引入拉格朗日乘子

$\alpha_i \geq 0, \beta_i \geq 0$ 式(5)优化问题由拉格朗日函数对偶式(6)给出。

$$L_d = \sum_{i=1}^6 \alpha_i - \frac{1}{2} \sum_{i=1}^6 \sum_{j=1}^6 \alpha_i \alpha_j y_i y_j (x_i \cdot y_j)$$

约束条件为 $\sum_{i=1}^6 \alpha_i y_i = 0; 0 \leq \alpha_i \leq C; i = 1, \dots, n$ 。二次规划问题的解为 $\alpha^* = (\alpha_1^0, \alpha_2^0, \dots, \alpha_n^0)$

$\alpha_i^0 > 0$ 所对应的训练样本,称之为支持向量。

对于实际中难以线性分类的情况,支持向量机通过选择合适的非线性变换将输入向量映射到高维特征空间,在高维空间建立线性超平面。设存在映射 $X \rightarrow Z, X \in R^d, Z \in R^k, k \geq d$,它可以把输入样本 $x_i \in X$ 映射到 k 维特征空间 Z 。若函数 $K(x_i, x_j) = \psi(x_i) \cdot \psi(x_j)$,则称 K 为内核函数。代入式(6)得:

$$L_d = \sum_{i=1}^6 \alpha_i - \frac{1}{2} \sum_{i=1}^6 \sum_{j=1}^6 \alpha_i \alpha_j y_i y_j K(x_i, x_j)$$

在高维空间 Z 中,最终的分类决策函数是:

$$f(x) = \text{sgn} \left(\sum_{\text{支持向量}} \alpha_i^0 y_i k(x, x_j) - b \right)$$

常用的内核函数有:多项式内核函数,高斯径向基函数 RBF,指数径向基函数 ERBF 等。在实验采用的是指数径向基函数,其函数如下:

$$G(x, y) = e^{-\frac{\|x-y\|^2}{2\sigma^2}}$$

支持向量机训练算法采用序列最小优化 SMO 算法。该算法的优点是针对少数样本的二次规划问题可以有解析解的形式,从而避免了多样本情况下的数据解不稳定及耗时问题,同时也不需要大的矩阵存储空间。其工作集的选择采用启发式,通过两个嵌套的循环来寻找待优化的样本变量,循环优化,直到全部样本都满足最优条件。

支持向量机分类识别的 MATLAB 代码见附录 3。

5.4.4 模型求解结果

研究选用 6 种时期音乐类型:旧上海、抗战和解放战争、建国初、文革时期、八九十年代、当代这六大类。实验数据库包含 600 首流行歌曲,且每段时期音乐类型各有 100 首,歌曲全部下载自酷狗音乐。

每类 80 个样本作为训练集,20 个样本作为测试集。训练测试支持向量机,循环迭代参数 C 和 D ,从实验中得到精度最高的 C 和 D 。当存在多个相同的最高测试率时,取支持向量少的那组参数作为最优参数。以下展示支持向量机分类实现;

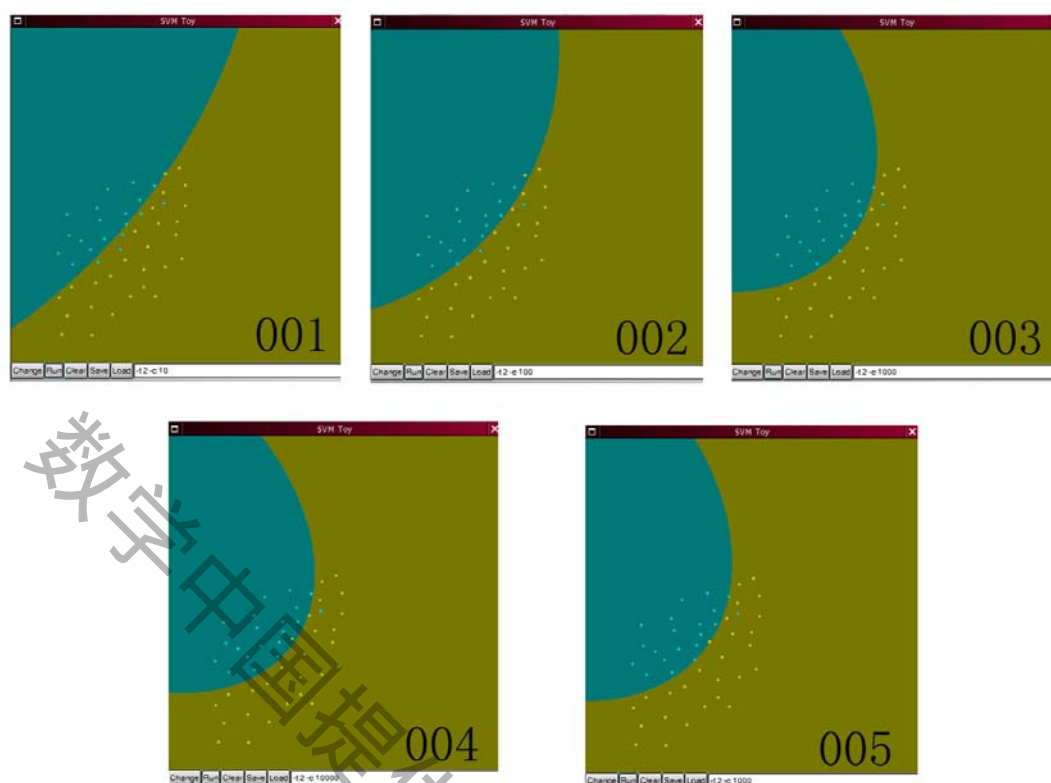


图 4 支持向量机分类实现显示的演示

最终的实验结果见下表：

表 X 基于相关系数的加权 Bayes 分类算法时代分的实结果

所处时代	正确识别数	错误识别的其编号	正确率
旧上海	19	1025	90%
抗战和解放战争	20		100%
建国初	19	3064	95%
文革时期	19	3075	95%
八九十年代	20		100%
当代	16	6043,6056,6076 ,6081	80%

音乐年代判定在多媒体应用中有重要的应用价值。本文采用支持向量机分类音乐类型,分类正确率可 90%左右,除了对当代音乐年代判别差强人意之外,其他对音乐年代判别分类表现最好。表明该方法合理、有效。

5.5 模型四:基于多媒体内容的音乐检索的识别年代

基于内容的检索是目前互联网上多媒体数据的研究中的一个新的方向,这种检索具有如下特点:

- (1) 从多媒体数据本身的内容中提取信息的特征线索。
- (2) 基于内容的检索方式是一种模糊匹配。
- (3) 基于大型数据库(集)的快速检索【15】。

接下来,我们将从选择合适的系统结构、音频格式转化、语谱分析与指纹提取、指

文库构建、音乐指纹比对及结果排序这五个方面来详细阐述基于内容的检索。

5.5.1 系统需求分析与概要设计

本系统主要为用户提供一种基于内容的音乐检索服务，即用户上传一段原始歌曲片段（10s~30s），系统根据这个歌曲片段，在服务器上音乐库中查找到相对应的歌曲信息，然后将信息返回给用户【16】【17】。

