基于智能分类算法的音乐和弦识别与分析

摘要：和弦的辨别能力是在音乐学习当中必不可少的技能。但是，为进行过专业训练的人很难快速准确的对一首音乐中的和弦进行快速准确的听辨。随着计算机技术的发展，利用快速傅里叶变换与各种智能算法，本文拟使用上述算法，对音乐的和弦进行识别与分析。拟采用由 Takuya FUJISHIMA 于 1999 年提出的 PCP（Pitch Class Profile）算法为基础，结合 DNN 等智能算法加以改进，增加音乐的节拍跟踪，实现对于音乐中和弦的智能识别系统。

关键词：音乐，和弦识别，智能算法，PCP，节拍跟踪

Identification and Analysis of Musical Chords Based on Intelligent Classification Algorithm

Abstract: The ability to distinguish chords is an essential skill in music learning. However, it is difficult for a trained person to listen quickly and accurately to the chords in a piece of music. With the development of computer technology, we use Fast Fourier transform and various intelligent algorithms to identify and analyze the chords of music. Based on the algorithm of Pitch Class Profile (PCP) proposed by Takuya Fujishima in 1999, and combined with intelligent algorithms such as DNN, this paper proposes to improve it by adding the beat tracking of music and realizing the intelligent recognition system of chords in music.

Key words: Music, Chord recognition, intelligent algorithms, PCP, beat tracking

目 录

1 引 言

自动和弦识别算法是指能够从音频信号中提取和弦信息的计算机程序。和弦是音乐中最基本的元素之一，它能够表达音乐的调性、节奏、风格和情感。自动和弦识别算法的研究具有重要的理论意义和实际价值。从理论上来说，自动和弦识别算法可以揭示音乐的结构和规律，为音乐分析、理解和创作提供有力的工具。从实际上来说，自动和弦识别算法可以应用于多种场景，如音乐教育、音乐制作、音乐检索、音乐推荐等。因此，设计一个高效、准确、鲁棒的自动和弦识别算法是音乐信息检索领域的一个重要课题，也是本文的研究目标。

1999 年由斯坦福大学的 Takuya FUJISHIMA 提出的 PCP（Pitch Class Profile）模型，为后续的所有和弦识别算法提供了理论基础。随着计算机科学与人工智能模型的发展，后续提出了诸多以神经网络为基础的更加优秀的算法，如深度色度提取与谐波-冲击分离算法。

然而，当前的各种研究都将重点集中于旋律组乐器的频谱分析上，忽略了节奏在音乐中的实际作用。本文拟将和弦识别算法与节拍跟踪算法进行结合，拟构建一个更具有乐理性与稳健性的和弦识别与分析系统。同时，本次实验拟提出一个节拍跟踪算法。下文将详细描述本次实验中拟采用的实验方法与实验结果。

2 音乐理论简介

2.1 律法与和弦

2.2.1 律法定义与十二平均律

现代音乐音阶中的音以一定的频率点位进行定义，这种定义方式称作律法。历史中出现过很多不同的律法，例如纯律、十二平均律、五度相生律等[1]。现代乐器调律基本按照十二平均律进行调律，在数学上，十二平均律的计算也更加方便。故本次实验仅探讨在十二平均律下的情况。

十二平均律将两个频率比为 2：1 的两个音称作八度（Octave）。在乐理上，这两个音有着同样的音名。

十二平均律将一个八度内的音平均分为十二份，相邻的音的频率比相同。

十二平均律将标准音定为 440Hz（标准 A），以这个音为标准，计算出其他音的频率。

根据以上特点，可以得出在十二平均律下各个音的计算方法：

将一个八度内的音分为十二份，假设每个音是前一个音的 x 倍，标准音的频率为 F_0，易得方程：

$$x^{12}F_0 = 2F_0 \qquad (2-1)$$

解得

$$x = \sqrt[12]{2} \qquad (2-2)$$

设十二平均律中十二个音的下标分别 F_0、F_1……F_{11},已知 F_x 的频率，由上述公式易得求 x 的公式为：

$$x = 12\log_2(F_x/F_0) \qquad (2-3)$$

2.2.2 音阶与功能和弦

上文已经提到了在十二平均律中各个音频率的算法。在音乐理论中，一个八度内的十二个音构成的音阶称之为半音阶，每个音分别被记作：

C、#C/bD、D、#D/bE、E、F、#F/bG、G、#G/bA、A、#A/bB、B.

再以八度循环。

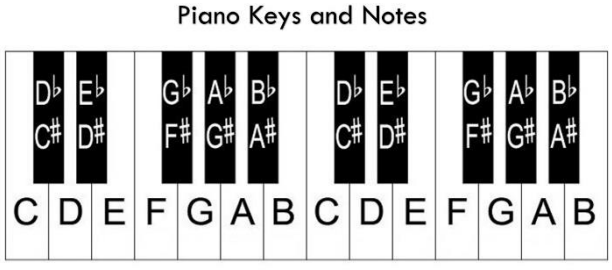


图 2.1 钢琴键盘

十二平均律中的某几个音的组合，组成调式音阶。

在音乐中存在许多各式各样的音阶，如大小调音阶、中古调式音阶、中国五声音阶、日本音阶等。但是在本项目中，我们着重探讨常用的自然大小调音阶的情况。

自然大调的一个八度由七个音组成，音阶之间按照全音、全音、半音、全音、全音、全音、半音的音程关系组成。以 C 大调为例，此音阶由 C、D、E、F、G、A、B 七个音组成。C 称为音阶的主音（Tonic）。

自然小调与自然大调相似，以半音、半音、全音、全音、半音、全音、全音的音程关系构成。以 a 自然小调为例，构成音阶的音为 a、b、c、d、e、f、g。

以其中的一个音为根音，向上三度与五度构成三和弦，称为调内和弦。以自然大调音阶中的 C 大三和弦为例，由 C、E、G 三个音组成，以主音 C 为根音的和弦被定义为主和弦。同理，音阶中的每一个音都可以作为根音构成和弦，分别记做 C、Dm、Em、F、G、Am、Bdim。音阶中的音构成的和弦共同组成功能和弦组。和弦按照一定的乐理规律进行连接，构成和弦进行。

和弦的连接一般遵从和弦功能。最简单的和弦进行及其逻辑基础是：在主和弦之后引入一个或者几个不稳定的和弦，形成明显的紧张性，这种紧张性在进行到或者回到主和弦时得到解决[2]：

T（稳定）-> 非 T（不稳定，紧张）-> T（稳定，紧张的解决）

下面给出一个具体的例子：

以 C 自然大调为例，从主和弦开始，在主和弦之后加入几个不稳定的和弦：

C（主和弦，稳定）->G（属和弦，不稳定）-> C（主和弦，稳定）

这个进行在古典和声学中被成为正格进行。它揭示了和弦进行中一个最本质的规律：构建紧张感，然后进行解决。在实际的音乐创作中，和弦的进行并不会像这样简单，其中将涉及到附属和弦、同功能组和弦替代等手法，但是这样普遍的规律是后续进行和弦识别与预测的重要参考[3,4]。

以十二平均律中不同的音作为主音，可以按照相同的规律推出所有十二个自然大调音阶与自然小调音阶。不同调式的识别是后续对和弦识别的重要前提条件。

2.2 节拍与节奏

当前对计算机算法实现和弦的算法，大多集中于对音频信号的频率进行分析，而忽略了节拍在音乐中发挥的重要作用。在本次实验中，拟将节拍在时间上的特征纳入和弦识别与分析的考量范围之内。在本次实验中，拟提出一个对音乐信号中的节拍进行检测

的算法。

2.2.1 节拍、小节、曲速

拍是音乐中组织音符的基本时间单位，所有的音符都基于这一个具有周期性的时间概念来组织。这个概念可以类比计算机科学中“时钟”的概念，它为计算机中所有的操作提供同步的标准。

