

基于遗传算法的机器作曲

第 4 题第 1 组

2023 年 12 月

摘要

关键词: a,b,c

目录

1 引言

2 遗传算法

Algorithm 1 Genetic Algorithm

- 1: Initialize population:
 - 2: Create a population with random individuals
 - 3: Evaluate fitness of individuals:
 - 4: Calculate fitness value for each individual in the population
 - 5: **repeat**
 - 6: Select parent individuals:
 - 7: Use selection algorithm to choose individuals as parents
 - 8: Perform crossover:
 - 9: Apply crossover operation to selected parent individuals to generate new individuals
 - 10: Perform mutation:
 - 11: Apply mutation operation to the new individuals
 - 12: Evaluate fitness of new individuals:
 - 13: Calculate fitness value for the newly generated individuals
 - 14: Select surviving individuals:
 - 15: Choose which individuals survive and form the new population
 - 16: **until** termination condition is met
 - 17: **return** Best individual or best solution when termination condition is reached
-

3 适应度函数

适应度函数作为遗传算法的核心，直接决定了遗传算法模型如何选择遗传过程中的乐曲。选择适应度函数的主要挑战在于确定如何量化乐曲的质量以确定其遗传的概率。然而，音乐本身具有主观性，难以挖掘绝对客观的量化指标评价任意乐曲的质量。为了使遗传算法模型尽可能客观地评价生成的乐曲质量，我们调查并尝试了诸多基于乐理特征的适应度算法，将适应度函数定义为各个乐理特征评价的加权。

另外，我们利用卷积神经网络模型 (CNN)，收集了乐曲数据集以训练一个乐曲的分类器，并对于给定的乐曲将分类器的预测结果作为乐曲质量的另一个评价准则。

综合以上两种方法，用 λ 表示权重， f 表示基于不同乐理特征的适应度函数，更高的适应度函数评分意味着更低的乐曲质量，使乐曲在遗传中的贡献越小。 $mode$ 表示是否采用卷积分类器结果， τ 表示分类器预测结果，我们的模型适应度函数为

$$f(x) = \sum_{i=1}^n \lambda_i \cdot f_i(x) + \lambda_{CNN} mode \cdot \tau(x) \quad (1)$$

3.1 基于乐理特征的适应度函数

我们参考了文献，选择实现了其中 5 个最基本的适应度函数。我们认为，基于乐理特征的适应度函数应该遵循音程协和，调性一致的准则。所以我们首先确定基于音程协和性的适应度函数，以及基于调性一致性的适应度函数，使遗传算法模型倾向于保留具有更多协和音程，以及更具有自然大调调性的乐曲。

此基础上我们考虑了音乐音符的多变性。为了防止遗传算法总是收敛于一列等音序列（尽管这也被定义为协和且具有良好的调性），或者只包含最协和的纯一度与纯八度音程的序列，我们考虑了基于音程的自相似度，以及基于音类分布的适应度函数，使遗传算法模型倾向于保留使用了更多种类音程，以及更多种类音类的乐曲。

最后，我们考虑了乐曲节奏的多变性，在遗传过程中引入了休止与延长变异，并使用基于节奏相似度的适应度函数来评价乐曲的节奏质量，使得生成结果在节奏上有别于初始随机种子的八分音符序列节奏。

3.1.1 基于音程协和性的适应度函数

基于各音程的性质，我们对所有相邻音程的协和性进行评分（跳过休止符或者延音符）。为简单起见，对音程定义进行简化，只考虑两个音级实际相差的半音数，而忽略音名间的度数。具体取值如表 1 所示。

表 1: 音程协和性的评分		
音程种类	音程	评分
完全协和	纯一、四、五、八度	1
不完全协和	大小三度、大小六度	2
二度	大小二度	3
七度	大小七度	3
超八度	超过八度的音程	5

