

令和3年度 特別研究報告書

機械学習による楽曲の雰囲気制御
とその応用について

龍谷大学 理工学部 情報メディア学科

T180437 高山 凌輔

指導教員 三好 力 教授

内容梗概

現在、数多くの歌や曲が世に出され、我々の耳に入ってきている。世に出ている多くの音楽の多くはそれぞれ雰囲気を持っている。例えば、明るい、暗い、楽しい、悲しいなど曲それぞれに持っている雰囲気は異なる。これまで様々な音楽家たちや作曲家たちの数百年にわたる懸命な研究により、音楽理論は確立され、これらの楽曲が内包する雰囲気は決定される。楽曲の持つおおまかな雰囲気を感じ方はあまり変化するものではなく、各楽曲により既に決定されているが、人が楽曲を聴いたときに感じる細かな感情には個人差がある。これはこれまでの研究により明らかであり、音楽と人の感情には密接な関係があることを示した。

そこで、本研究では機械学習を用いて人の心理状態を予測し、楽曲の雰囲気の制御を行い、人の感情と音楽にはどれほどの関係があり、また楽曲の雰囲気により人の感情はどれほど変化するのか、その検証と手法を提案する。人の心理状態を計測する方法には様々なものがあるが、本研究では心拍数を計測することによって人の心理状態を予測する手法を提案する。また、人の心理状態を予測する手法として、心拍数を測ることにどれぐらいの有用性があるのかについても検討する。

目次

第1章 はじめに

1.1 研究目的

第2章 既存技術

2.1 機械学習

2.1.1 機械学習の概要

2.1.2 AI、機械学習、ディープラーニングの違い

2.2 AI・人工知能による作曲

2.2.1 「AIVA」

2.2.2 「Ecrett Music」

2.2.3 「MuseNet」

2.2.4 「Amper Music」

2.3 心拍計測アプリ「Instant Heart Rate」

2.4 音楽のテンポ、音量、リズムの変化による人間の心拍の変化

第3章 提案手法

3.1 既存技術からの提案

3.2 心拍の計測、モニタリング

3.3 機械学習を用いた心拍数値による人の精神状態の予測、推定

3.4 人の精神状態による楽曲の生成、編曲

第4章 実験

4.1 実験環境

4.2 実験方法

4.2.1 ランニングの直前と直後の心拍数の計測

4.2.2 観測者の精神状態を予測するための機械学習

4.2.3 予測された精神状態に合わせた楽曲の生成

4.3 実験結果

4.3.1 ランニング直前・直後の心拍の計測結果

4.3.2 機械学習の結果

4.3.3 楽曲の生成

第5章 考察

5.1 結果の考察

5.2 今後の課題

第6章 まとめ

謝辞

参考文献

付録

第1章 はじめに

1.1 研究目的

音楽は昔から人の手で作られており、これまでに多くの音楽が作曲家によって世の中に生まれてきた。作曲は音楽行為の中で音楽の次第を考案する段階である。西洋音楽にあっては作曲とは具体的には楽譜を作成する作業である。特に著名作曲家の自筆による楽譜は珍重され、音楽博物館などに展示されていることがある。楽譜を紛失し再現できなくなった作品もある。DTM技術が発達した昨今では、作曲家が自らパソコンを操作し、専用の作曲ソフトを使用して曲を制作した後、デジタルデータ形式のまま納品することが多くなり、ポピュラー音楽では、楽譜を書かずに歴史に残る名曲を作る者が、東西問わず、数多く誕生している。

作曲者は、自分のイメージや伝えたいメッセージなどを内包させ、曲を作っていく。本来、それらを決めるのはその曲の作曲者であり、作曲者の作りたいイメージによって雰囲気は決定されていくものだ。世に出ている音楽にはカバーという形で、既存の音楽に影響や感銘を受けた人たちがその曲をアレンジし、さらに自分のオリジナリティを入れて世に出すという手法がある。そのアレンジにはアーティストそれぞれの個性やオリジナリティによって大きく変化し、元の原曲とは全く違う雰囲気の曲に仕上がっていたりする。

さて、我々がこれらたくさんの音楽を耳にしたとき、それぞれの雰囲気を感じるができるのは、人に感情があるからである。そして、人の感情と音楽には関係があるとされている。私はこの関係に興味を持ち、この関係を人々へ役立てることはできないだろうかと考えた。

本研究は、人の感情を機械学習により分析、予測し、曲の雰囲気を指定して作曲あるいは編曲することで、人の感情をコントロールし精神を安定させることで、人の感情や心情をよりよいものにすることを目的としたものである。

第2章 既存技術

2.1 機械学習

本研究は機械学習[1]を用いた楽曲の雰囲気制御を目的としているため、以下に基本的な機械学習の概要について説明する。

2.1.1 機械学習の概要

機械学習とは、コンピューターがデータから反復的に学習し、そこに潜むパターンを見つけ出すことである。そして、学習した結果を新たなデータにあてはめることで、パターンにしたがって将来を予測することができる。人手によるプログラミングで実装していたアルゴリズムを、大量のデータから自動的に構築可能になるため、さまざまな分野で応用されている。最も広く採用されている機械学習手法は、教師あり学習と教師なし学習の2つである。機械学習の大部分は教師あり学習である。教師なし学習やそのほか、半教師あり学習、強化学習という手法も使われている。