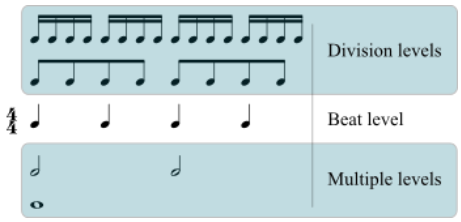


图 2.2 音符在小节中的组织方式

节拍层（Beat level）：指示乐段中节拍的概念，在常见的 4/4 拍音乐中，通常以四分音符为一拍，每小节的第一拍为重拍，第三拍为次重拍，二、四拍为轻拍。

分拍层（Division levels）：将四分音符进行二次、四次或者更加多次的均分，构成分拍层。换句话说，分拍层由八分音符、十六分音符等组成。

合拍层（Multiple levels）：节拍层的音符的组合称为合拍层。

在本实验中，将分拍层、合拍层的音符与节拍层的音符进行区分，尽可能减小分拍层与合拍层的音符对节拍层音符的判断是本次实验的关键所在。

拍按照一定的数量与形式组织成小节（Measure），组织方式通常以拍号（Signature）表示。以现代歌曲中常见的 4/4 拍为例子，它表示以四分音符为一拍，每小节由 4 拍构成。小节是乐句的基本组成。每小节的第一拍被称为下拍（Downbeat）。在本次实验中，下拍时间戳是判断和弦变换点位的主要参考标准。

曲速是指每分钟的拍数（Beats per Minute，BPM），这是衡量音乐快慢的量化数据。在本次实验中，将提出一种检测歌曲曲速的算法。

2.2.2 音头、音乐中节拍组织的特点

音头，是指一个音符开始发出声音的瞬间。在实际演奏中，表现为按下琴键（钢琴）的瞬间，或者鼓棒接触到鼓面的瞬间（打击乐器），也就是在谱面上显示出的音符的位

置。上文提到，音符一般按照节拍为时间基础进行组织，也就是说，音头一般出现在拍上或者分拍层（Division Levels）上。

由于音头是一个音符开始发出声音的瞬间，而一个音符的延音，响度一般会缓慢衰减，即声音的能量会逐渐衰减。故我们可以对频谱的能量在时间维度上的变化来观察音头的大致位置，即能量的极大值点可以认为是一个可能的音头。

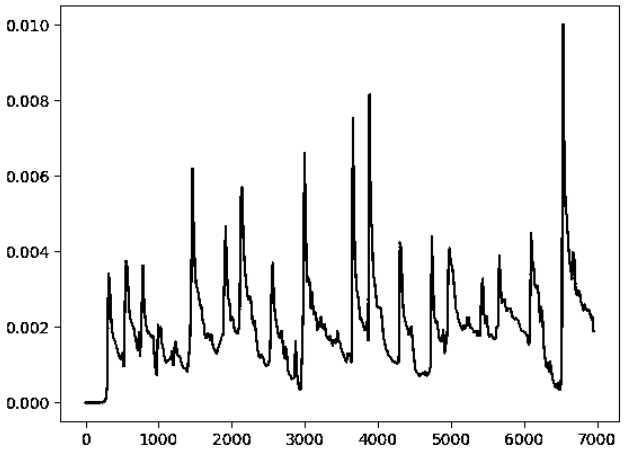


图 2.3　一个音乐片段随时间的响度变化

图 2.3 显示了一个音乐片段的响度变化。可以看出，音频的响度符合上述音头的响度衰减特征。在本次实验中，拟提出一个算法，通过音频的周期性的响度，检测音频的 BPM。

3 系统设计概述

本章将对本系统的设计思路进行简述

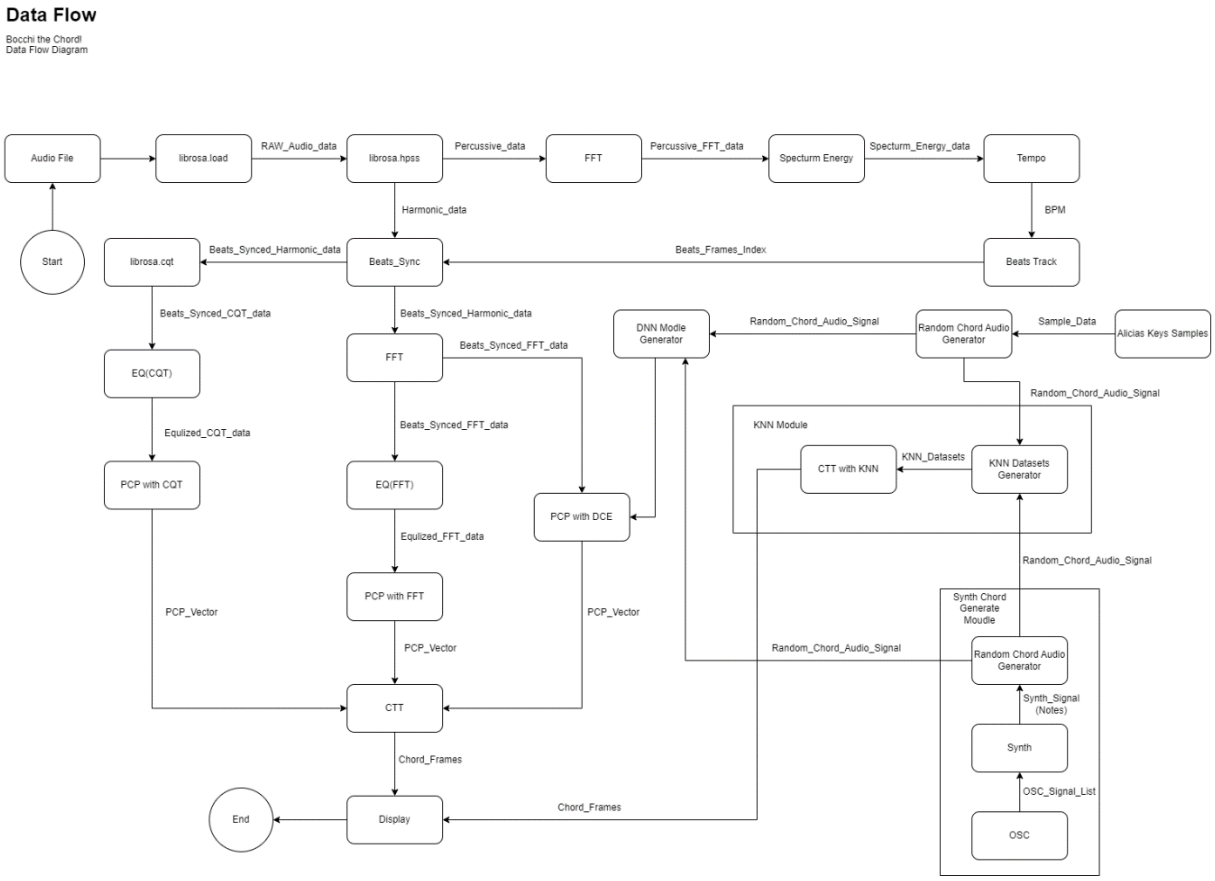


图 3.1 系统的模块设计与数据流向概述

3.1 谐波-冲击分离（Harmonic-Percussive Source Separation，HPSS）算法

2008 年由东京大学的 Nobutaka Ono 等人提出的谐波-冲击分离（Harmonic-percussive source separation）算法利用支持向量机（SVM）实现了节奏组乐器与旋律组乐器的频谱分离。这个算法通常被用作色度提取的数据预处理环节。[10] 本次实验也将尝试利用此算法对音频信号进行预处理，同时，也将此算法运用于节拍跟踪算法，将谐波-冲击分离算法与均衡器函数结合，有望提高系统的稳健性。

