

トランペットのロングトーン練習に対する 即時フィードバック自動生成の研究

泰えりか^{†1} 青柳龍也^{†1}

概要：トランペットの上達には、問題点やアドバイスなどのフィードバックを基礎練習の直後に得られることが重要である。しかし、アマチュア奏者は、このような即時フィードバックを得ることが難しい。そこで、本研究ではトランペット奏者の基礎練習を対象に、即時フィードバックを自動生成するシステムの開発を行っている。今回は基礎練習の中からロングトーンを取り上げ、アマチュア、音楽大学生、プロの奏者計9名の演奏を録音し、機械学習を利用して演奏の音響的特徴からフィードバックすべき問題点を検出する実験を行なった。

キーワード：機械学習、音響特徴分析、トランペット、基礎練習、フィードバック

1. はじめに

トランペット奏者が上達するためには基礎練習が重要である。基礎練習とは、出した音を維持すること、指定された長さで演奏すること、音階を滑らかに演奏することなどを目的とし、その目的を達成するために楽曲を用いず行われる練習で、トランペットの基礎練習には音を長く延ばすロングトーン、音階を練習するスケール、連続する同じ音を綺麗に区切るタンギング、同じ運指で滑らかに音を変化させるリップスラーなどがある。

しかしながら、これらの基礎練習を行う際に、奏者自身がそれぞれの基礎練習の目的を達成できているかを客観的に評価することは難しい。普段聞いている自分の声と、録音された自分の声の聞こえ方に違いがあるように、トランペット奏者が聞いている自身のトランペットの音と、周りの人に聞こえるトランペットの音には違いがある。そのため、奏者自身は基礎練習の目的が達成できていると思いついていても、客観的に聞いてみると改善の余地があることが少なくない。客観的な評価を得るための代表的な例として、トランペット教室に通いプロ奏者のレッスンを受けるというものがあるが、レッスン以外の時間で基礎練習を行う場合には自己評価が必要となる。また、自身の音を録音してフィードバックを行う方法もあるが、再生する手間があり、場合によっては何度か聞き直さなければいけない場合もあり、効率が悪い。

この問題を解決するために、トランペットの基礎練習支援システムを提案する。本システムでは一般的によく行われている基礎練習をサポートし、演奏者に対して即時のフィードバックを与えることを想定する。フィードバックは演奏の良し悪しだけでなく、演奏を改善するためにどのように練習を行うべきかをアドバイスすることを目指す。

フィードバックを与えるにあたり、何を改善すべきかを的確に言い当てる必要がある。改善すべきポイントがわかれば、与えるべきアドバイスや練習のアプローチ方法が絞

られるからだ。後述の関連研究より、本研究では機械学習を用いて音が遅滞なく発音されているかどうかを評価することを試みた。

2. 関連研究

トランペットの教則本として最もの有名なものの一つに、Joseph Jean-Baptiste Laurent Arban 著の金管教則本がある。現在、この教則本は多数の金管奏者によって校訂されており、曾我部清典校訂版[1]によれば、トランペットの音は“素直・明瞭に遅滞なく”発音されることが求められる。また、[1]の“最初の練習曲”では、本研究で取り扱う基礎練習に似た練習曲が提示されているが、それらを練習する際には“鮮やか”かつ“力強い響き”で、“出した音を保持”するよう指示されている。

既存のアプリで基礎練習に利用できるものは、メトロノームやチューナーの他に KORG の cortosia がある[2]。このアプリは、楽器の「いい音」を評価するアプリで、ピッチの安定度、音量の安定度、音色の安定度、音色の豊かさ、アタックの明瞭度の5つの要素を総合的に評価し、100点満点で採点する。チューナーと同様に、リアルタイムで「いい音」かどうかを知ることができる。このアプリでは音が“遅滞なく”発音されているかについては明確になっていない。そこで、本研究では単音が遅滞なく発音されているかを評価する手法を考案する。

3. 本研究のアプローチ

Python の sklearn ライブラリを使用し、機械学習を行う。学習に使用する特徴量には、Python の音楽情報処理ライブラリである librosa を使用し、音の出だし部分の波形の二乗平均平方根の値を用いた。分類器には、AdaBoostClassifier, BaggingClassifier, ExtraTreesClassifier, GradientBoostingClassifier, SVC, RandomForestClassifier を使用し、パラメータを調整した上で、それぞれの分類器の auc の値を比較することで精度を評価した。なお、使用した Python のバージョンは 3.5.2、実行環境は macOS 10.12.2 Sierra である。