2.1.2 AI、機械学習、ディープラーニングの違い

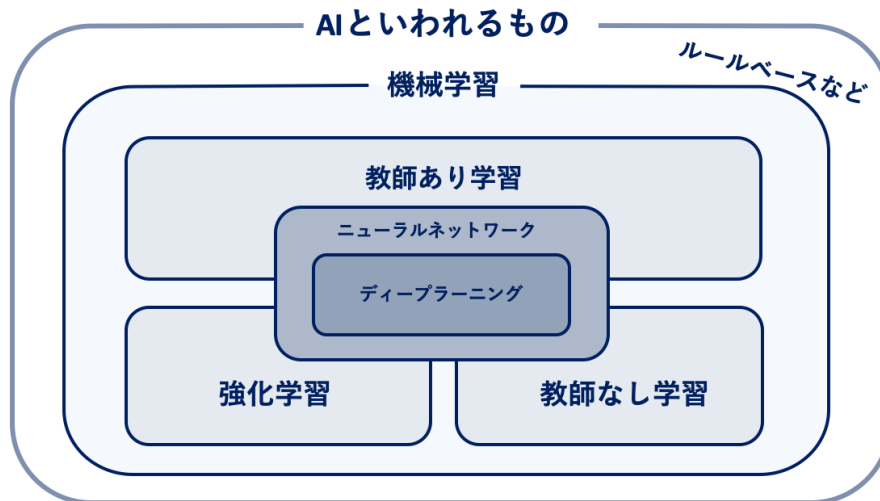


図1. 機械学習・AI・ディープラーニングの関係性

上記の画像から分かるように、AIの中に機械学習があり、機械学習の中にディープラーニングの技術が内包されている。機械学習はデータの中のどの要素が結果に影響を及ぼしているのか(特徴量)を人間が判断、調整することで予測や認識の精度をあげている。一方、

ディープラーニングはデータの中に存在しているパターンやルールの発見、特徴量の設定、学習なども機械が自動的に行うことが特徴であり、人間が判断する必要がないのが画期的であった。ディープラーニングで人間が見つけられない特徴を学習できるようになったおかげで、人の認識・判断では限界があった画像認識・翻訳・自動運転といった技術が飛躍的に上がった。

2.2 AI・人工知能による作曲

近年は、音楽業界においても AI が積極的に活用され始めており、特に「作曲」という分野で AI が存在感を増している状況である。[2]

これまで、人間が作曲を行う場合は、初めに音楽理論を頭で把握し、曲のコンセプトを決定する作業から進められていた。その次に、曲調を決めた上でコード進行を決定していく。そして、そのコードにメロディ、サウンドを加えることによって、初めてひとつの曲が完成することになる。

一方、AI が自動作曲を行う場合は、まず初めに大量の譜面を AI に学習させていくところから始まる。学習させることで、コードのパターンを AI に記憶させる。そして、曲調のような一定の指示を AI に出すことによって、AI は学習したデータをもとに作曲作業を行っていく。基本的に、作りたい曲のコンセプトさえ定まっていれば、AI は作曲の 3 要素の「スケール」「コード」「メロディ」を網羅することができる。

既に AI だけで楽曲制作を行える状況が生まれているように思えるかもしれないが、人間よりも AI が優れているとは言い切れない。現状では「AI に譜面のパターンを学習させる」という作業が大前提として必要であり、そのパターンをもとにした楽曲しか作ることはできない。そのため、一流の作曲家が行っているような、「これまでになかった独創的な音楽を創り上げる」といった作業を、AI だけの力で実現するのは難しいと言わざるを得ない。むしろ、自動作曲された曲のメロディが単調になってしまうケースも少なくないため、まだまだ改善の余地がある分野といえる。

以下に、AI による作曲の例をいくつか紹介する。

2.2.1 「AIVA」

AIVA[3]は「どうやって AI で人の心を動かす音楽を作るか」ということがテーマになっており、AI を使うことで過去の名曲に存在する特定のパターンなどを見つけ出し、人間の作曲プロセスをより短縮化することができるといった内容。ここで AI が行っているのは、楽曲のパターン分析をもとに曲の土台を作るといったこと。以下のようなことは AI ではなく「人間の手」によって行われている。

- ・ AI にモーツァルトやショパンなどの楽曲を学習させる
- ・ AI によって作られた曲をもとに人間が演奏する
- ・ AI による過去作品の盗作がないかチェックする
- ・ オーケストラ用に編曲する
- ・ クライアントの要望に応じてアレンジを加える

しかし、今後、もっと進化させることでオーケストラを作曲したり、ゲームや映画のシナリオを読ませることで、その場面に合った曲を作るといった機能も追加していく予定がある。