3.2 周期乘数法

本节拟提出一个对音乐进行 BPM 测算的方法：周期乘数法：

上文提到，音乐的音频采样信号强度跟随音头的变化，而音头的时间随着音乐的节拍周期性出现。根据此原理，可以检测音乐响度强度的周期性变化，求得音乐的 BPM。

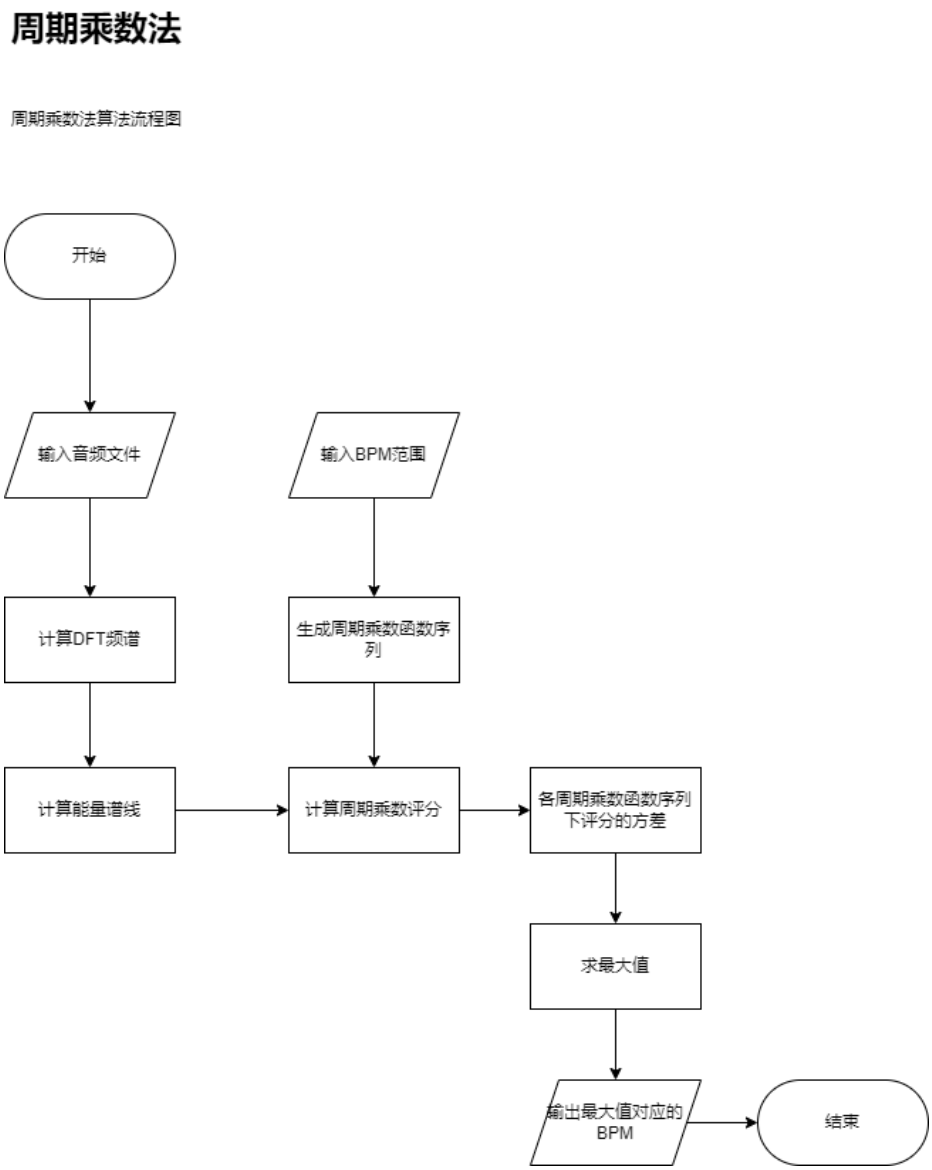


图 3.2 周期乘数法的算法流程

下面给出周期乘数法的具体实现：

拟构建一个简单周期乘数函数：

$$T(t) = cos\left(\frac{2\pi \cdot BPM}{6000(ms)} \cdot t\right) \qquad (3-1)$$

音频采样的响度系数（能量谱线）

$$E(n) = \sum_{l=0}^{\frac{N}{2}-1} W(l) \cdot |X(l)| \qquad (3-2)$$

其中$W(l)$为一个均衡器（Equalizer，EQ）函数，它的数学意义是 DFT 变换结果的频率点位在能量计算中的权重，物理意义是能量谱线对某个频率点位的敏感程度。在实际音乐中，节奏组乐器（打击乐器）对节奏的指示性相比于旋律组乐器更加显著。而节奏组乐器的频率特征与旋律组乐器有着明显的区别：

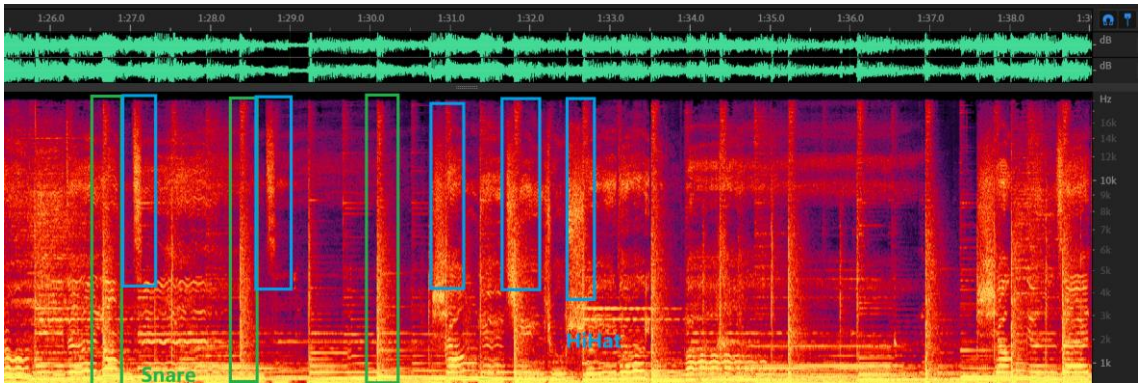


图 3.3 一个典型的节奏组乐器的频率响应特征

上图显示了一个典型的节奏组乐器（架子鼓）中各乐器的频率响应特征：

手击镲（HI Hat）：图中蓝色框部分：集中在高频部分，中心频率 10KHz，频率范围从 5KHz 至 20KHz。

小军鼓（Snare）：图中绿色框部分，在中频与高频部分均对频谱有着较大的冲击。

合理设计均衡器函数，可以使得能量谱线对节奏组乐器的频率响应更为敏感，提高识别的准确率。

计算待测试乐段能量谱线$E(n)$在当前预测 BPM 下的评分：

$$Score_{predict} = \frac{1}{F} \sum_{n=0}^{F} E(n) \cdot T\big(time(n)\big) \qquad (3-3)$$

其中F为乐段的帧总数。$E(n)\cdot T(n)$即为周期乘数。

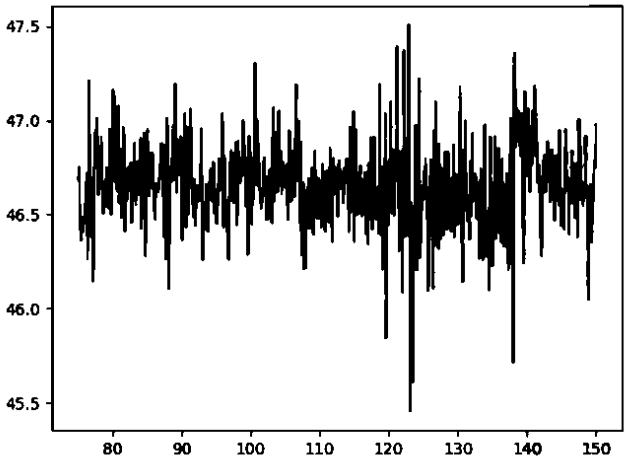


图 3.4 一个音乐片段在各 BPM 预测中的得分

由于在实际情况中，一段音乐并不是总严格按照节拍开始，也就是说，周期乘数函数的相位与音频采样的相位存在相位差，且相位差会对最后的评分结果产生不可忽略的误差。然而，忽略相位差的影响，一个周期与音频采样的节拍周期有着显著区别的周期乘数函数，预测 BPM 的评分接近于一个平均、无序的值，而周期与音频采样节拍相近或相等的周期乘数函数，预测得分总是明显大于或小于那个平均、无序的值（类比于物理中“共振”的现象），故在此取各 BPM 下的预测评分结果的方差，来反映当前音频在各预测 BPM 下的预测得分。