根据该规则，对某一遗传个体的音程进行评分，根据评分结果计算出音乐个体每小节的音程平均得分，如式 (2) 所示。

$$a_i = \frac{1}{n_i} \sum_{j=1}^{n_i} x_{i,j} \quad (2)$$

式中， $x_{i,j}$ 表示第 i 小节的第 j 个音程， n_i 表示第 i 小节相邻两个音符的音程数目（跳过休止符和延音符）， a_i 表示第 i 小节的平均音程得分。接着计算音乐个体每一小节的音程得分方差，如式 (3) 所示。

$$b_i^2 = \frac{1}{n_i} \sum_{j=1}^{n_i} (x_{i,j} - a_i)^2 \quad (3)$$

式中, b_i^2 表示第 i 小节的音程得分方差。

随后, 给出参考音乐的每小节平均音程得分 μ 和音程得分方差 σ , 并定义两个适应度函数, 如式 (4)(5) 所示。

$$f_1 = \sum_{i=1}^n \zeta_i |\mu_i - a_i| \quad (4)$$

$$f_2 = \sum_{i=1}^n \eta_i |\sigma_i^2 - b_i^2| \quad (5)$$

式中, f_1 、 f_2 分别表示基于音程得分均值和方差的两个适应度函数, ζ_i 表示第 i 小节的音程得分均值的影响权重, η_i 表示第 i 小节的音程得分方差的影响权重。

3.1.2 基于调性一致性的适应度函数

这一适应度函数使模型倾向于选择符合自然大调调性的乐曲。但是, 由随机种子开始的遗传过程并不保证收敛到某个特定的自然大调, 因此我们考虑了多种调性。对于所有 13 个自然大调, 我们分别统计不同调性的调内音符出现频率, 并将出现频率最高的调定义为该个体的调性, 对应的频率用于衡量调性一致性, 如式 (11)(12) 所示。

$$Key_j(I) = \frac{|K_j|}{|I|} \quad (6)$$

$$Tonality(I) = \max_j Key_j(I) - \frac{1 - \max_j Key_j(I)}{n - 1} \quad (7)$$

其中 K_j 表示属于第 j 个调性的音符数量 ($j = 1$ 表示 C 自然大调, $j = 2$ 表示 #C 大调以此类推), Key_j 表示第 j 个调性的调内音出现频率, n 表示遗传个体的音列长度, 即 $|I|$, $Tonality(I)$ 表示调性一致性, 与出现频率最高的调性对应的频率有关, 调性一致性越高, 该调性的调内音出现频率也就越高。这里同样用 3.1.2 的式 (8) 来得到基于调性一致性的适应度函数 $f_{Tonality}$ 。

3.1.3 基于音程自相似度的适应度函数

基于音程自相似度的适应度函数使模型倾向于选择包含更多种类的音程, 以避免算法收敛到只包含完全协和音程的个体上。这里的音程做了与 3.1.1 相同的简化。定义音程自相似度表示不同类的音程在音乐中出现的频率占比的均值, 计算公式如式 (6)(7) 所示。

$$SelfSimilarity(I) = \max(1, \frac{2\mu}{|I|}) \quad (8)$$

$$\mu = \frac{1}{|S|} \sum_{s \in S} count_s(I) \quad (9)$$

式中, $SelfSimilarity(I)$ 表示某一遗传个体 I 的音程自相似度值, $|I|$ 表示该遗传个体的音列长度, S 表示 I 中所有音程的种类集合, $|S|$ 表示音程的种类数目, s 表示某类出现在 I 中的音程, 如纯八度, $count_s(I)$ 表示音程 s 在 I 中出现的次数, μ 表示不同类的音程在 I 中出现的次数的均值

随后, 我们将自相似度与文献中的推荐值来衡量遗传个体在这一指标下的质量, 得到最终的适应度得分, 如式 (8)(9) 所示。

$$FeatureScore(t, e(I)) = \frac{1}{\frac{-1}{(x(t)-t)^2} (e(I) - t)^2 + 1} \quad (10)$$

$$x(t) = \begin{cases} 1, & \text{if } t < 0.5 \\ 0, & \text{if } t \geq 0.5 \end{cases} \quad (11)$$

式中, t 表示参考音乐的标准值, $e(I)$ 表示遗传个体在特征 e 上表现的值, 这里的 $e(I)$ 特指 $SelfSimilarity(I)$, $FeatureScore(t, e(I))$ 表示基于特征 e 的适应度函数 f_e , 越小说明该遗传个体表现的越好。据此得到基于音程自相似度的适应度函数 $f_{SelfSimilarity}$ 。

3.1.4 基于音类分布的适应度函数

音类分布用于衡量该遗传个体音符对应的音类分布情况，计算公式如式 (13) 所示。

$$PitchDistribution(I) = \frac{\frac{n - \max_j P_j(I)}{11} \cdot 12}{n} \quad (12)$$