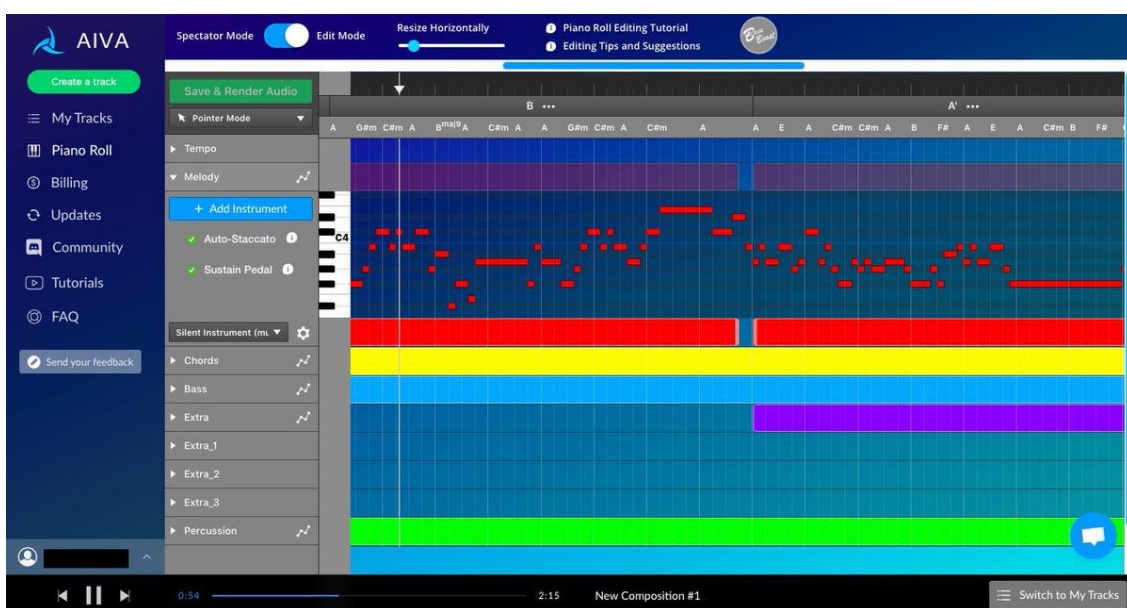


図 2. 「AIVA」 [3]より引用

2.2.2 「Ecret Music」

「Ecret Music」 [4]は、主に映像用の音楽をつくるための AI 作曲ツール。AIVA と同様オンライン上で扱うもので、曲の長さや雰囲気を選ぶだけで簡単に作曲することができる。

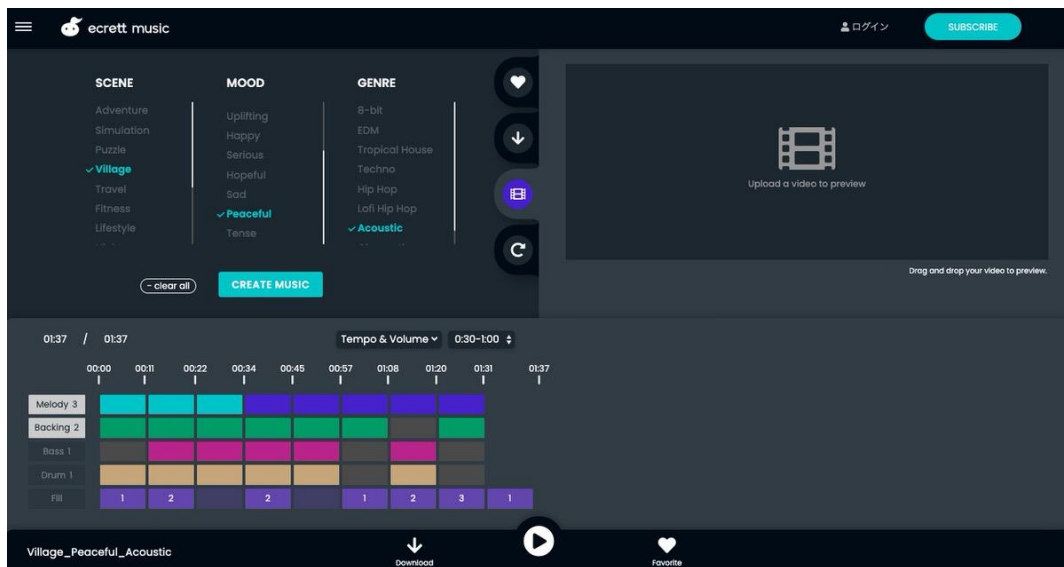


図 3. 「Ecrett Music」 [4]より引用

2.2.3 「MuseNet」

MuseNet[5]は機械学習によって、モーツァルトやショパンなどのクラシックの巨匠の楽曲と、ビートルズといった現代のアーティストの楽曲を組み合わせ、新しい音楽をつくることを可能にするサービス。公式ページでは「モーツァルトのトルコ行進曲の冒頭を使ってショパンのスタイルで作曲する」などといった面白い試みをいろいろ見ることができる。