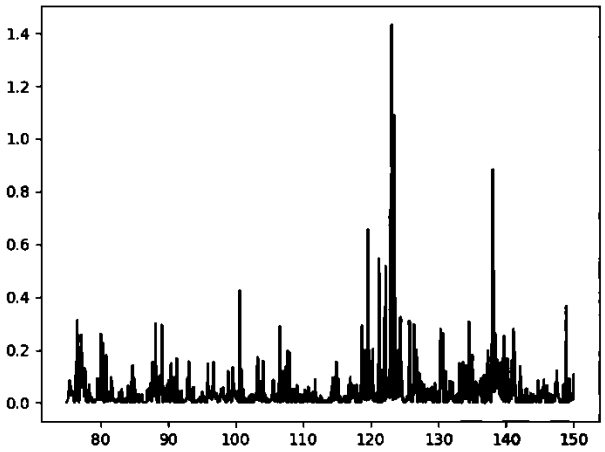


图 3.5 取方差后各预测 BPM 下的评分

在后续的实验结果当中，此方法取得了基本准确的预测结果。但是存在着计算量大、处理时间过长等问题。

3.3 DFT 离散傅里叶变换与频域谱平滑处理

3.3.1 DFT 离散傅里叶变换

傅里叶变换在信号处理中的作用，一般是将一个时域函数转换为频域函数。一个周期性连续函数的傅里叶函数的复数形式如下所示[5]：

$$c_n = \frac{1}{2l}\int_{-l}^{l} f(x)e^{-\frac{n\pi x}{l}i}dx \qquad (3-4)$$

在现实生活中，声音由物体的振动产生。一个质点的简单受迫振动方程一般可以由一个连续的周期性函数表示。但是在数字音频领域中，一个连续的音频信号需要进行采样，转换成一个离散的函数进行储存。对音频的采样速率使用采样率表示，单位为 Hz。根据奈奎斯特-香浓采样定理，如果周期函数 x(t) 不包含高于 B cps（次/秒）的频率，那么，一系列小于 1/(2B) 秒的 x(t)函数值将会受到前一个周期的 x(t)函数值影响。

因此 2B 样本/秒或更高的采样频率将能使函数不受干扰。相对的，对于一个给定的采样频率 fs，完全重构的频带限制为 B ≤ fs/2。为了平衡计算量与信号失真，本次实验拟将音频的采样率统一定为 44100Hz。

由于在计算机中存储的音频信号为离散形式，故对其进行的傅里叶变换也应该是离散的。下面给出离散傅里叶变化的数学表达式：

$$X(k)=\sum_{n=0}^{N-1}W(n)x(n)e^{-\frac{2\pi ikn}{N}} \qquad (3-5)$$

其中，数组$X(k)$为 DFT 频谱，下标 k 为频率点位, $W(k)$为窗口函数。N 为 FFT 大小，$x(n)$为输入信号。在$x(n)$为实数的情况下，由于每个频率点位存在一对共轭的复数结果，$X(0)$至$X(N/2-1)$的结果即为完整的 DFT 频谱。由于仅对信号在各个频率点位的强度（振幅）进行考察，故 DFT 频谱的结果应对复数结果取模。

在对音乐信号进行处理时，通常不会将整个采样进行 DFT 处理。一般将信号切分成音频帧之后，在进行处理。一帧的长度即为 FFT 大小。为了保证每一个音频帧之间的连续性，分帧时通常会确保两帧之间存在交叉。同时，为了防止分帧之后产生能量泄露，通常对信号进行加窗。窗口函数是一个以 FFT 窗口大小为长度的函数，以本次实验中所使用的汉宁（Hanning）窗为例：

$$W(n)=\frac{1}{2}\left[1+\cos\left(2\pi\cdot\frac{n}{N-1}\right)\right] \qquad (3-6)$$

汉宁（Hanning）窗可以看成是升余弦窗的一个特例，汉宁窗可以看作是 3 个矩形时间窗的频谱之和，或者说是 3 个 sinc（t）型函数之和，而括号中的两项相对于第一个谱窗向左、右各移动了 π /T，从而使旁瓣互相抵消，消去高频干扰和漏能。

3.3.2 频域谱平滑处理

一般来说，我们普遍认为，一般的音乐中的和弦转换与音乐节拍强相关。也就是说，通常认为和弦在拍点时变化，故一拍的时间内，所有的音频帧都包含本小节内和弦的所有信息。以拍同步的方式，将一拍内的所有音频帧的频域谱信息进行平滑处理，可以显著提高信息的利用效率，一个最简单的平滑方式，就是用平均值，将一拍时间内的所有帧的频域谱平滑为一帧：

$$X[k]=\frac{1}{F_n}\sum_{n=0}^{F_n}x_n[k] \qquad (3-7)$$

其中$X[k]$为平滑后的帧，F_n为一拍中音频帧的总数，$x_n[k]$为一拍中第n帧的频域谱。

在接下来的实验中，所使用的音频帧，都经过拍同步的频域谱平滑处理。

3.4 音级色谱图（Pitch Class Profile，PCP）与和弦类型模式（Chord Type Templates，CTT）

3.4.1 PCP 的求解流程

PCP（Pitch Class Profile）模型由斯坦福大学的 Takuya FUJISHIMA 于 1999 年提出。这是一种根据律法原理对音频信号进行简单频率分析的方法。它是一个十二维的向量，每个维度表示当前音频帧在十二平均律中每个音的强度。[7]

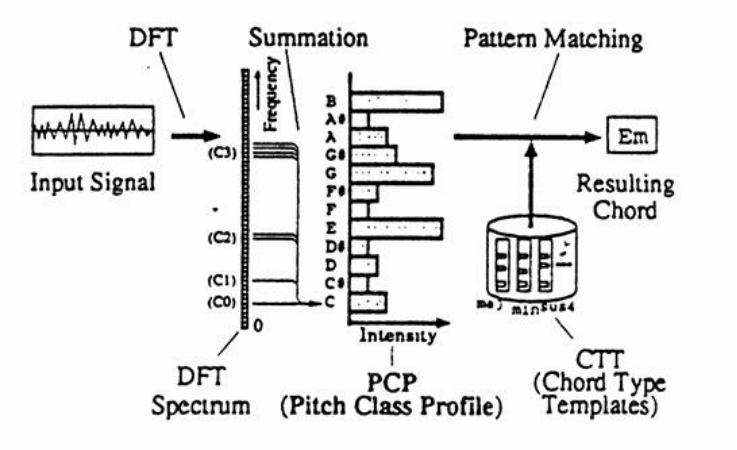


图 3.6 使用 PCP 进行和弦识别的求解流程

下面给出 PCP 向量的求解流程：

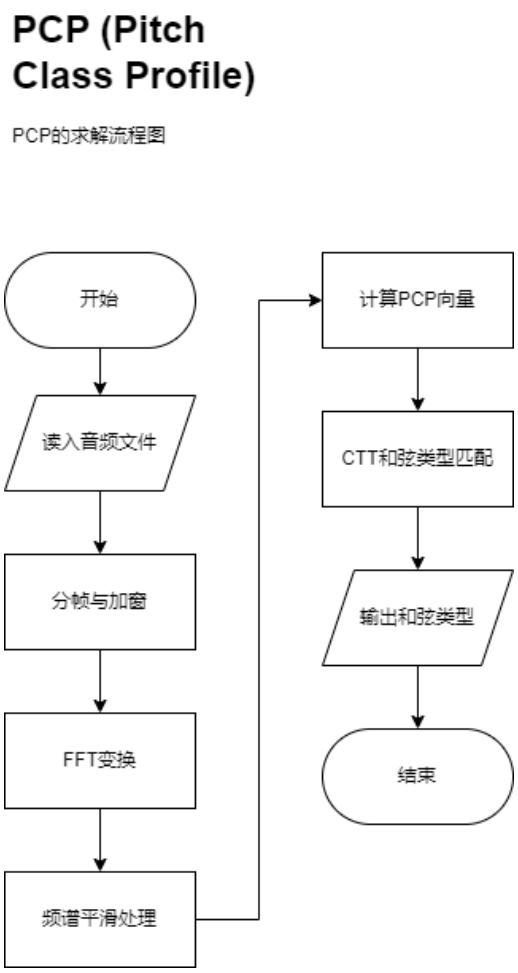


图 3.7 PCP 向量的计算与和弦类型匹配流程

在得到了 DFT 频谱$X(k)$之后，可以对音频的 PCP 向量进行计算：
定义 PCP 向量

$$PCP(p) = \sum_{1\ s.t.M(l)=p} ||X(l)||^2 \qquad (3-8)$$

其中，$M(l)$为一个将频率点位映射至 PCP 下标的矩阵，其定于如下：

$$M(l) = \begin{cases} -1 & l = 0 \\ round(12\log_2((f_a \cdot \frac{1}{N})/f_{ref}) Mod 12 & l = 1,2,3,\dots,N/2-1 \end{cases} \qquad (3-9)$$