5.5.1.1 系统框架概要设计

本节主要有两部分组成，分别是系统整体框架设计以及功能模块设计。为了维护系统的可靠性，稳定性和可扩展性，系统采用了一种Master/Slave 结构。

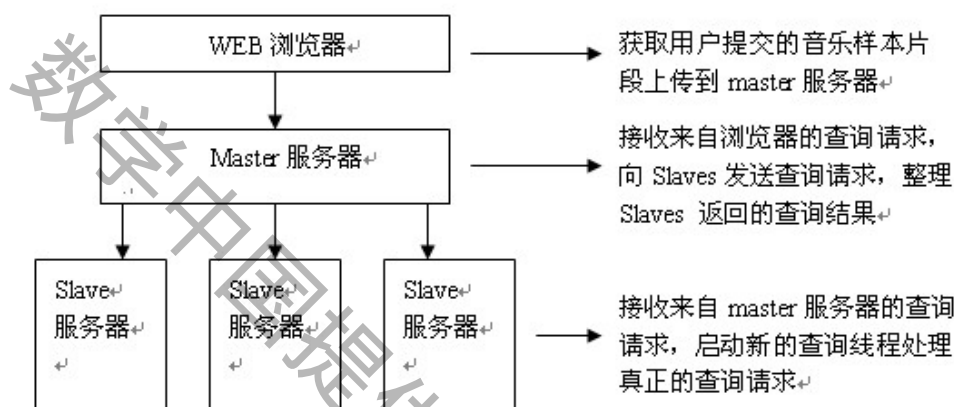


图5 音乐检索系统体系结构图

(1)WEB 浏览器: 获取用户上传的音乐样本片段，上传到 Master 服务器，并向 Master 服务器发送查询请求。

(2) Master 服务器: 主要负责接收来自浏览器的查询请求，然后将查询请求发送给每一个跟它相关联的Slave 服务器上。

(3) Slave 服务器: 所有的Slave服务器接收到Master发送的查询请求后，提取音乐指纹，根据该生成的音乐指纹处理该相应的查询，然后将查询结果返回给 Master 服务器。