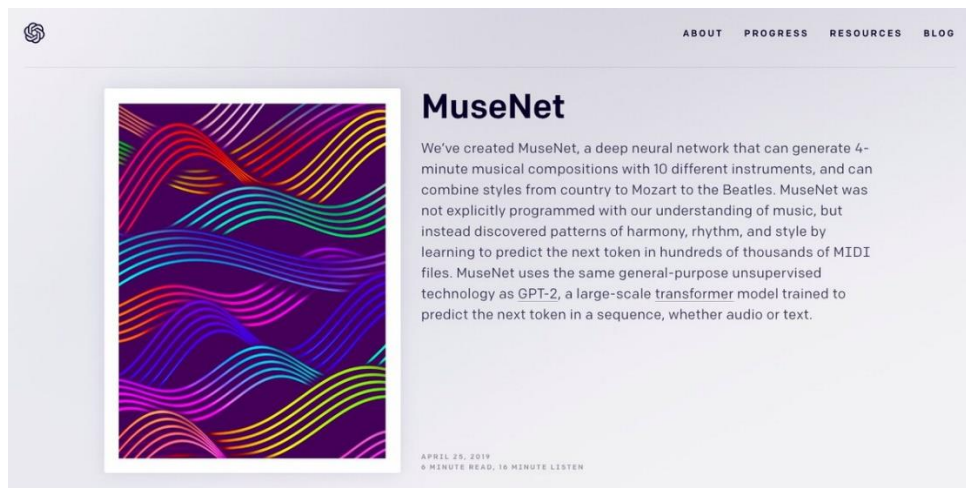


図 4. 「MuseNet」 [5]より引用

2.2.4 「Amper Music」

「Amper music」 [5]は人工知能を活用した作曲アプリ。ジャンルのカテゴリを選択して曲の長さを指定することで楽曲を生成することができる。

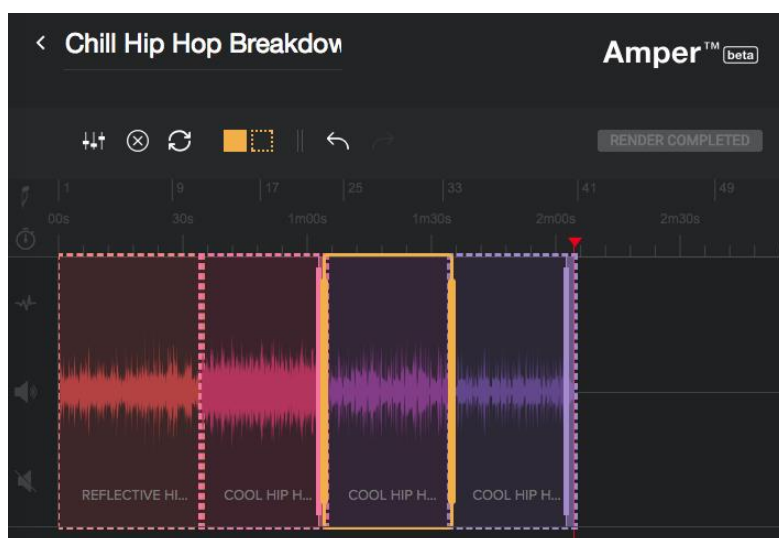


図 5. 「Amper Music」 [5]より引用

このように、AIによる作曲は積極的に取り組まれており様々な企業がソフトウェアを提供しているが、私が目的とする、音楽で人の感情を制御することに重きを置くソフトは見受けられなかった。

2.3 心拍計測アプリ「Instant Heart Rate」

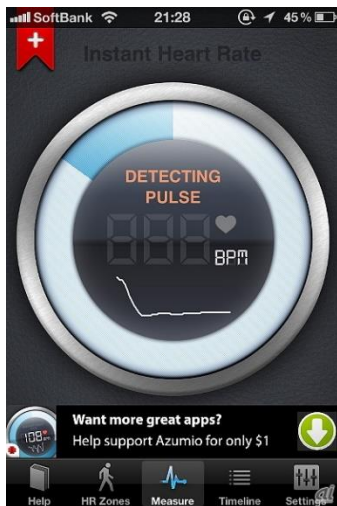


図 6. 「Instant Heart Rate」 [7]より引用

「Instant Heart Rate」 [7]は、iPhone のカメラ機能を使って心拍数を測るアプリ。カメラに指を当て、LED の光で脈を読み取ることにより、正確な脈を測定できる。

グラフでは、年齢に応じた平均値と比べて脈が早いか遅いかが表示できるほか、リスト表示で過去に測定した値とも見比べることができる。過去の計測も保存されているので、長期間測定を続けていけば、値の変動も把握しやすくなる仕組み。脈が測れる具体的な仕組みとしては、背面 LED を発光させ、血の流れによって変化する指先の微妙な色の変化をカメラで読み取っている。つまり「手のひらを太陽にかざして血の流れを見る」のと近いことが、iPhone の背面で行われている。強くカメラのレンズを押さえすぎたり、指先が冷たい状態では、正確な測定ができない。