其中，f_{ref}为标准音频率，记为$PCP(0)$. $f_a \cdot \frac{1}{N}$为$X(l)$的频率。

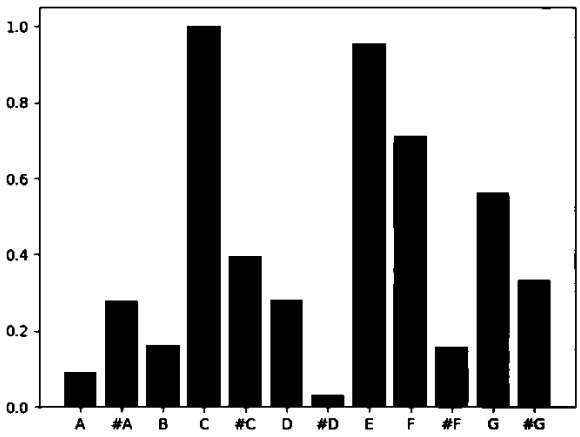


图 3.8 一个 C 大三和弦的 PCP 向量

PCP 向量用于进行模式匹配。和弦类型模板（Chord Type Templates，CTT）为一个十二维的向量，用于判断和弦的种类。若一个和弦是以 C 为根音的和弦，则

$$CTT_c(p) = 1$$

否则为 0.

一个 M7（大七和弦）和弦的 CTT 向量如下所示：

$$PCP_{CM7}(p) = (1,0,0,0,1,0,0,1,0,0,0,1)$$

最邻近算法下 PCP 向量在一个 CTT 中的预测得分为：

$$Score_{nearest,c} = \sum_{p=0}^{11} \big(T_c(p) - PCP(p)\big)^2 \qquad (3-10)$$

加权求和法下 PCP 向量在一个 CTT 中的预测得分为：

$$Score_{weighted,c} = \sum_{p=0}^{11} W_c(p) \cdot PCP(p) \qquad (3-11)$$

理论上来说，加权求和法在实际的和弦检测算法中能得到更好的结果。但是，如何确定每一个和弦类型的权重向量是一个复杂的问题。用现在的眼光来看，确定一个和弦

类型的权重向量再进行加权求和，非常类似于深度神经网络的一个层级的构造。故下文将会探讨深度神经网络在和弦类型匹配工作中的表现。

3.4.2 PCP 的严重不足与改进方案

未经改进的 PCP 算法存在着一个致命的缺陷：由于使用 DFT 对信号进行频域的处理，而 DFT 对频域数据采用线性标度，这种特性与人耳的听觉特性极度不符合。通常情况下，人耳在低频区域对频率的灵敏度更高。这一点在各种律法的定义中也能得到很好的体现：

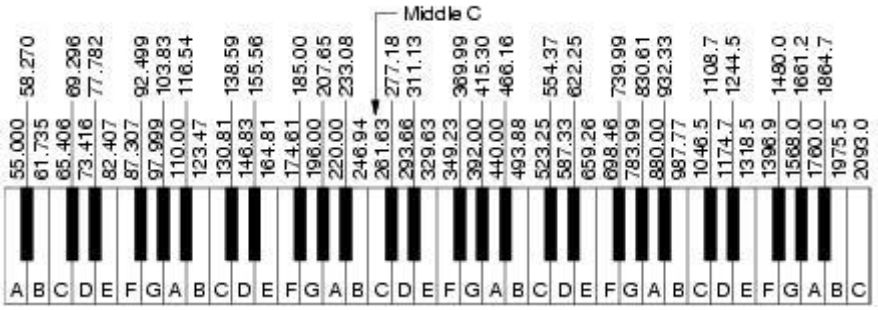


图 3.9 在十二平均律下各个部分音符的频率（A=440Hz）

以标准 A（440Hz）的频率举例，标准 A 的下半音（#G）的频率为 415.30Hz，上半音（#A）为 466.16Hz，在采样频率为 44100Hz、FFT 大小为 1024 的情况下，440Hz 周围的频率点位为 387.60Hz（第 9 点位）、430.66Hz（第 10 点位）、473.73Hz（第 11 点位）。

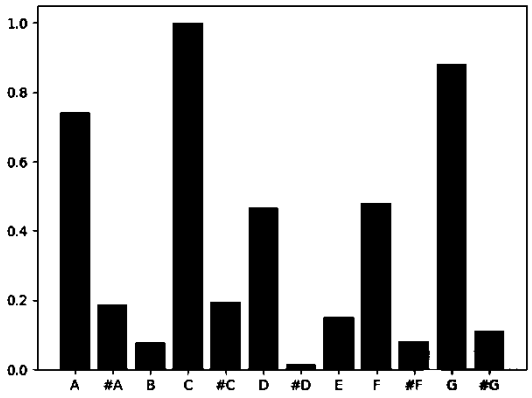


图 3.10 更低一个八度的 C 大三和弦的 PCP 向量

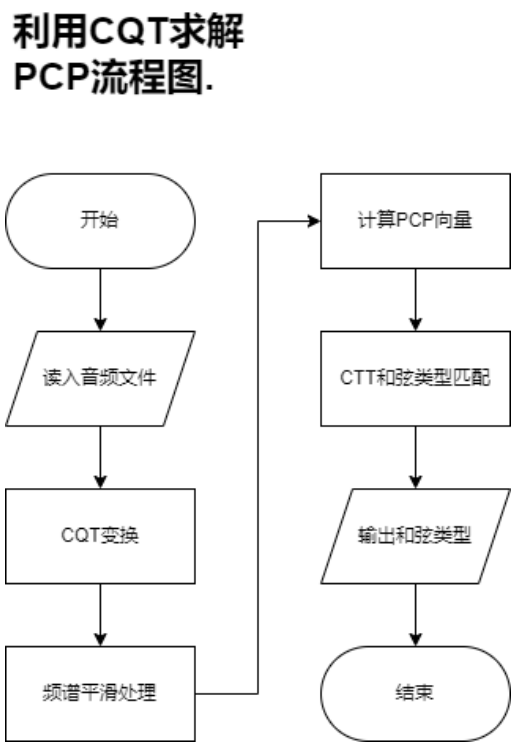
据图 3.10 所示，在钢琴键盘的低音区域，DFT 的分辨率显然达不到需求。CMaj 和弦（C、E、G）中，三度音 E 的大小远小于一个和弦外音 A 的强度。这是由于 DFT 在

低频部分分辨率严重不足导致音高的计算误差所引起的。考虑增加一个高通滤波器，将考察的重点频率放在高频区域上。但是，一段音乐中的低频部分通常包含大量信息，简单将其滤除将会导致信号的丢失。

同时，由于音乐中的节奏组乐器的影响，一个 PCP 帧内会受到节奏组乐器的噪音干扰。传统的方法是使用起音（Attack）检测[7]。在本次实验中，将会尝试使用 HPSS（冲击-谐波分离）来代替传统的起音检测。