式中, $P_j(i)$ 表示 I 中属于音类 j 的音符个数, 音类分布 $PitchDistribution(I)$ 越大, 表示该个体中音符更均匀地分布在各个音类当中。同样用 3.1.2 的式 (8) 来得到基于音类分布的适应度函数 $f_{PitchDistribution}$ 。

3.1.5 基于节奏相似度的适应度函数

节奏相似度衡量的是不同小节之间的节奏相似程度, 我们认为节奏相似度越高的音乐会更加动听。计算公式如式 (14) 所示。

$$RhythmSimilarity(I) = \frac{2}{nm(m-1)} \sum_{i=1}^{m-1} \sum_{i'=i+1}^m \sum_{j=1}^n |r_{i,j} - r_{i',j}| \quad (13)$$

其中 $r_{i,j}$ 表示第 i 小节第 j 个音符的类型, m 表示小节数, 这里我们将休止符、延长音、八分音符分别对应 $r=0$ 、1、2。因此, $RhythmSimilarity(I)$ 越大, 意味着节奏相似度越高, 这里我们直接将 $RhythmSimilarity(I)$ 作为基于节奏相似度的适应度函数 $f_{RhythmSimilarity}$ 。

3.2 基于卷积神经网络的适应度函数

神经网络模型已经被广泛地应用于分类任务。其中, 基于循环神经网络 (RNN) 以及此基础上的 LSTM 模型在处理序列数据上表现出优越的性能。但我们认为在这一任务中, 较短的向量 (32 位八分音符向量) 与较小的数据空间 (每个音符只能在接近两个八度范围内取离散值) 使得乐曲模态数据中蕴含的序列信息相比于传统的文本模态更少, 在较大模型容量下更容易产生数据过拟合的现象。

另一方面, 动听乐曲的特征包括片段的重复, 音程的协和, 调性的一致, 在一定程度上都可以理解为特定的局部模式特征。例如, 音程的协和性可以理解为较小感受野范围内的模式特征, 片段的重复与调性的一致可以理解为较大范围的模式特征。基于这一观点, 我们设计实现了卷积神经网络作为乐曲的分类器。

我们收集了超过 200 首流行乐曲数据作为正样本, 并生成了相当数量的随机片段作为负样本, 训练了卷积网络模型。并在测试集上获得了超过 90% 的准确率。在遗传算法中, 我们使用卷积网络模型的负样本预测概率值作为适应度函数评分, 具体为

$$\tau(x) = softmax_{-}(logit_{+}, logit_{-}) = \frac{e^{logit_{-}}}{e^{logit_{+}} + e^{logit_{-}}} \quad (14)$$

然而, 我们在实验中发现, 这一分类模型并不能很好地指导遗传算法的进行。由于训练数据量过小, 数据的过拟合现象仍然存在, 模型输出的概率预测值倾向于接近 0, 1 二值。因此, 我们尝试了直接使用模型的 logit 输出作为适应度函数值, 并将其线性变换到 0-1 区间内, 使得在训练过程中该项适应度函数值平稳变化, 且与其他项保持一致的数量级, 即最终的适应度函数为

$$\tau(x) = W(0.5 + \frac{logit_{-}}{2 \cdot Bound}) \quad (15)$$

其中 $W(x) = \min(\max(x, 0), 1)$ 为 0-1 窗口函数, 保证适应度函数取值位于 0-1 之间

4 实验

4.1 遗传算法超参数

4.2 模型训练集，测试集

5 结果分析

这主要有以下两个因素：第一，模型的训练数据与遗传算法数据间存在较大偏差。模型的训练数据包括收集的流行乐曲片段与随机片段，而遗传算法中模型的输入为遗传中的个体，与训练集存在明显的数据偏移，导致模型无法泛化到遗传种群上。

第二，模型的训练任务被建模为分类任务而非回归任务，这意味着模型的输出是离散值而非离散值。考虑到上述的数据偏移因素之后，模型的概率输出并不可信。我们尝试了将任务由二分类细化为三分类以及多分类，以不包括 CNN 模型的遗传算法输出作为中间类别样本，但只得到了微小的改进。我们认为，使用回归模型作为非乐理特征的评分器是更合适的，但这同样为数据收集带来了问题

6 小组分工

关于适应度函数的文献调研

遗传算法，卷积模型实现

模型数据集收集

实验报告撰写

7 参考文献