音楽と感情、そして心拍数には関係があるとされている。次にこれについて説明する。

2.4 音楽のテンポ、音量、リズムの変化による人間の心拍の変化

音楽は心拍数に影響するが、ある音楽が心臓にどう影響するかは人によってそれぞれ異なることが分かっている。研究[8]では、ペースメーカーを装着した軽症の心不全患者 3 人に、クラシックのピアノコンサートで生演奏を聴かせ、音楽のテンポ、音量、リズムに大きな変化があったときの心臓の電氣的活動を、ペースメーカーのリード（導線）から直接測定する。具体的には、音楽を聴いて変化した対象者の鼓動が回復するまでの時間が測定

された。その結果、同じ音楽を聴いていても、心拍変動からの回復時間は人によって著しく異なることが判明した。このことから、ある人にとっては落ち着いた音楽が、別の人にとっては興奮する音楽となり得ることが分かった。同じ音楽の変化に対して、2人の人間に統計的に有意な変化が生じるとしても、その反応は、一方の人ではリラックスに、もう一方の人では興奮やストレスにつながるという具合に、正反対になることがあるという結果になった。そして、穏やかな曲調だった音楽が激しいものに変化した場合、それを予測していなかった人はストレスを感じるかもしれないが、音楽の盛り上がり部分が来ることを予測していた人にとっては開放感につながるという例を挙げている。人の心拍数と音楽には関係があり、さらにその音楽が人の精神状態にも影響することが分かる。であるならば、人の心拍数を計測することで人の精神状態を予測し、その予測から楽曲生成を制御することで、人の精神状態を予測することができるのではないだろうか。

第3章 提案手法

3.1 既存技術からの提案

私が提案するシステムは以下の通りである。

- 人の心拍数を測定、モニタリングする。
- 測定した心拍数から精神状態の予測、推定
- 予測、推定した心理状態から、それに対応した楽曲の編曲
- 編曲した音楽を聴取した観測者の精神状態の安定

これらのシステムの実現を考えたい。以下に、より具体的にシステムの構成と手法について述べる。

3.2 心拍の計測、モニタリング

心拍数の測定、モニタリングは、2.3でも上げたようなアプリを用いて計測を行う。

3.3 機械学習を用いた心拍数値による人の精神状態の予測、推定

心拍の計測で得られた心拍数のデータを訓練用データ、テストデータとして用いて機械学習を行う。

k-近傍法を用いて機械学習を行う。k-近傍法では、自分の近くにあるデータを何個にするかをkとして指定する。指定したkの値によって分類の精度が変わるので、どのくらいの値にするかが重要である。

集めた心拍数のデータセットと、それらが「安静時」なのか「興奮時」なのかを分類するデータセットとをそれぞれ学習させる。学習させることで、心拍数のデータだけでそのデータが「安静時」か「興奮時」なのか予測する。

3.4 人の精神状態による楽曲の生成、編曲

予測した人の精神状態をもとに、再生したい楽曲の雰囲気指定して生成、編曲する。安静時であるならば、明るくアップテンポな雰囲気の曲に編曲し再生することで、気持ちを高めてランニングに対する意欲向上を目指し、逆に興奮時であるならば、穏やかで落ち着いた雰囲気の曲を編曲し再生することで、気持ちを落ち着かせてランニングに対する意欲を下げてクールダウンを目指す。

第4章 実験

4.1 実験環境

本実験では計算機を用いて、Google Colaboratory を利用しての機械学習と「Amper Music」を利用しての楽曲の生成を行った。また、スマートフォンに心拍を測定することができるアプリをインストールした。これらの実行に用いた計算機、スマートフォンの環境を以下に示す。

表 1. 計算機の構築環境

OS	Windows10 (x64)
メモリ	16GB
CPU	Intel(R) Core(TM) i7 4790 @ 3.60GHz
GPU	NVIDIA GeForce GTX 1060 6GB

表 2. スマートフォンの性能詳細

名称	iPhone XR
OS	iOS 12
SoC	Apple A12 Bionic
CPU	6 コア, GHz
GPU	Apple GPU (4-core graphics)
メモリ	3GB

4.2 実験方法

4.2.1 ランニングの直前と直後の心拍数の計測

提案手法 3.2 で提案した、心拍の計測が実際に成功するか確認するため、計測の実験を行った。2.3 で紹介した「Instant Heart Rate」を用いて心拍数を計測していく。心拍数を計測するにあたって、心拍数に大きな変化が表れる場面を考えた。様々な場面を考えた結果、ランニングの直前と直後の場面が適していると考えた。人の精神状態も安静時か興奮時に分かれ、心拍数も大きく変化しやすい場面であるためである。このため、本研究ではランニン

グの直前と直後の心拍数を計測する。私はもちろん、私の兄弟や友人にも協力してもらい、それぞれランニングを行い、直前と直後の心拍数を3回以上測定した。

4.2.2 観測者の精神状態を予測するための機械学習

提案手法 3.3 での、k 近傍法を用いた機械学習で、実際に心拍数のデータセットを分類し、精神状態の予測、推定ができるか確認するための実験を行った。Python や機械学習、深層学習の環境を整えることが出来る Google Colaboratory を利用して機械学習を行う。計測した心拍数のデータは訓練用データとテストデータで 8:2 に分けて学習させ、与えたデータが「安静時」なのか「興奮時」であるのかを分類させ、予測する。その予測の正答率から分類の精度を求めることで、予測の精度を求めた。