5.5.1.2 系统功能模块划分

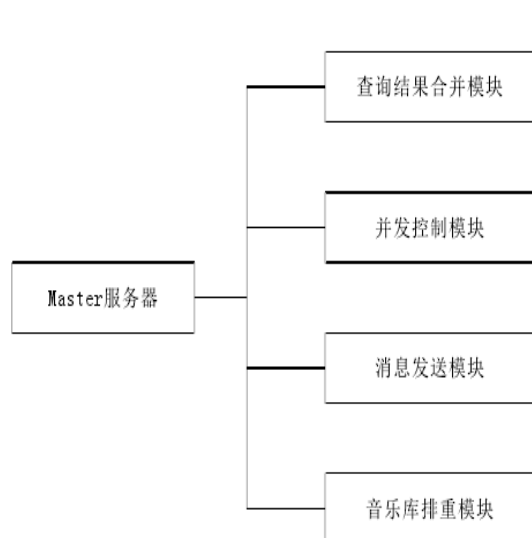


图6 Master 服务器功能模块结构图

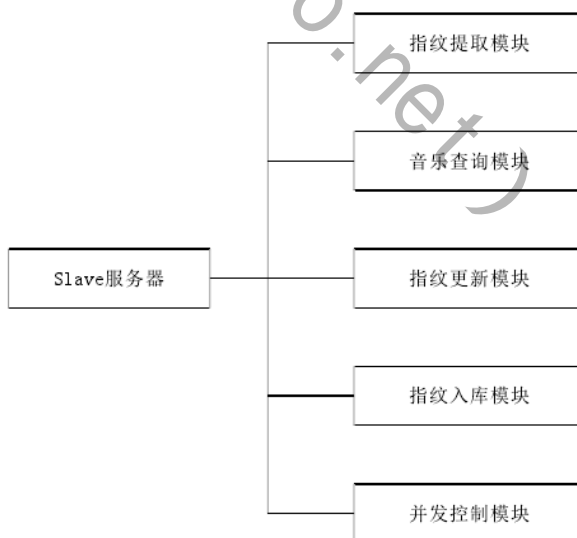


图7 Slave 服务器功能模块结构图

5.5.1.3 系统功能模块调用关系

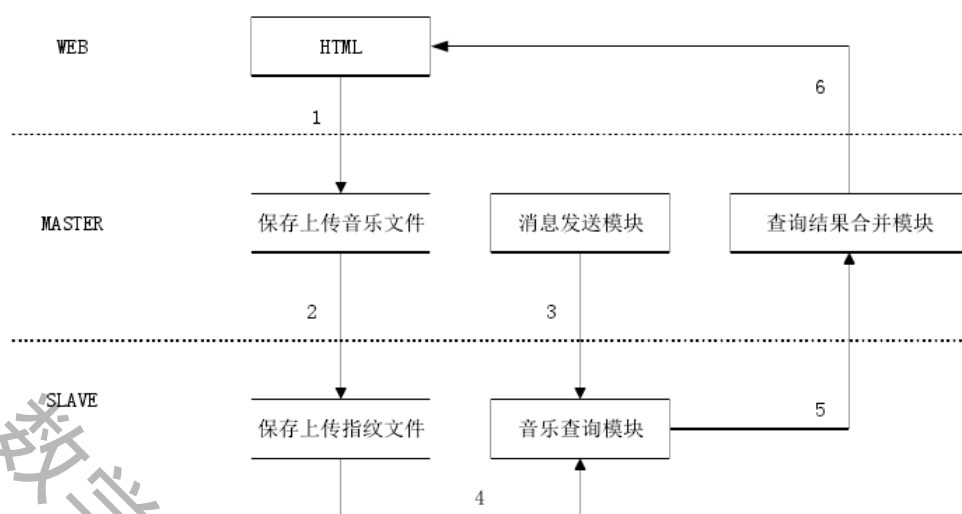


图8 音乐查询功能模块调用关系图

音乐库排重过程中各个功能模块交互过程

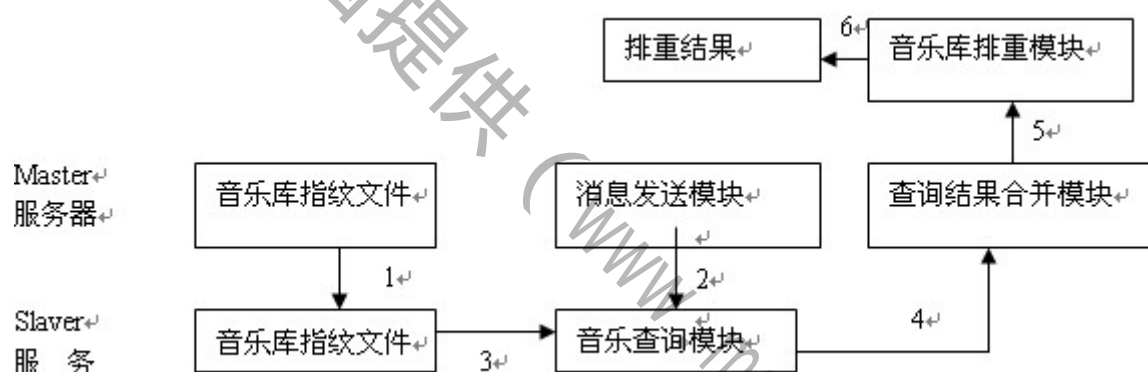


图9 音乐库排重过程功能模块调用关系图

音乐库指纹更新模块调用过程

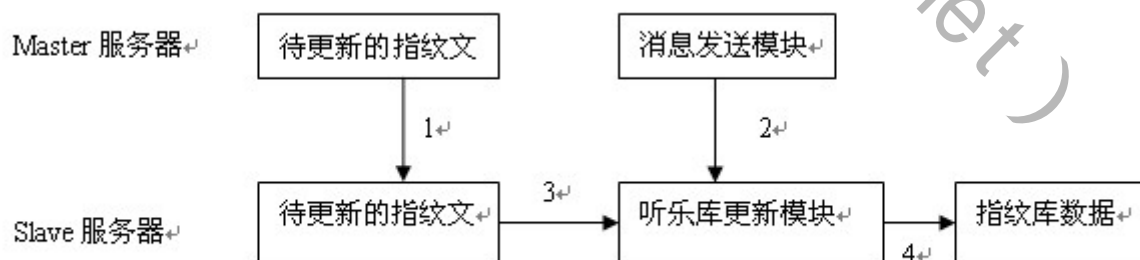


图10 音乐库指纹更新过程功能模块调用关系

5.5.1.4 系统配置设计【18】

音乐检索配置

Music_Query_Config.xml

```
<?xml version="1.0" encoding="utf-8"?>
<Music_Query_Config version="1">
    <Query_Mode>查询主程序运行方式</Query_Mode>
    <Finger_Count>指纹库中指纹数量</Finger_Count>
    <Finger_List>指纹文件列表</Finger_List>
    <Model_File>指纹库文件</Model_File>
    <Model_Bucket>构建指纹库时 hash 表桶数</Model_Bucket>
</Music_Query_Config>
```

Slave 服务器列表配置

Slaves_List.xml

```
<?xml version="1.0" encoding="utf-8"?>
<Slaves_List version="1">
    <Master_ip>Master 服务器 IP 地址</Master_ip>
    <Master_port>Master 服务器 socket 端口</Master_port>
    <Slave_Info count="slave 服务器数量">
        <slave0 ip="Slave服务器IP地址" port="Slave服务器socket端口"/>
    </Slave_Info>
</Slaves_list>
```

网络接口文件格式定义

该配置文件用来定义通过网络接口上传的音乐查询片段的格式。

QQMusic_FingerPrintQuery.xml

```
<?xml version="1.0" encoding="utf-8"?>
<QQMusic_FingerPrintQuery>
    <requestId>查询请求 ID</requestId>
    <starttime>查询片段在原始歌曲中的开始时间</starttime>
    <totalplaytime>原始歌曲播放时间</totalplaytime>
    <format>原始音乐歌曲格式</format>
    <clipplaytime>查询片段播放时间</clipplaytime>
    <musicdata encoding="base64">查询片段音乐数据</musicdata>
</QQMusic_FingerPrintQuery>
```

检索结果数据格式定义

QQMusic_FingerPrintQuery_Result.xml

```
<?xml version="1.0" encoding="utf-8"?>
<QQMusic_FingerPrintQuery_Result version="1">
    <requestId>查询请求 ID</requestId>
    <TransformTime>格式转换时间</TransformTime>
    <ExtractTime>提取音乐指纹时间</ExtractTime>
    <SearchTime>音乐数据检索时间</SearchTime>
    <MusicInfo count="音乐检索结果个数">
        <Item0 MusicId="结果歌曲在音乐库中的 ID" SongName="歌曲名">
```

```
Singer="歌手名" MatchTime="片段匹配时间偏移值"
URL="结果歌曲对应的网络播放地址"/>
</MusicInfo>
</QQMusic_FingerPrintQuery_Result>
```

5.5.2 系统实现

整体结构采用 WEB 浏览器+Master 服务器+Slave 服务器的结构实现。

1.Master 服务器实现

Master 服务器的核心功能是语谱特征的提取和生成指纹信息。

1)查询结果合并模块首先将获取所有 Slaves 服务器返回的查询结果，对这些结果按照权值进行排序后，返回给 WEB 浏览器进行展示。

2)并发控制模块；

3)、消息发送模块实现

Master 服务器的消息发送模块，向与该 Master 服务器相关联的 Slaves 服务器发送命令消息。命令消息包括音乐检索命令，指纹更新命令和指纹入库命令三种。

命令消息格式如下：

```
[query_flag][finger_name][s_time][e_time][real_s_time][insert_flag][finger_list][save_flag]
```

2、网络接口实现

网络接口设计采用 python HTTPServer 框架来实现【19】。HTTPServer 是 python BaseHTTPServer 包中的一个类，用于提供独立的 HTTPServer 服务。要实现自定义的 HTTPServer 服务器，首先定义一个类继承自 BaseHTTPRequestHandler：

```
class QQHTTPRequestHandler(BaseHTTPRequestHandler)
```

然后重写 BaseHTTPRequestHandler 中的 do_GET()方法和 do_POST()方法，分别处理 HTTPServer 接收到的 GET 请求和 POST 请求【20】。