下一节介绍一个更加适合于音乐音频分析的频域转换算法：CQT（Constant Q Transform，常量 Q 宽带变换）。

3.5 CQT（Constant Q Transform，常量 Q 宽带变换）：



3.11 利用 CQT 进行 PCP 的求解与和弦匹配

上文提到，在以 DFT 为频域转换方法的 PCP 中，存在着低频区域分辨率严重不足的问题，这是由于 DFT 的数学特性所决定。

在 DFT 中，由于窗口长度固定，DFT 结果中每个滤波器的频率差都是定值，为：

$$f_{res} = \frac{f_S}{N} \qquad (3-12)$$

其中，f_{res}是相邻滤波器之间的中心频率之差（滤波器的宽度），也就是频率的“分辨率”。f_S为音频采样的采样率，N为窗口大小，或者说 FFT 大小。

再考虑到在十二平均律中乐音的频率的排列方式：

在低音区域，两个半音的频率相差非常小，如 A1=55.0Hz，#A1=58.27Hz。

而在高音区域，两个半音之间的频率相差相对较大，如 A6=1670.0Hz，#A6=1864.66Hz。

在传统的 DFT 中，由于窗口长度固定，无论在低频区域还是在高频区域，频率分辨率都是一致的。这就导致了频域谱在低频的分辨率严重不足，而在高频区的分辨率过高，导致音高识别产生误差。若要提高在低频区域的分辨率，就要提高 FFT 的窗口大小。但是仅仅提高窗口大小，会导致频域谱在时间轴上的分辨率的降低（每个 FFT 帧的长度增加了）。在后续的数据分析过程中，显然会降低数据的准确性。

再考虑到十二平均律中各个音符的频率特征：相邻音符的比值是一个固定值，也就是说，音高与频率的关系可以被抽象的理解为一个指数函数，在音域较高时，相邻两个音之间的频率只差较大，而在低音域，相邻两个音之间的频率相差较小。

根据律法的特点，可以考虑设计一个新的频域转换函数，令相邻两个滤波器之间的中心频率之差随着频率的提高动态变化。这样既能在低频区域获得较高的分辨率，也能在高频区域适当减小分辨率，减小在进行频率与音高之间进行转化时的误差。

这种算法就是下文将要提到的 CQT（Constant Q Transform，常量 Q 宽带变换）。

CQT 算法是 DFT 的一个改进形式，相比于 DFT 的线性标度，它使用对数标度对音频进行频域转换，这使得它更加适合于音乐信号的分析与处理。[11]

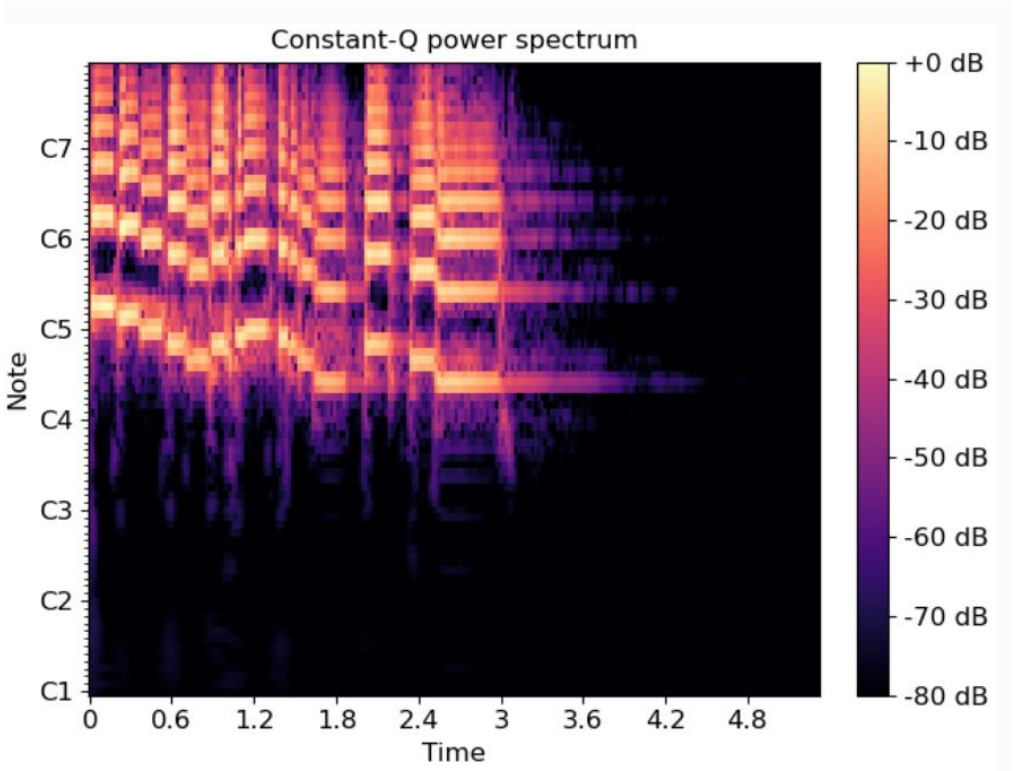


图 3.12 一段音乐的 CQT 频谱，使用 LibROSA 绘制

本质上，CQT 是一系列在不同频率上的滤波器。已知在十二平均律中，相邻的两个音之间的频率之比为一个定值，且两个相同音名且相差八度的音的频率比为 2 倍。根据此规则，设计一个滤波器序列：第 k 个滤波器的频谱宽度δf_k等于前一个滤波器宽度的倍数：

$$\delta f_k = 2^{1/n} \cdot \delta f_{k-1} = \left(2^{1/n}\right)^k \cdot \delta f_{\min} \qquad (3-13)$$

其中 δ fk 是第 k 个滤波器的带宽，fmin 是最低滤波器的中心频率，n 是每个倍频程的滤波器数。

上文已经提到，在 DFT 中，滤波器的带宽为：

$$f_{res} = \frac{f_S}{N} \qquad (3-14)$$

要改变每个滤波器的带宽，只需要改变 DFT 的窗口大小。

可以将 CQT 理解为一个动态窗口大小的 FFT。上文已知一个音频帧的傅里叶变换如下所示：

$$X(k)=\sum_{n=0}^{N-1}W(k)x(n)e^{-\frac{2\pi ikn}{N}} \tag{3-5}$$

定义滤波器宽度δf_k，Q 因子

$$Q=\frac{f_k}{\delta fk} \tag{3-15}$$

第 k 个窗口的窗口长度：

$$N[k]=\frac{f_s}{\delta f_k}=\frac{f_s}{f_k}\cdot Q \tag{3-16}$$

于是，每个频率点位由上述短时傅里叶变换中的

$$\frac{2\pi k}{N}$$

变为 CQT 变换中的

$$\frac{2\pi Q}{N(k)}$$

同时，窗口函数也变化为：

$$W(k,n)=\frac{1}{2}\left[1+\cos\left(2\pi\cdot\frac{n}{N(k)-1}\right)\right] \tag{3-17}$$

综合上述式子，将短时傅里叶变换的结果转换为 CQT 结果的公式成为：

$$X(k)=\frac{1}{N(k)}\sum_{n=0}^{N(k)-1}W(k,n)x(n)e^{-\frac{2i\pi Qn}{N(k)}} \tag{3-18}$$

在这里，求和之后要乘上$1/(N(k))$的原因是由于窗口长度发生了改变，要对每一个窗口的结果进行归一化。否则长度更长的滤波器将会给出更大的乘积。

经过这样的变换，传统的离散傅里叶变换中因为频率线性标度产生的低频区域分辨

率不足的问题，可以由 CQT 变换的对数标度的滤波器组解决。运用 CQT 变换代替 DFT 变换求解 PCP，将在低频区域得到更高的分辨率。

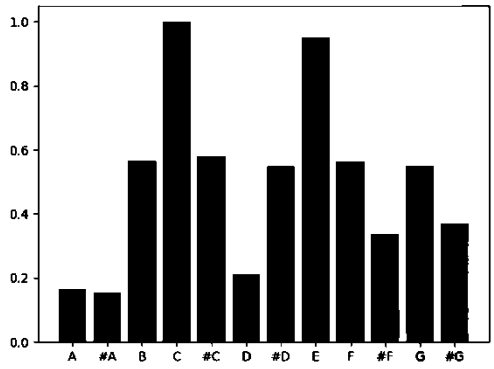


图 3.13 利用 CQT 算法得到的 PCP 向量

从上图中可以看出，CQT 相较于 DFT，在同一个和弦中（C 大三和弦），三度音（E）的识别准确率要更加准确。总体上来说，虽然 CQT 计算出的 PCP 也存在着较为显著的误差，但是相比于 DFT 产生的误差，CQT 产生的误差大多是由于半音频率泄露所产生。也就是说，产生的误差大多位于目标音高的上半音与下班音。这种误差在后续的数据处理上更容易辨别，在 CTT 评分中，半音的误差相对于评分系统的影响，对比与 DFT 的在更大的音程上产生的误差更小，对和弦识别的准确率也更低。