4.2.3 予測された精神状態に合わせた楽曲の生成

提案手法 3.4 での、予測した人の精神状態をもとに、再生したい楽曲の雰囲気指定して生成、編曲ができるか確認するための実験を行った。楽曲の生成は 2.2.4 で紹介した「Amper Music」を利用する。データは「安静時」と「興奮時」なのかを予測したデータをもとに、それに合わせた楽曲を生成する。「安静時」ならばアップテンポな雰囲気楽曲を生成し、「興奮時」ならば落ち着いた楽曲を生成する。

4.3 実験結果

4.3.1 ランニング直前・直後の心拍の計測結果

4.2.1 の方法より、以下にランニング直前と直後の心拍数の測定結果を記す。ランニング直前を安静時、直後を興奮時としている。BPM は 1 分間で何回の脈拍があるかの単位である。

表 3.私の心拍の測定結果

回数	安静時[BPM]	興奮時[BPM]
1 回目	66	150
2 回目	65	145
3 回目	70	165
4 回目	75	170
5 回目	68	156

表 4.弟の心拍の測定結果

回数	安静時[BPM]	興奮時[BPM]
1 回目	60	165
2 回目	58	154
3 回目	74	134

表 5.妹の心拍の測定結果

回数	安静時[BPM]	興奮時[BPM]
1 回目	64	158
2 回目	49	148
3 回目	58	165

表 6.友人 A の心拍の測定結果

回数	安静時[BPM]	興奮時[BPM]
1 回目	65	167
2 回目	60	173
3 回目	63	159

表 7.友人 B の心拍の測定結果

回数	安静時[BPM]	興奮時[BPM]
1 回目	58	165
2 回目	53	158
3 回目	64	152

表 8.友人 C の心拍の測定結果

回数	安静時[BPM]	興奮時[BPM]
1 回目	65	165
2 回目	56	175
3 回目	72	168

4.3.2 機械学習の結果

4.2.2 の方法より、k 近傍法を用いた機械学習による分類の精度の結果を以下に示す。k の値を 1 から 15 までそれぞれ学習を行い、k の値によるそれぞれの分類の精度を示す。

表 9. k の値によるそれぞれの分類の精度

k	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
分類精度 [%]	86	86	86	86	86	86	87	86	86	86	86	85	86	86	86