3、Slave 服务器实现

1).并发控制模块实现

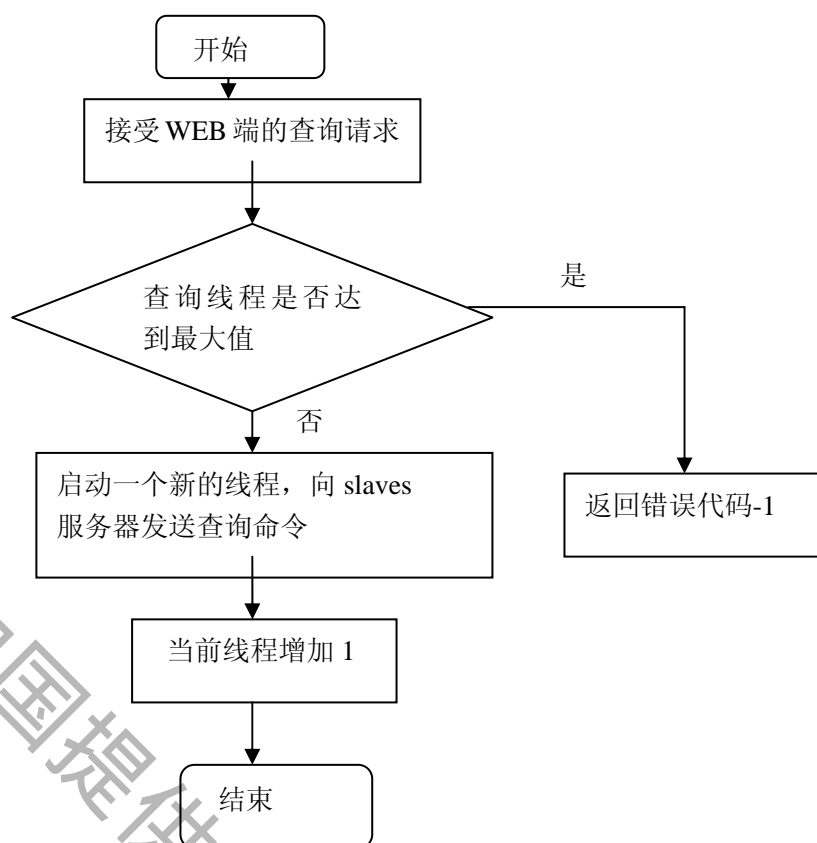


图11 并发控制模块实现

Slave 服务器的并发控制模块，主要用于控制查询线程数。

2). 指纹更新模块实现【21】

指纹更新模块的主要功能是向内存中的 hash table 中添加上新的音乐指纹文件信息。该模块运行时，更新 g_insertOrSaveFlag 标记，此时，Slave 服务器不响应新的查询请求，但并不停止当期正在运行的查询线程。当接收到查询请求时，若检测到有查询线程正在运行，则等待一个时间片段后，再次检测是否有查询线程正在运行，依次类推，直到检测到所有的查询线程处理完毕。若超过规定的时间后，还有查询线程正在运行，则放弃本次更新操作，并返回更新失败提示信息。否则，将新增的查询指纹添加到内存中的 hash table 中。最后，取消 g_insertOrSaveFlag 标记，Slave 服务器重新接收查询请求。

5.4.3 音乐检索过程设计与实现

音乐查询模块是本系统最核心的模块，检索的准确直接影响系统的实用程度。

5.4.3.1 音乐数据预处理

使用 Linux 下开源工具 ffmpeg，将上传的音乐文件转换为单通道，4K 采样率的 wav 文件，转换方法为“ffmpeg -i sample.wav -ar 4000 -ac 1 -ab 60sample_new.wav”。该类型 wav 文件的数据格式如下图所示【22】【23】。

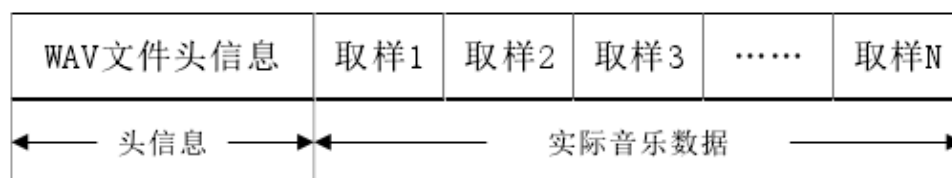


图12 单通道-8K 采样率的 wav 文件格式

5.5.3.2 语谱图数据处理

语谱图是一幅灰度图，用来表示音乐文件的特征。语谱图的横轴为时间 T ，纵轴为频率 F 。图中每一个点的能量强度则用灰度来表示。语谱图信息使用二维矩阵表示【24】。

对于得到的语谱图矩阵数据，不能直接进行求特征点的操作，原始的语谱图中含有其它非音乐的噪声信息，这些信息对生成音乐指纹来讲都是无效信息。

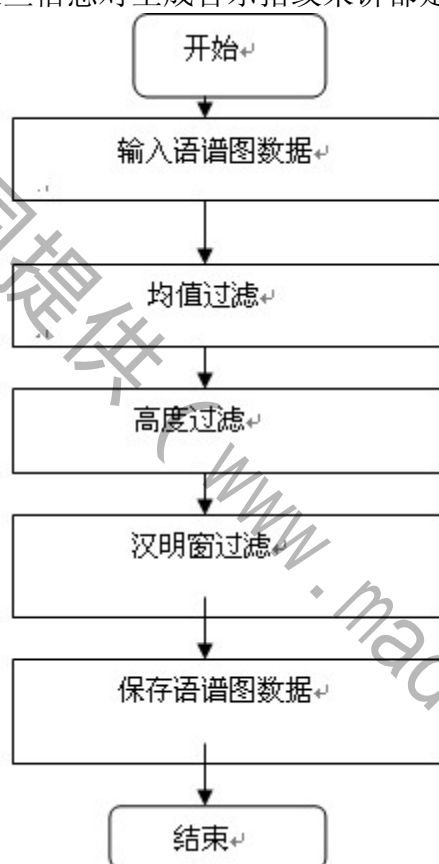


图13 语谱图去噪流程

汉明窗过滤的实现:

汉明窗滤波也是一种一维滤波，针对一个频率固定，时间连续变化的序列进行过滤。

设频率 F 对应的高通滤波后的能量序列为 $\{H_1, H_2, H_3, \dots, H_n\}$ ，汉明窗过滤后的

能量序列为 $\{N_1, N_2, N_3, \dots, N_N\}$ ，汉明窗窗长为 $2*m$ ，滤波函数

$$RN(n) = 0.54 - 0.46 * \cos[2 * \pi * n / (2 * m - 1)],$$

则 $N_I = RN_1 * N_{I-m} + RN_2 * N_{I-m+1} + \dots + RN_{2m} * N_I + m$ 。

5.5.3.3 提取特征点

1) 输入数据及格式

Dooble 型二维数组，该二维数组为均值过滤、高通过滤和汉明窗过滤后得到的语谱图二维数据。横坐标为语谱图时间采样点索引值，纵坐标为语谱图频率采样点索引值，其中的数据为预处理之后得到的采样点能量数据。

2) 处理流程及意义

特征点提取的首要要求是，保证同一首歌曲在任何情况下提取的特征点一致。在进行均值过滤时，采取了分帧处理的方法。如下图所示，其中较小的点和较大的点分别为同一个片段两次提取的特征点，这两次提取的特征点只有少部分重合或相近，效果不佳。

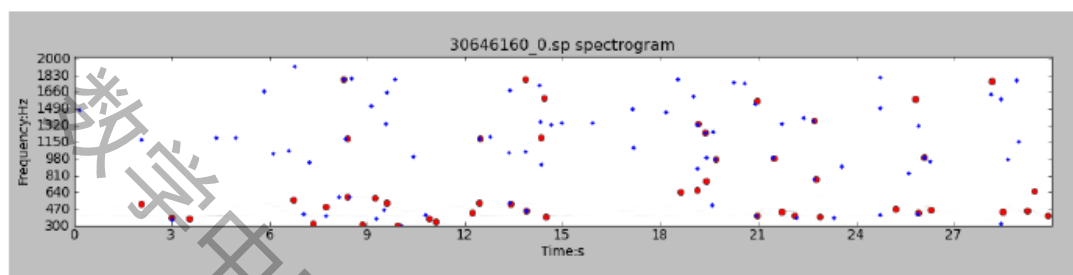


图14 特征点比较

设 wav 文件的对应的能量序列 $T = \{E_1, E_2, E_3, \dots, E_N\}$ ，对于其中的每个能量值 E_i ，若 $E_i > E_{i-1}$ 且 $E_i > E_{i+1}$ ，则 E_i 为能量序列中的几个极值位置，保存该点的时间 t_i 。最后，可以得到一个时间序列 $T = \{T_1, T_2, T_3, \dots, T_m\}$ ，其中每个时间点对应的能量值都为原始能量序列中的极值。当选取特征点时，在时间序列 T 中的时间点上选取。特征点提取流程图如下图所示：