3.6 KNN 聚类

KNN（K-Nearest Neighbor，K-最临近）算法是一种基本分类与回归方法。在本项目中，我们需要对一个音频片段的频率响应特征进行聚类，判断它属于哪个和弦类型，以及由哪一个音作为根音所构成。KNN 非常适合执行此类任务。

KNN 利用训练数据对特征向量空间进行划分，然后将划分后的空间标记为不同的类别。

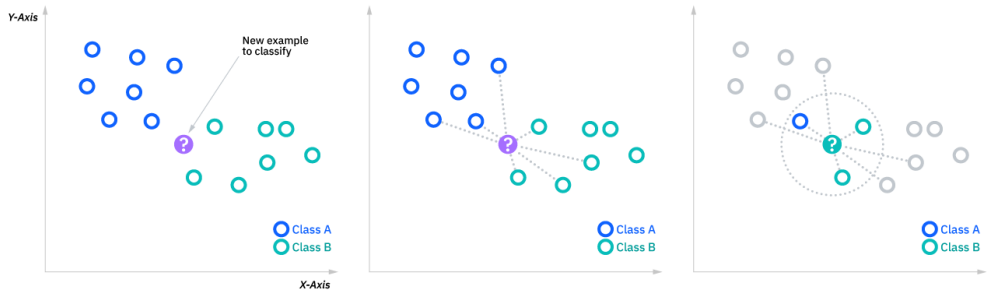


图 3.14 KNN 算法概述图，来自 IBM

对于实例 X，有给定的训练数据集：

$$T = \{(x_1, y_1), (x_z, y_2), \cdots, (x_N, y_N)\}$$

其中：

$$x_i \in x \subseteq R^n$$

是实例X_i的 n 维的特征向量。

$$y_i \in Y = \{c_1, c_2, \cdots, C_k\}$$

为实例的类别。

根据给定的训练数据集，预测实例 X 在 Y 中的类别。

KNN 的执行步骤如下：

1.确定距离测度方法。闵可夫斯基距离定义如下：

$$d_{xy} = p\sqrt{\sum_{k=1}^{n}(x_k - y_k)^P} \qquad (3-19)$$

闵可夫斯基距离是一类距离的定义。其中，p 是可变参数。当 p=1 时，称为曼哈顿距离，p=2 时，称为欧氏距离，当 p 趋近于无穷时，称为切比雪夫距离。

在实际使用时，欧氏距离（L2 范数）是最常用的距离测度方法。根据以上定义，欧氏距离的定义如下：

$$d_{xy} = \sqrt{\sum_{k=1}^{n}(x_k - y_k)^2} \qquad (3-20)$$

2.计算实例 X 在给定距离测度方式下距离训练集 T 中所有的实例 X_i 的距离，并且

将所有距离进行排序

3.确定 K 值。在排序完成后的数据中选出 K 个数据，根据多数投票原则对实例 X 进行分类。将实例 X 归于离 X 最近的 K 个数据中最多的那个分类的类别。

设 CQT 滤波器的最低中心频率F_{min}=32.70Hz(C0)，每个倍频程中的滤波器个数$n = 12$（半音），考虑 7 个八度音程内的频率范围，得到一个音频帧的 CQT 频谱：

$$CQT[k] = \frac{1}{N(k)} \sum_{n=0}^{N[k]-1} W[k,n]x[n]e^{-\frac{2i\pi Qn}{N[k]}} \qquad (3-21)$$

其中：

$$k \in [0,83], k \in \mathbb{Z}$$

CQT 频谱的结果是一个 84 维的向量，向量的每一个维度代表着待测试的音频帧在十二平均律中不同八度内各个音符的强度。将 CQT 结果作为训练集实例X_i与待聚类实例X的特征向量。计算待聚类实例X与训练集样本实例X_i的欧氏距离：

$$distance_{(X,X_i)} = \sqrt{\sum_{k=0}^{83}(CQT_{Xi}[k] - CQT_X[k])^2} \qquad (3-32)$$

再按照上文提到的 KNN 聚类方法对待聚类实例X进行评判。

3.7 深度色度提取（Deep Chroma Extract）

色度（Chroma）与上文提到的 PCP 向量类似，它是一个包含时间序列的色度向量，代表音频中特定时间的谐波内容[8]。本次实验中，拟利用一个 DNN 深度学习网络[6]，来对音频中的色度信息进行提取。

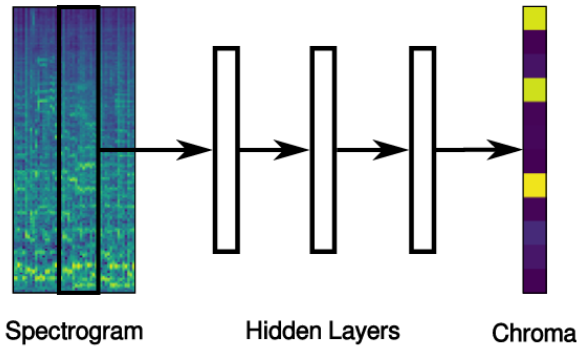


图 3.15 深度色度提取的概述图[7]

DNN 是一个包含了 L 个隐藏层，每个隐藏层h_l包含了U_l个计算单元。每个计算单元的输入取决于上一个隐藏层中每个计算单元的输出：

$$h_l(x) = \sigma_l(W_l \cdot n_{l-1}(x) + b_l) \qquad (3-33)$$

其中，x是计算单元的输入，$w_l \in \mathbb{R}^{U_L \times U_{L-1}}$是计算单元链接的权重，$b_l \in \mathbb{R}^{U_l}$是计算单元的偏置。$\sigma_l$是一个通常为非线性的激活函数。

本次实验中，模型的输入是一个包含时间序列的 CQT 频谱矩阵 S 中的 n 个连续帧，这是为了平滑频谱中噪音对输出结果产生的影响。输出结果为一个包含时间序列的色度向量。

为了训练模型，本实验采用 Alicias Keys 钢琴采样，生成位于 7 个八度中由 12 个音作为根音所构成的 5 种不同类型的和弦（大小三和弦、大小七和弦以及属七和弦）。

采用二元交叉熵定义模型的损失函数：

$$\mathcal{L} = \frac{1}{12}\sum_{i=1}^{12} -t_i\, log(p_i) - (1-t_i)\, log(1-p_i) \qquad (3-34)$$

p为模型的预测输出，t为目标输出。

深度色度提取相当于将 CQT 频域谱进行重新聚类，进一步提高音符预测的准确性。

4 实验结果与评价

本次实验提出的周期乘数法在常规的流行音乐的 BPM 测算之中取得了很好的鲁棒性。本次实验选取了 15 首不同风格、语言与曲速的歌曲进行测试，结果的准确率基本

达到 100%。

ID	Song	BPM	BPM_Predict	ERROR	Precise	Genre
0	星座になれたら-結束バンド	123	123	0	TRUE	J-Rock
1	I Really Want to Stay At Your House-Rosa Walton	125	125	0	TRUE	Soundtrack
2	告白-王欣宇	68	68	0	TRUE	Mando-Pop
3	Tamaki-RADWIMPS	70	70	0	TRUE	Soundtrack
4	打上花火-Daoko; 米津玄師;	96	96	0	TRUE	Soundtrack
5	言って。-ヨルシカ;	180	180	0	TRUE	J-Rock
6	流光记-银临;	112	112	0	TRUE	Mando-Pop
7	Re:make-ONE OK ROCK	172	172	0	TRUE	J-Rock
8	恋爱困难少女-ChiliChill	96	96	0	TRUE	Mando-Pop
9	预言-打扰一下乐团	92	92	0	TRUE	Mando-Rock
10	春を告げる-yama	120	120	0	TRUE	J-Pop
11	Futurepop-ANK	106	106	0	TRUE	EDM-Futurehouse
12	星球上的追溯诗-熊子; 味素;	130	130	0	TRUE	EDM-Futurehouse
13	想去海边-夏日入侵企划	130	130	0	TRUE	Mando-Rock
14	永久の宴 - Aiobahn,YUC'e	127	127	0	TRUE	J-Pop
15	红昭愿 - 音阙诗听	115	115	0	TRUE	Mando-Pop