4.3.3 楽曲の生成

4.2.3 の方法より、安静時と興奮時に再生する、アップテンポな曲と落ち着いた曲をそれぞれ生成した。以下にその結果を示す。

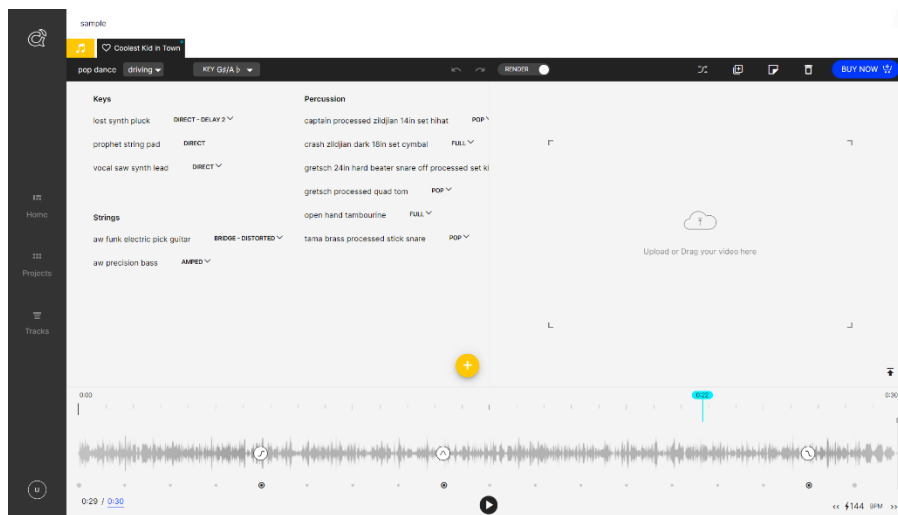


図 6. アップテンポな楽曲を「Amper Music」によって生成した画面

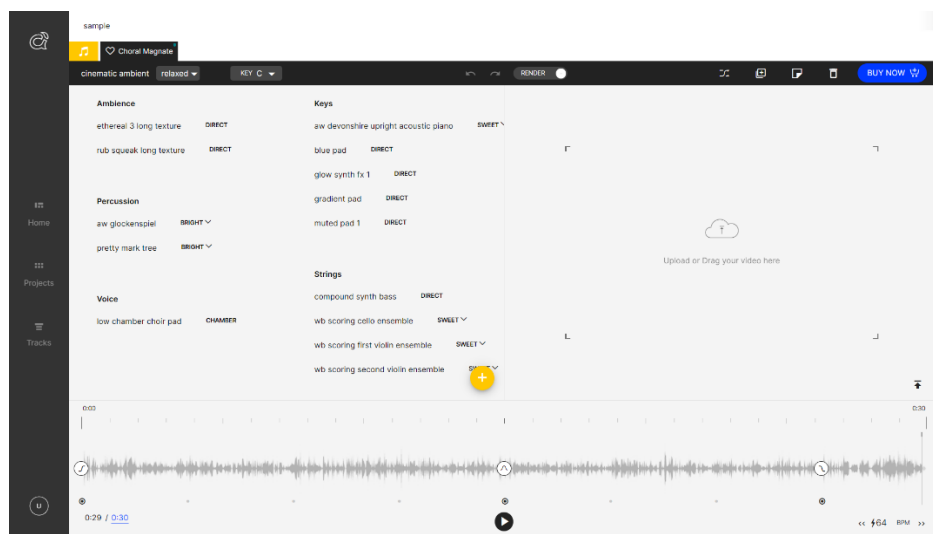


図7.落ち着いた楽曲を「Amper Music」によって生成した画面

第5章 考察

5.1 実験結果の考察

実験結果 4.3.1 より、安静時は 50~80BPM の間で、興奮時は 140~180BPM の間で計測された。このことから、安静時と興奮時の心拍数を正確に測定することができたと考えられる。また、結果を見ると、心拍数は個人差があり、人によって安静時と興奮時の平均的な心拍数は微妙な差があることが分かった。このことから、心拍数には個人差があるものの、安静時と興奮時では明確に差があり、それぞれある程度決まった範囲内の間の数値が出るということが分かった。これらの結果から、心拍の数値によって人の精神状態は予測することが可能なのではないかと考えられる。

実験結果 4.3.2 より、機械学習による k 近傍法の精度は、k がどの値の時でも 85%以上の精度になっていた。このことから、本実験においての機械は、学習により最低でも 85%と高い数値の確率で、与えられた値が安静時なのか興奮時なのかを分類し予測することができるといえる。よって、計測者の精神状態を十分に予測できていると私は考える。しかし、さらに精度を上げる学習を行うことも検討したい。今回の実験では、k の値を変更しても、あまり精度は変わらなかった。仮に k の値をさらに増やしたとしても、精度はそこまで変わらないと考えられる。そこで、さらに学習用データを追加したり、逆に減らしてみたりすることで、精度に変化が表れるか検証の余地がある。さらに、より細かい分類分けも可能だと考えられる。例えば、今の分類に追加で、「やや安静」や「少し興奮」や「ちょうどいい」などというような、極端な分類ではなく微妙な精神状態の分類分けを行うことで、より正確な人の精神状態の分析、予測が可能になると考えている。

実験結果 4.3.3 より、それぞれ安静時に再生するアップテンポな曲、興奮時に流す落ち着いた曲を生成、編曲することができた。私がそれぞれ聴取したとき、アップテンポな曲のときは気分が高まり、活動的な気分になり、逆に落ち着いた曲のときは気分が静まり、リラックスした気分になった。したがって、私を被験者とするならば実験は成功したと言える。しかし、問題点として、これがもし他の被験者であったとしても同じことが言えるだろうか。人が音楽を聴いたときに感じる感情には個人差があることが 2.4 から分かっている。また、本実験では安静時と興奮時のそれぞれ一曲ずつ楽曲を作成した。上記でもあげたように、今回の実験以外の精神状態のときにも対応させるには、それぞれ対応する楽曲を作成する必要が出てくる。これらの問題点全てを考慮するためには数曲だけでは対応できない。したがって、より多くの場合にも対応できるように、さらに多くの様々な楽曲を作成することが必要であり、それらの楽曲を聴いたときに被験者が感じる感情の調査を私だけでなく複数人に行うことも検討しなければならない。

5.2 今後の課題

今回の実験では、ランニングを行う直前と直後という極めて限定的なシチュエーションでの心拍の測定を行った。ここまで限定的な場面では、心拍が上がった原因はランニングという運動を行ったからであり、原因と結果が明確である。しかし、これがランニング以外の例ではどうだろうか。例えば、緊張、不安、羞恥心などを感じた時、人の心拍数が上がることもある。人の精神構造は複雑であり、個人の持つ感情はそれぞれであるため、これらの感情を感じたときに心拍数がどれほど上昇するのもまた人それぞれである。また、これらの感情を受けたことで、身体が示す反応は心拍数だけなのだろうか。嬉しいときの笑顔や緊張しているときの発汗や悲しいときに涙が流れるといったような様々な身体の反応の全ては感情を予測する手掛かりになるのではないだろうか。

私の研究の今後の課題はこの点にある。心拍数と人の感情とは関係性は存在するが、具体的にどんな感情であればどれくらいの心拍数であるかを断定することはできない。そしてまた、心拍数以外の身体の反応を考慮することも必要である。感情を測定、予測するためには、あらゆる身体の反応を計測、観察し、それら全てを考慮した上でなければ、完全な人間の感情の制御システムは構築できないだろう。あるいは、未だ発見されていない未知の要因が人間の感情には関わっているのかもしれない。

第6章 まとめ

本研究は、人の心拍数と機械学習を用いて人の心理状態を予測し、楽曲の雰囲気制御を行った。一般的で実用性のある、感情分析による自動編曲システムを作り、人の精神の安定、またはよりよいものにする手法を提案した。