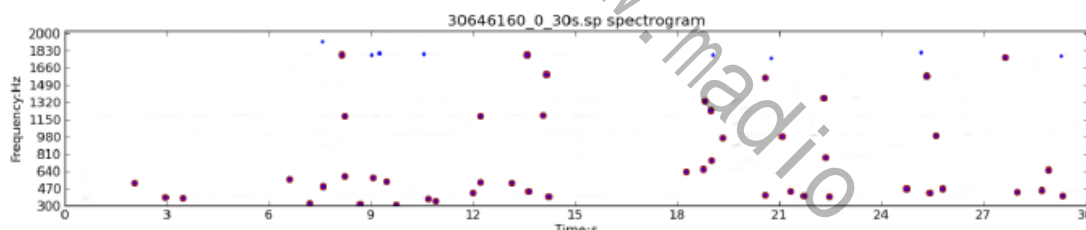


图15 改进后的特征点比较图

每个选定时间轴上的点 (t, f) ，设其能量值为 $\text{value}(t, f)$ ，若 $\text{value}(t, f) > \text{value}(t-1, f)$ 且 $\text{value}(t, f) > \text{value}(t+1, f)$ 且 $\text{value}(t, f) > \text{value}(t, f-1)$ 且 $\text{value}(t, f) > \text{value}(t, f+1)$ ，即点 (t, f) 的能量值大于与其相邻的四个能量点，则选取点 (t, f) 为一个特征点保存。图 4-12 显示该方式处理后选取特征点的结果。可以看出，两种情况下选取的特征点大部分相同。

5.5.3.4 生成音乐指纹

处理流程如下：

对于选取的每一个特征点 X ，选取其周围的一个矩形区域作为该点的目标区域，该区域中的特征点集合 $S = \{W_1, W_2, W_3, \dots, W_n\}$ ， W_i 的坐标为 (T_i, F_i) ， X 与 S 中的每一个点 W_i 生成散列数据。生成方法如下：

$$\text{Hash}(X, W_i) = (F_x, F_i, \Delta T) : (\text{ID}, T_x)$$

在实际处理过程中，定义了两个结构体变量 `struct finger_key` 和 `finger_value` 来分别表示 $(F_x, F_i, \Delta T)$ 和 (ID, T_x) 。

```
typedef struct finger_key    //定义finger_key结构体
{
    unsigned int f1:10;      //第一个采样点频率值
    unsigned int f2:10;      //第二个采样点频率值
    int deltaT:12;           //两个采样点之间的时间间隔
}KEY;

typedef struct finger_value //hash_map valueList 中元素
{
    unsigned int truckId; //该指纹对应的歌曲 ID
    unsigned int t1; //该指纹对应的时间

    //在构造 Hash Map 时调用需要重载的运算符
    bool operator==(const struct finger_value& v) const
    {
        return (truckId == v.truckId) && (t1 == v.t1);
    }
    bool operator<(const struct finger_value& v) const
    {
        return (truckId < v.truckId) || (truckId == v.truckId && t1 < v.t1);
    }
}VALUE;
```

5.5.4 系统运行与测试

5.5.4.1 Master 服务器运行说明

Master 服务器运行后，等待查询请求。当有查询请求提交后，Master 服务器的运行结果如下图所示。其中显示了待查询的指纹文件名，以及查询片段的开始时间和结束时间，以及该指纹文件对应的音乐片段在原始歌曲中的对应的真实的开始时间。同时，显示了与 Slave 服务器的连接情况，以及每一个 Slave 服务器返回的查询结果，并显示最终经过合并后的查询结果【25】。

5.5.4.2 Slave 服务器运行说明

Slave 服务器运行后，等待 Master 服务器发送的查询请求。Slave 服务器运行结果如下图所示。其运行结果分为三部分。第一部分显示当前要查询的指纹文件信息以及检索文件的指纹文件开始和结束时间。第二部分显示检索出的候选歌曲信息以及对应于该歌曲的匹配成功的指纹个数。第三部分显示了最终选取的歌曲信息。

5.5.5 模型求解

1) 系统测试环境

- (1) 系统版本：Linux version 2.6.16.60-0.21-TENCENT64-100130
- (2) 内存大小：32G

(3) 交换区大小：2G

(4) CPU：8 核心，每个核心处理速度 2.99GHz

2) 音乐检索结果排序测试与分析(4)

音乐样本库的容量为 1000 时，随机选取了 9 个音乐片段，包括原始音乐片段，加入白噪声干扰，加入回声干扰，噪声环境下录音片段等，对音乐检索模块进行的单元测试。测试结果如下表，表中的检索结果为歌曲的 ID 号。

编号	查询片段	检索结果	匹配指纹个数	说明
1	30000002.wav	30000002	1237	查询片段为原始歌曲
		30013181	1	
		30013191	1	
2	30000002_0-10.wav	30000002	41	查询片段为原始歌曲,长度为 10s
		30013181	1	
		30014011	1	
		30013191	1	
3	30000002_0-20.wav	30000002	100	查询片段为原始歌曲,长度为 20s
		30013181	1	
		30014011	1	
		30013221	1	
		30005002	1	
4	30000002_10-20.wav	30000002	46	查询片段为原始歌曲,长度为 10s
		30013221	1	
		30011141	1	
		30010211	1	
5	30000002_20-50.wav	30000002	114	查询片段为原始歌曲,长度为 30s
6	30000002_echo.Wav	30000002	2	查询片段为加入回声干扰的歌曲
		10014161	1	
		30014171	1	
7	30000002_white.wav	30000002	467	查询片段为加入白噪声的歌曲
		30015004	1	
		30012101	1	
8	30003002_r.wav	30013001	1	录音歌曲片段
9	30005002_r.wav	30010141	1	录音歌曲片段
		30005002	1	

将选取的1000个片段，分别分为旧上海、抗战和解放战争、建国初、文革时期、八九十年代、当代这6个不同的时期，用我们建立的系统做训练和测试，实验结果见下表：

表9 基于内容的音乐检索模型确定所属时代的实结果

所处时代	正确识别数	正确率
------	-------	-----

旧上海	170	85%
抗战和解放战争	193	96.5%
建国初	178	89%
文革时期	175	87.5%
八九十年代	191	95.5%
当代	186	93%

