# 音乐节拍跟踪

音频节拍跟踪的任务是从一个音乐输入中恢复一段时间序列，该时间序列与人类可能敲击其脚的时间相一致。本课题的目标是建立一个节拍跟踪器和找到一些适合这个跟踪器训练的数据集，例如数据集SEC-MIREC，舞厅的数据集，测试集的GTZAN节奏等。

用于自动节拍跟踪算法评估的地面实况标注包括标记节拍并将其存储在每一段音乐片段的文本文件中。这些注释可以第一次使用一个自动节拍跟踪检测节拍，然后手动纠正这些节拍的位置。在标注过程中，跟踪器可以被一次次地训练。

此任务有大量的评估指标，可以通过它们将检测到的beat标记为正确或不正确而区别开来。大多数这些指标使用的概念是约70毫秒的公差窗口围绕注释的节拍。如果检测到的敲打在注释周围的窗口中，则检测到的beat是正确的，并且为所有评估度量值给出了1分，除了cemgil score，它给出了0到1之间的得分，这取决于检测到的节拍来自注释的距离。基于连续性的评估措施如cmlc、cmlt、amlc、amlt如果在17.5%的速度和相位公差范围内报告，且适用于先前检测到的节拍，则认为beat正确。

研究了艺术方法的两种状态——essentia的beattrackermultifeature和madmom的mmbeattracker。在essentia的方法中，首先计算不同的开始检测函数，并通过基于第一阶hmm的概率方案来确定摘录的速度和节拍。madmom的方法是数据驱动的，并结合了关于注释数据的rnns，为每个摘录输出一个beat激活函数。madmom方法的第二阶段是动态贝叶斯网络，它通过隐藏和观察变量之间的依赖关系，计算摘录、音乐节拍和摘录的节奏模式。还对SMC \_MIREC和舞厅数据集进行了评估，首先在两个数据集上运行检测节拍的方法，然后使用mir \_eval库评估检测到的节拍。结果表明，madmom的实现优于essentia的实现。

为了检验这两种跟踪器的稳定性，将不同的数据增强算法应用于数据集，并对其各自的性能进行了检验。madmom的性能略有下降，但它仍然在原始数据集上的结果的3 - 4%范围内产生结果。在应用部分扩增系统为discussedhere后，essentia的性能有所提高。一个简单的预处理步骤，应用一个带阻滤波器来滤除中间频率，以清除冲击的发作，也增加了essentia的性能。通过指定一个更高的可能的速度值范围，对madmom的函数进行参数优化，在它们的性能提高了更高的速度候选范围。

使用python合成一个基于敲击的数据集(代表更改单个片段中的时间标记)。通过允许在速度、鼓分量强度等方面的随机偏差，在很小程度上模拟了microtiming误差。madmom的实现通过该数据集运行，以检测节拍，实现了令人满意的评价结果。

还提出了一种新的评价指标，以解决一片inthisblog中的韵律变化。此度量给出了正确解码测量位置的片段百分比。  
  
所有的代码用于评价算法包含[在这里](https://github.com/kushshr/AudioBeatTracking)

对合成数据集和它的注释的代码是包含[在这里](https://github.com/kushshr/drumpy)。该数据集包括音频和注释可以通过执行文件合成\_ drums.py基因从上面的链接。一些合成的作品可以听[在这里](https://drive.google.com/open?id=0BzQOxy4vnEMiQktUeVVvbGczS1k)

中期报告包含[在这里](https://drive.google.com/open?id=0B595ymBeNCqjUVcwdXNQTUU3TjA)

在本周我主要是看有关于音乐节拍跟踪这一系列的文章，可以看做是对于madmom的一个应用吧，而节拍跟踪最重要的不在于实现而在于评价，如何才算是好的跟踪方法，这是要重点研究的，希望我所看到的一点点东西能够对其他同学稍微有些启发。