提案手法では、心拍数から人の精神状態を機械学習により予測し、その予測に対応した楽曲の生成、編曲をすることで、人の精神状態の安定を目指す提案をした。

提案手法を実際に検証するために、実験では、私含め数人の協力のもと、複数人のランニングの直前と直後の心拍数を計測した。また、k近傍法を用いた機械学習により、取得したデータセットを「安静時」と「興奮時」に分類した。さらに、分類したデータから、その精神状態に対応するような楽曲の生成と編曲を行う実験を行った。

その結果、心拍数は安静時と興奮時とで正しく計測できた。また、機械学習の分類の精度も85%以上と高い数値が出たため、精度の高い分類ができたといえる。また、楽曲の生成と編曲では、私個人の感覚ではあるが、精神が安定する楽曲を作ることができた。以上のことから、今回の実験では、私が実現したい提案と手法のシステムは完全ではないが部分的に実現することができた。

今後の課題としては、機械学習の分類の精度をさらに上げていくことと、様々な人と感情にも対応した楽曲の作成があげられる。分類の精度を上げることで、さらに正確な精神の分析を可能としたい。また、より多くの人と感情にも対応した楽曲を作成することで、私が実現したい、より一般的な、感情分析による自動編曲システムを作り上げたい。

謝辞

本論文を作成するにあたり、多くのご指導、ご助言をいただきました三好力教授に深く感謝の意を表します。また、実験に協力してもらった私の兄弟や友人の皆様にも感謝申し上げます。

参考文献

[1] 機械学習

<https://ainow.ai/2019/11/26/180809/>

[2] AI・人工知能が作曲？自動で音楽を生成する自動作曲ソフト 10 選

https://aismiley.co.jp/ai_news/6-ai-softwares-that-automatically-compose-music/

[3] 「AIVA」

<https://www.aiva.ai/>

[4] 「Ecrett Music」

<https://ecrettmusic.com/>

[5] 「MuseNet」

<https://openai.com/blog/musenet/>

[6] 「Amper Music」

<https://www.ampermusic.com/>

[7] 「Instant Heart Rate」

<https://japan.cnet.com/article/35033695/>

[8] 堀田晴子・深村貫太・井上健,被験者の心拍数に応じたテンポによる音楽聴取時の心拍変動について,臨床教育心理学研究,Vol.33,pp.1-8,2007

付録

```
!pip install mglearn
import mglearn
import matplotlib.pyplot as plt
# matplotlib inline

from sklearn.neighbors import KNeighborsClassifier
from sklearn.model_selection import train_test_split

#x, y = mglearn.datasets.make_forge()

# 数値データ

import numpy as np
import random

x = np.random.randint(0, 200, (26, 2))
print(x)

x.shape

from google.colab import drive

drive.mount('/content/drive')

import pandas as pd
y = pd.read_csv('drive/My Drive/data/data.csv')

# 正解ラベル
print(y)

# y のデータの型
y.shape

# データの可視化
# discrete_scatter: 行と列、グルーピングを指定する
# 第1引数: 散布図に描写する各データの X 値
# 第2引数: 散布図に描写する各データの Y 値
# 第3引数: 散布図に描写する各データの LABEL
x.discrete_scatter(x[:, 0], x[:, 1], y)
plt.show()

# test 用、train 用データを生成
X_train, X_test, y_train, y_test = train_test_split(x, y, random_state=0)
print(
    X_train.shape)
print(
    X_test.shape)

# 学習させる(k 値は 3 に選択)
clf = KNeighborsClassifier(n_neighbors=3)
clf.fit(X_train, y_train)

# 予測結果と正解ラベルを見比べる
print(clf.predict(X_test))

print(y_test)

# どのくらい正しいか
clf.score(X_test, y_test)

clf_10 = KNeighborsClassifier(n_neighbors=10).fit(X_train, y_train)
clf_10.score(X_test, y_test)

# for 文を使って 1-15 までのモデルを試す
for n_neighbors in range(1, 16):
    clf = KNeighborsClassifier(n_neighbors=n_neighbors).fit(X_train,
y_train)
    print("Test set accuracy: n_neighbors={}, {:.2f}".format(n_neighbors,
clf.score(X_test, y_test)))

# 内部の境界線の変化を図で確認する

# subplot: グラフを書くためのキャンバスを用意(グラフの行、列、大きさ)
# fig: ウィンドウ、axes: タブ
fig, axes = plt.subplots(1, 5, figsize=(15, 3))

# for 文で k 値が 1,3,5,10,15 のときグラフを描写する
for n_neighbors, ax in zip([1,3,5,10,15], axes):
    clf = KNeighborsClassifier(n_neighbors=n_neighbors).fit(X_train,
y_train)

    # plot_2d_separator は領域を 2 つに分割できる
    mglearn.plots.plot_2d_separator(clf, x, fill=True, ax=ax, alpha=0.5)
    mglearn.discrete_scatter(x[:, 0], x[:, 1], y, ax=ax)
    ax.set_title('{} neighbors'.format(n_neighbors))

plt.show()
```