由以上结果可以看出,对于原始歌曲的片段,能够较准确的检索出正确结果,而对于白噪声和噪声环境下的录音片段,在片段长度较长时,才能正确检索出结果。由测试结果可以看出,基本上达到了预期要求。

同时还可以看出,对于每一个待检索片段,正确的检索结果都位于候选结果的前两位,保证了检索结果对于用户有良好的体验。

六、模型评价

模型优点:

在数据预处理时使用 H-K 算法的对 MIDI 格式音乐的主旋律提取,并且利用 C#从而构造出音乐六维特征向量。在不借助于需要购买注册专业的音乐软件,使用 Visual Studio 加上简单的编程就可以提取出 MIDI 的音乐特征,是一种经济、简单、高效的提取音乐特征的方法。

模型一加权朴素 Bayes 和基于相关系数的加权朴素贝叶斯分类的分类效果合理,较为准确,是一种便捷、有效的分类方法。

模型二采用由模型一求解出音乐主旋律六维特征向量作为音乐特征,使用支持向量机对音乐所处的时代进行划分,分类正准确而且较高。表明该方法合理、有效。

模型三对于每一个音乐文件,都能生成一个语谱图用来表示这个音乐文件,对该语谱图进行分析处理后就能得到一个唯一标识该音乐文件的信息。论文设计一种指纹提取算法,由不同的音乐文件提取的指纹信息之间的差异明显,指纹信息生成方式简洁,易于保存。该系统能够处理检索文件为原始音乐文件的检索请求,在处理海量音乐数据时能够有效的去除其中的重复文件。是一种利用计算机 Web 程序处理音乐分类的智能方法。

模型缺点

本文选择的音乐数据仅仅考虑了从民国到当代中国大陆的流行音乐发展状况,并没有考虑港澳台地区流行音乐发展状况,所以在流行音乐时代划分的全面性上有失偏颇。

模型二和模型三对于当代音乐的时代划分的结果差强人意。其中音乐编号为6043通过模型二和模型三的方法都没有入选。音乐编号为6043的歌曲为《青花瓷》,具有浓厚的古典风格的当代流行歌曲,和一般的流行歌曲相比独具特色,所以我们被成功划入当代歌曲。

另外在模型四中目前系统对于原始音乐片段的检索速度和准确率都比较好,当待检索的音乐片段为录音的音乐片段时,查询检索的准确率比较低

优点:

七、模型改进

1.为了进一步提高对当代流行音乐的分类效果,应该扩当训练集,多采集当代音乐数据作为训练样本。

2.对于流行音乐的时代划分应该综合全局考虑,本模型应该考虑到港澳台地区在 20

世纪 50 到 70 年代的流行音乐的风格，应该专门为港澳台流行音乐划分好时期并且相应的数据整理和时代划分。

3. 基于支持向量机时代判定分类器算法没有做优化，计算速度较慢。应该进一步改进支持向量机分类器的算法。

八、小短文

流行音乐不同时期的特点

流行音乐，又有“大众音乐”之称，实质在于流行，流行意味着这种文化产品是通俗的，是经过大众媒介和市场推广的，是讲述人们自身生活经验的。流行音乐自诞生起便备受追捧，它之所以能够经久不衰，不仅因为有悠扬通俗的旋律，更是因为每个时期的作曲家、演唱家能够很好的把握不同年代人们的喜好，其本身与时俱进的特点，便是长立于音乐之林的重要因素。

(一) 旧上海时代：纸醉金迷

○成功因素及特点

这个时期流行音乐主导因素是表现爱情、拜金主义，比如《天涯歌女》《夜上海》。而且由于其所处的场景的性质一灯红酒绿、充满着金钱和色诱，也决定了这些场景的听众不可能有较高的欣赏水平，很多就成为以“拜金”、“色情”为内容的音乐了【1】。

○时代背景

20 世纪三、四十年代的上海是远东地区最繁荣的港口和经济、金融中心，也是近代亚洲唯一的国际化大都市。这段时期，上海的文化发展非常蓬勃，是近代最富有活力的时期。很快，中国的第一首流行歌曲——《毛毛雨》出现了。这也标志着中国流行音乐的诞生。

(二) 抗战救国时期：民族存亡

○成功因素及特点

这个时期流行音乐主导因素是爱国民族主义和社会动乱。其中展现爱国民族主义流行歌曲全部在表现抗日救国、鼓励广大人民英勇奋战，如《救亡进行曲》。展现社会动乱流行歌曲全部在表现当时人民流离失所、烽烟四起的悲惨生活，比如《松花江上》。

○时代背景

1937 年到 1949 年是中国极为动荡的时期，特别是八年的抗日战争，给国家与民族带来了沉重的灾难。而这个时期，却又是中国流行音乐发展的第一个黄金时期。1937 年是抗战爆发的一年，文化界、音乐界大量的作家作曲家开始创作进步歌曲，呼吁同胞抵抗日寇侵略，其中“时代曲”以其易于传唱的通俗性和富有时代气息的新潮性，迅速成为大众最为接受的音乐类型，从而流行音乐的发展也同期进入黄金时期。

(三) 建国初期：自信豪迈

○成功因素及特点

这个时期流行音乐主导因素是政治宣传以及对社会转型催化。建国初期流行音乐都是以战斗性进行曲为主，短小的齐唱、合唱群众歌曲体裁获得较为充分发展的年代。这主要是由当时的社会环境所决定的，比如扫荡国民党反动残余势力、进行社会主义三大改造运动、五年计划的提出和实施。就算是一些抒情歌曲也带有明显的战斗性，如《告诉我，来自祖国的风》、《远航归来》等。总之，在那个朝气蓬勃的年代，正如诗人所形容的：“凡是能开的花，全在开放；凡是能唱的鸟，全在歌唱”。

○时代背景

在新中国成立以后，涌现出了许多优秀的、至今耳熟能详的经典的歌曲和配乐。这

一时期的歌曲，以乐观自信、昂扬豪迈的基本格调，反应了当时广大人民的乐观主义情绪和积极奋发的精神面貌，比如《全世界人民一条心》《在祖国和平的土地上》等表现的是胜利、欢呼的音调；《歌唱祖国》等在热情歌颂祖国的新生，往往伴随着自豪、宽阔、雄壮的气势；《咱们工人有力量》体现了青春的活力和勇往直前的气势【2】。

（四）文革时代：单调乏味

○成功因素及特点

这个时期流行音乐主导因素是政治专制主义。在文革时期流行的歌曲可以说只有4首，当时全国只许唱《东方红》、《大海航行靠舵手》《三项纪律八项注意》这三首中国歌曲和一首《国际歌》。“文化大革命”使中国音乐事业遭受了史无前例的破坏和灾难。