图 4.1 周期乘数法检测歌曲 BPM 的实验结果

但是本算法也存在着显著的问题：首先，本算法无法对变速的音乐进行 BPM 的测算。其次，由于设计到较多次数的迭代操作，算法的运行速度较慢。可以考虑与 Oneset 检测相结合，改善算法的运行速度。

对于基于 DFT 的 PCP 向量计算，由于乐器泛音的影响与低频区域分辨率严重不足的问题，会产生较为严重的误差，影响后续的和弦匹配结果。为了解决 DFT 在低频区域的分辨率不足的问题，引入了 CQT 算法，将 DFT 中的线性标度改进为对数标度，使得频域谱的结果更加适合于音乐信号的分析。

由 CQT 生成的 PCP 向量在加权求和法中对和弦的聚类产生了较高的准确性。然而，在 KNN 聚类中，CQT 产生的半音泄露问题却对模型的准确率产生了较为显著的影响。

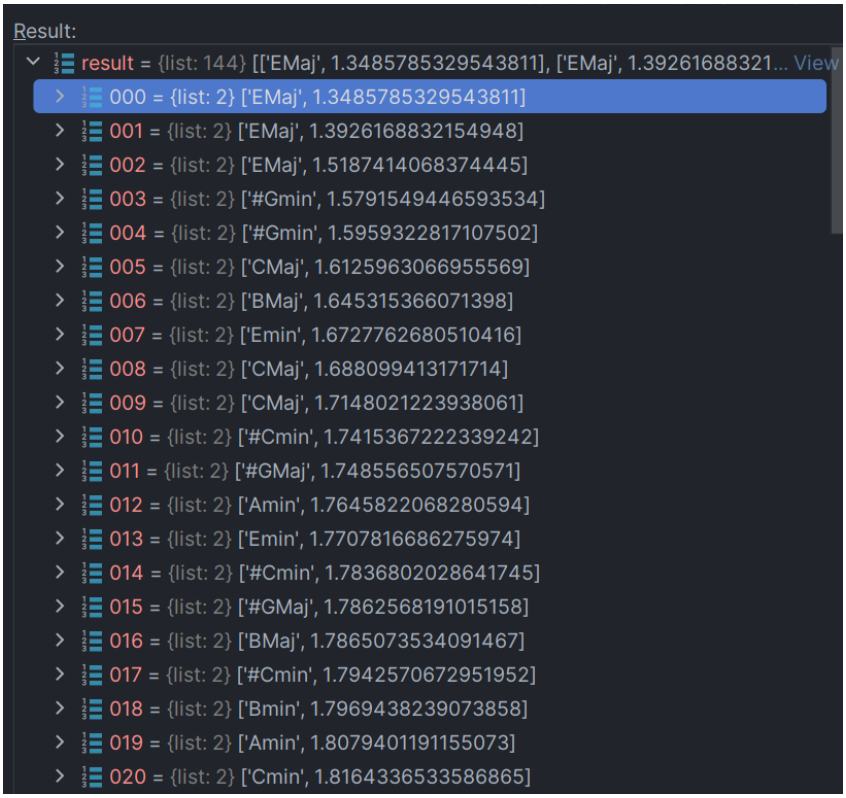


图 4.2 一个 Cmaj 和弦的 KNN 结果

图 4.2 是一个 Cmaj 和弦的识别结果。在这个 KNN 模型中，我使用了 Alicias Keys 钢琴音源采样作为训练数据，在 7 个不同的八度音程内，在 12 个不同的根音上构成了由大小三和弦、大小七和弦以及属和弦构成的 5 种不同的和弦。

在由 CQT 频域谱计算得到的与训练数据集的 84 维空间欧式距离中，模型给出的最近三个样本数据标签均为“EMaj”。这是由于 CMaj 和弦的三度音（E）与 EMaj 和弦的根音（E）相同，CMaj 和弦的根音与 EMaj 和弦的五音（B）、CMaj 和弦的五音（G）与 EMaj 和弦的三音（#G）都相差半音导致的，这也是 CQT 频域谱的不足所在。

但是，KNN 在和弦类型的聚类中取得了较好的准确性，可以考虑将 KNN 的结果与 PCP 的结果相结合，提高整个系统的准确性。

利用 DNN 网络进行深度色度提取，这是近几年在计算机和弦识别领域较为新颖的研究课题。但是目前缺少完善的训练数据集，数据集大多集中在单一乐器的演奏上，适用于完整音乐的数据集较少。

5 总结与展望

本次实验尝试探究计算机技术与人文艺术类学科的交叉，在音乐和弦的识别工作中取得了初步的进展。计算机自动和弦识别系统是一种能够从音频信号中提取和弦信息的技术，它在音乐分析、生成、教育等领域有着广泛的应用。自 1999 年 Takuya FUJISHIMA 提出 PCP 模型以来，和弦识别技术也随着时间不断迭代。本文尝试了历年来诸多不同的算法，进行对比与改进，尝试提高计算机自动识别和弦系统的准确率。

计算机科学与音乐领域的结合，体现了当今世界各学科交叉的趋势，然而目前来说，各种人工智能算法与其他智能算法的运用大部分集中在计算机视觉、NLP 等领域，计算机科学在音乐领域中的研究相对较少。未来的发展方向有以下几个方面：

首先，针对目前在计算机算法在音乐研究中数据集较少的问题，有望能产生更多多元化的、顺应时代发展的、更精确的数据集。

其次，是计算机科学在音乐领域中更多实用功能的落地，除开和弦识别外，目前也出现了诸如人声伴奏分离算法、自动作曲与作词 AI、AI 人声合成等先进应用。随着计算机科学的不断发展，有望出现更多在音乐创作、音乐分析领域中更为实用的应用。

参考文献

[1] 李重光. 音乐理论基础 [M].北京：人民音乐出版社,1962:253.

[2] 伊·杜波夫斯基，斯·叶甫谢耶夫，伊·斯波索宾，符·索科洛夫. 和声学教程[M].北京：人民音乐出版社，2008:435

[3] 阿诺德·勋伯格. 作曲基本原理 [M]. 上海：上海音乐出版社，1984

[4] 吴祖强. 曲式与作品分析[M]. 北京：人民音乐出版社，2003:408

[5] 同济大学数学系. 高等数学 下册 第七版 [M]. 北京：高等教育出版社，2014：325-326

[6] 王万良. 人工智能导论 第五版[M]. 北京：高等教育出版社，2020：211-250

[7] Takuya FUJISHIMA，Realtime Chod Recognition of Music Sound: a System Using Common Lisp Music[J]. ICMC Proceedings,1999,464-467

[8] Filip Korzeniowski and Gerhard Widmer. “Feature Learning for Chord Recognition: the Deep Chroma Extractor” [J],17th International Society for Music Information Retrieval Conference,2016

[9] Siddharth Sigtia, Nicolas Boulanger-Lewandowski, Simon Dixon. “Audio Chord Recognition with a Hybrid Recurrent Neural Network” [J], 16th International Society for Music Information Retrieval Conference, 2015.

[10] Ono, Nobutaka, et al. "Separation of a monaural audio signal into harmonic/percussive components by complementary diffusion on spectrogram." [J]，2008 16th European Signal Processing Conference. IEEE, 2008.

[11] Schoerkhuber, Christian, and Anssi Klapuri. “Constant-Q transform toolbox for music processing.” 7th Sound and Music Computing Conference, Barcelona, Spain. 2010.

[12] Chai, Wei, and Barry Vercoe. "Detection of Key Change in Classical Piano Music." In ISMIR, pp. 468-473. 2005.

[13] Allen, Paul E., and Roger B. Dannenberg. "Tracking musical beats in real time." In ICMC. 1990.

[14] Oliveira, Joao Lobato, Fabien Gouyon, Luis Gustavo Martins, and Luis Paulo Reis. "IBT: A Real-time Tempo and Beat Tracking System." In ISMIR, pp. 291-296. 2010.

[15] Cabral, Giordano, Jean-Pierre Briot, and François Pachet. "Impact of distance in pitch class profile computation." In Proceedings of the Brazilian Symposium on Computer Music, pp. 319-324. 2005.