○时代背景

与“文化大革命”前十七年以及“文化大革命”后中国音乐发展的局面相比较，“文化大革命”十年间的音乐正处于“马鞍形”的最低谷。几乎所有的“文革”以前的优秀歌曲作品都被扣上了“封、资、修”的帽子，罗织的罪名往往十分可笑。比如：1957年王洛宾创作过一首具有新疆维吾尔族音乐风格的歌曲《日夜想念毛主席》，歌曲的主题思想“想念毛主席”用了“萨拉姆毛主席”这句维吾尔语。这首歌曾经唱遍全国。到了文革时期，作曲家被指控为“借用谐音手法来表达要‘杀了毛主席’的反革命分子”。这是文革中摧残艺术家和文艺作品的典型方法之一。

（五）20世纪八九十年代：人性自由

○成功因素及特点

这个时期流行音乐主导因素是自由主义，理想主义。文革刚刚结束，十一届三中全会后国家进行改革开放运动，人们禁锢思想和行为得到解放，所以人们需要追去自由和理想。《小草》等一批具有现实意义的自由主义歌曲的出现开始便显出人们对普通人的关注。80年代中后期的“西北风”和以崔健为代表的理想主义摇滚乐的出现则现实了流行歌曲作为一种文化现象所表现出的一种深刻的反思意识和积极的社会参与意识【3】。

○时代背景

经历了浩劫，歌曲作家们才更深刻地体会到：真情乃歌曲生命之所系。流行音乐从诞生之日起几乎就背上了另一个称呼：“靡靡之音”。其内容多为爱情，部分作品甚至歌颂腐化堕落的社会风气以营造出片歌舞升平的虚假繁荣有关，与当时我国社会的背景格格不入。在八十年代末九十年代初，人们逐渐开始比较客观的看待流行音乐，多抛开政治因素，单纯从艺术的角度来研究流行音乐，从而逐步为流行音乐正名。

（六）当代：百花齐放

○成功因素及特点

这个时期流行音乐主导因素是商业炒作，网络宣传，古典主义，而且就风格来说也是极为丰富和多样的。商业炒作例如“超女”“快男”的《想唱就唱》具有选秀浓厚的商业特点；网络宣传比如说凤凰传奇的《最炫民族风》和韩国PSY的《江南style》，两首歌曲通过网络飞速走火，产生了极为深远的影响。古典主义如周杰伦的《青花瓷》，“折柳，相思，红豆，娘子佳人在江南”，用现代的音乐唱出古典的味道。

○时代背景

随着电子传媒技术的深入发展，传播平台不断扩大，21世纪的流行歌曲会更加全球化。通过翻唱吸取他国音乐的优点使我们的音乐更优秀、走向更高层次。如：张学友的《吻别》被国外歌手翻唱，歌名是《Take me to your heart》【26】。为了迎合听众的口味，不同类型、不同风格的歌曲和歌手层出不穷，如：孙楠的高亢、刀郎的苍凉、刘若英的清甜、张含韵的乖巧等。与此同时，中国流行音乐“中国化”特征日益明显。

周杰伦的《双截棍》《龙拳》等歌曲中,在传统流行音乐中加入说唱音乐的元素,把中国民族乐器古朴与西方流行音乐的现代结合起来,使我们耳目一新;陶酷的《Susan说》则把京剧经典唱段《苏三起解》融入节奏感极强的R&B中,使中国国粹与现代流行音乐恰到好处地融为一体【27】。

小结

综上所述,我们可以发现,流行音乐经过了近一个世纪的发展,不同的时代具有不同的特点。但不论什么年代,流行音乐都给我们的生活增添了斑斓的色彩,带来许多乐趣。

九、参考文献

- [1]厉冬冬.多种风格对20世纪三四十年代上海流行音乐的影响与启示[M] 南京师范大学.2011年5月.
- [2]王占军.从新时期流行音乐发展看青年思想文化演变[J]中国青年研究.2009.10.
- [3]钱彤 谢岳.从大众社会心理读解80年代的中国流行音乐[J]大众文艺.2011(18).
- [4]MIDI 百度百科 <http://baike.baidu.com/view/7969.htm>
- [5]Lingand R,Yan D C. Customer RelationShip Management: An Analysis Framework and Implementation Strategies[J]. Journal of Computer Information System,2001(3):82-97.
- [6]音频格式 百度百科<http://baike.baidu.com/view/178954.htm>
- [7]杨军.MIDI消息和标准MIDI文件格式剖析及应用[J].中南民族大学学报(自然科学版),2009,22(Sup):62-64.
- [8]张宁,贾自艳,史忠植.数据仓库中ETL技术的研究[J].计算机工程与应用2002(24),17(8):59-60.
- [9]马建红,王万森.基于数据仓库的保险管理系统的设计与实现[J].微机发展,2004,14(7):66-66.
- [10]梁循.数据挖掘:建模、算法、应用和系统[J].计算机技术与发展,2006,16(1):86-87.
- [11]Embrecht s P , Lindskog F. Modeling dependence with copulas and applicati on to risk management [M] // Rachev S.Handb ook of heavy t ailed dist ributi ons in finance .Amst er dam : Elsevier , 2003 : 329 - 384 .
- [12]朴素贝叶斯分类器 维基百科 <http://zh.wikipedia.org/wiki/%E6%9C%B4%E7%B4%A0%E8%B4%9D%E5%8F%B6%E6%96%AF%E5%88%86%E7%B1%BB%E5%99%A8>
- [13]Gudmund R. St atistics : the conceptual approach [M] . New York : S p ringer 2Verlag , 1997 : 252 - 292 .
- [14]支持向量机维基百科 <http://zh.wikipedia.org/wiki/%E6%94%AF%E6%8C%81%E5%90%91%E9%87%8F%E6%9C%BA>
- [15]金喆.一种基于语谱特征提取的音乐检索系统的设计与实现[M] 哈尔滨工业大学.2011年6月.
- [16]谭瑞东.基于旋律哼唱的音乐检索[J] 大连理工大学.2008.12.
- [17]金毅 黄敏.基于旋律的音乐检索研究——旋律特征的表达和提取[M].现代图书情报技术.2003.4.

- [18] WUMX Wenlong. Design of and Research on High Performance Server Based on Linux [J]. Computer and Communications. 2007, 1:34.
- [19] Tarek Ziadé著, 姚军译. Python 高级编程[M]. 北京: 人民邮电出版社. 2010: 261-285.
- [20] 王远洋, 周渊平, 郭焕丽. Linux 下基于socket 多线程并发通信的实现[J]. 微计算机信息. 2009, 5(3): 70-72.
- [21] 郑继明 魏国华 吴渝. 有效的基于内容的音频特征提取方法[M] 计算机工程与应用. 2009.45(12).
- [22] 杨宗德, 邓玉春. Linux 高级程序设计[M]. 北京: 人民邮电出版社. 2009: 722-745.
- [23] Wesley J.Chun著, CPUG译. Python核心编程[M]. 北京: 人民邮电出版社. 2008: 156-163.
- [24] Richard Stevens. UNIX Network Programming, Vol 1: The Sockets Networking API[M]. 2009: 326-345.
- [25] Z Dong, Z Wei-yun. Analysis and Implementation of Multithread Mechanism under Linux [J]. Journal of Changchun Normal University(Natural Sciences). 2008, 12: 10.
- [26] 林青. 谈当代流行音乐的特点与受众的心理[J]. 大众文艺. 2010(17)
- [27] 叶婷. 中国流行音乐的发展与价值[J]. 深圳大学学报. 2005年7月. 22(4).

十、附录

附录 1

在 Visual Studio 2010 IDE 上利用 C# 从 MIDI 文件中获取音乐各个音轨旋律信息

首先通过制定路径创建一个 FileStream 文件流对象, 再利用 FileStream 对象的 Read 方法实现文件的读取。

```
.....
String fileName=Path.Combine(tempPath,"XXX.mid"); //创建文件路径
Byte[] record=new byte[1000];
FileStream fs=new FileStream(fileName, FileMode.OpenOrCreate,
                             FileAccess.ReadWrite, FileShare.None);
//利用指定的文件路径和文件操作方式创建 FileStream 实例.
fs.Read(record,0,1000);
string strDefault=System.Text.Encoding.ASCII.GetString(record);
....
/*得到了 MIDI 文件的字节信息流后, 为了便于处理, 可以将所得到的 MIDI 信
息字节数组部分内容先转化为 ASCII 码*/
Console.WriteLine(strDefault);
.....
```

接下来根据 MIDI 文件协议, 依次解析 MIDI 文件中各部分的含义, 通过音轨中的各个音符的演奏信息等内容构建出乐曲的特征。

附录 2

朴素 Bayes 算法 Matlab 代码

```
function out=my_bayes(X,Y)
```

%X 为原数据集,Y 是要预测的数据, out 是返回预测的结果

```

%%%%%%%%%%%%%%打开 test.txt 文件
clc;
file = textread('train1.txt','%s','delimiter','\n','whitespace','');
[m,n]=size(file);
for i=1:m
    words=strread(file{i},'%s','delimiter',' ');
    words=words';
    X{i}=words;
end
X=X';%转置
%%%%%%%%%%%%%%打开 predict.txt 文件
file = textread('predict1.txt','%s','delimiter','\n','whitespace','');
[m,n]=size(file);
for i=1:m
    words=strread(file{i},'%s','delimiter',' ');
    words=words';
    Y{i}=words;
end
Y=Y';%转置
%%%%%%%%%%%%%%训练部分
[M,N]=size(X);
[m,n]=size(X{1});
decision=attribute(X,n); %提取决策属性
Pro=probability(decision);%计算决策属性个分量概率
for i=1:n-1
    [post_pro{i},post_name{i}]=post_prob(attribute(X,i),decision); %求各条件属性后验概率
end
%%%%%%%%%%%%%%预测部分
uniq_decis=unique(decision); %求决策属性的类别
P_X=ones(size(uniq_decis,1),1); %初始化决策属性后验概率
[M,N]=size(Y);
k=1;
for i=1:M
    for j=1:n-1
        [temp,loc]=ismember(attribute({Y{i}}),j,unique(attribute(X,j)));%决策属性计算后验概率
        P_X=post_pro{j}(:,loc).*P_X;%各条件属性后验概率之积（Bayes 公式）
    end
    [MAX,I]=max(P_X);%寻找最大值
    out{k}=uniq_decis{I};%哪一类决策属性后验概率最大，则次样本属于那一类
    k=k+1;
    P_X=ones(size(uniq_decis,1),1);%再次初始化决策属性后验概率 P_X，以便为下一样本计算作准备
end

```

```

end
out=out'; %输出结果（转置形式）
%%%%%%%%%%%%%%各子程序
function y=attribute(X,n)
%功能为提取出原数据集 X 中的第 n 个属性所对应的一列值
[M,N]=size(X);
for i=1:M
    temp{i}=X{i}{n}; %将指定列值以 temp 暂量保存
end
y=temp'; %转置
%%%%%%%%%%%%%%
function [post_pro,post_name]=post_prob(E,D)
%E 为目标属性，D 为决策属性，post_pro 计算目标属性对应于决策属性的后验概率
%post_name 为所求的后验概率变量名称
[M,N]=size(D);
decision=unique(D); %决策属性种类
attri=unique(E); %条件属性种类
[m1,n1]=size(decision);
[m2,n2]=size(attri);
temp=cat(2,E,D); %连接条件属性和决策属性
post_pro=zeros(m1,m2); %后验概率初始化
for i=1:M
    for j=1:m2
        for k=1:m1
            post_name{k,j}=cat(2,{attri{j}},{decision{k}});
            if(isequal(temp(i,:),post_name{k,j}))
                post_pro(k,j)=post_pro(k,j)+1; %条件属性后验概率（频数）
            end
        end
    end
end
end
for i=1:m1
    post_pro(i,:)=post_pro(i,:)/sum(post_pro(i,:)); %求得条件属性后验概率
end
%%%%%%%%%%%%%%
function y=probality(E) %计算该属性类的概率
[M,N]=size(E);
class=unique(E); %求该决策属性的类别
[m,n]=size(class);
p=zeros(m,1); %先验概率 p 初始化
for i=1:M
    for j=1:m
        if(isequal(E{i},class{j}))
            p(j)=p(j)+1; %求各个样本的先验概率（频数）
        end
    end
end

```

```
        end
    end
end
y=p/M;%得各样本概率
```

附录 3

%支持向量机分类识别的 MATLAB 代码

% 产生训练样本与测试样本，每一列为一个样本

```
n1 =xlsread('F:\Data\001.xls');
```

```
x1 =xlsread('F:\Data\002.xls');
```

```
n2 =xlsread('F:\Data\003.xls');
```

```
x2 =xlsread('F:\Data\004.xls');
```

```
xn_train = n1;           % 训练样本
```

```
dn_train = x1;           % 训练目标
```

```
xn_test = n2;            % 测试样本
```

```
dn_test = x2;            % 测试目标
```

```
%-----
```

% 参数设置

```
X = xn_train';
```

```
Y = dn_train';
```

```
Xt = xn_test';
```

```
Yt = dn_test';
```

```
type = 'c';
```

```
kernel_type = 'RBF_kernel';
```

```
gam = 2;
```

```
sig2 = 2;
```

```
preprocess = 'preprocess';
```

```
codefct = 'code_MOC';
```

```
[Yc,codebook,old_codebook] = code(Y,codefct);
```

% 交叉验证优化参数

```
%[gam,sig2] = tunelssvm({X,Yc,type,gam,sig2,kernel_type,preprocess})
```

% 训练与测试

```
[alpha,b] = trainlssvm({X,Yc,type,gam,sig2,kernel_type,preprocess});
```

% 训练

```
Yd0=simlssvm({X,Yc,type,gam,sig2,kernel_type,preprocess},{alpha,b},Xt);
```

% 分类

```
Yd = code(Yd0,old_codebook,[],codebook);
```

% 结果统计

```
Result = ~abs(Yd-Yt)           % 正确分类显示为 1
```

```
Percent = sum(Result)/length(Result) % 正确分类率
```

```
%-----
```