^{†1} 津田塾大学

3.1 使用データセット

トランペットのアマチュア奏者7名、音楽大学生1名、プロ奏者1名の計9名のロングトーンの演奏を録音し、463データを学習に使用した。採集したデータの音域は、五線譜下第3線のFから五線譜第1線のFまでの2オクターブ間の半音階、音の長さは四分音符=60で4拍とした。録音環境は、雑音の少ない静かな個室で、演奏者はあらかじめ録音されたメトロノーム音をイヤフォンで聞いているため、収集したデータには純粋なトランペットの音が記録されている。サンプリングレートは44100、チャンネル数1のwavデータ形式で録音を行った。

演奏者ごとに録音されたデータから、librosaのonsetによって音の出だしを検出し、そこから4.5秒間ずつ切り出すことにより、データを単一のロングトーンごとに分割した。

遅滞ない演奏かどうかのラベル付けは、執筆者が手作業で行った。全てのデータに対して1度ずつ評価を行うが、一度だけの再生で判断がつかなかった場合、複数回再生して評価をつけた。

4. 実験

4.1 分類器のパラメータ調整

上述の通り、特徴量には音の出だし部分の波形の二乗平均平方根を使用し、分類器にはAdaBoostClassifier, BaggingClassifier, ExtraTreesClassifier, GradientBoostingClassifier, SVC, RandomForestClassifierの6つを使用した。各分類器は、sklearnライブラリのgrid-searchを使用してパラメータ調整を行った。調整したパラメータと調整した範囲、本実験で採用したパラメータを表1に示す。パラメータ調整には本研究で使用しているすべてのデータを使用し、個別に調整を行う。いずれの分類器もパラメータを変更しないデフォルトの状態よりも優れていることを確認した上で、パラメータを決定した。

表 1 各分類器のパラメータ調整

| SVC | | |
|------------------|--------------------------------|--------------|
| パラメータ | 値の範囲 | 採用した値 |
| C | 10^{-3} から 10^3 までで等間隔の20点 | 1.438449888 |
| gamma | 10^{-6} から 10^4 までで等間隔の20点 | 0.615848211 |
| kernel | linear'または'rbf' | rbf' |
| RandomForest | | |
| パラメータ | 値の範囲 | 採用した値 |
| n_estimators | 2から100 | 72 |
| criterion | gini'または'entropy' | entropy' |
| Bagging | | |
| パラメータ | 値の範囲 | 採用した値 |
| n_estimators | 2から100 | 71 |
| ExtraTrees | | |
| パラメータ | 値の範囲 | 採用した値 |
| n_estimators | 2から100 | 95 |
| criterion | gini'または'entropy' | gini' |
| GradientBoosting | | |
| パラメータ | 値の範囲 | 採用した値 |
| loss | deviance'または'exponential' | 2.976351442 |
| learning_rate | 10^{-3} から 10^3 までで等間隔の20点 | exponential' |
| n_estimators | 50から200 | 114 |
| AdaBoost | | |
| パラメータ | 値の範囲 | 採用した値 |
| n_estimators | 10から200 | 0.001 |
| learning_rate | 10^{-3} から 10^3 までで等間隔の20点 | 128 |

4.2 音の出だし部分の検出

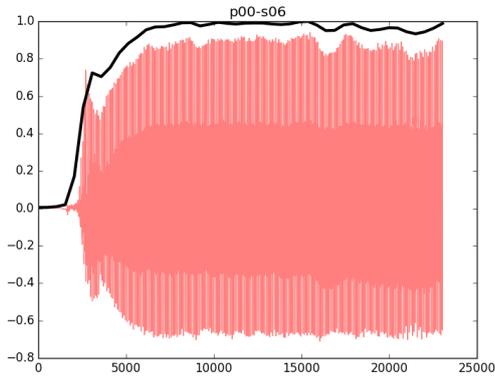
各分類器のパラメータが調整されたところで、次に特徴量となる音の出だし部分について検討する。

二乗平均平方根の計算は、librosaのrmseをデフォルト値で使用したため、フレーム長は2048サンプル、512サンプルごとにrmsを求めている。音の出だし部分として認識されるのは、せいぜい最初の0.5秒程度であると予想し、5フレームから55フレームまで、5フレーム刻みで変化させて、すべての分類器の予測値を利用したアンサンブル学習を行い、aucの値の変化を観察した。ここでのアンサンブル学習は、今回使用した6つの分類器の予測を、各分類器のaucの中央値を1として相対的に重み付けをし、ラベルを算出した。

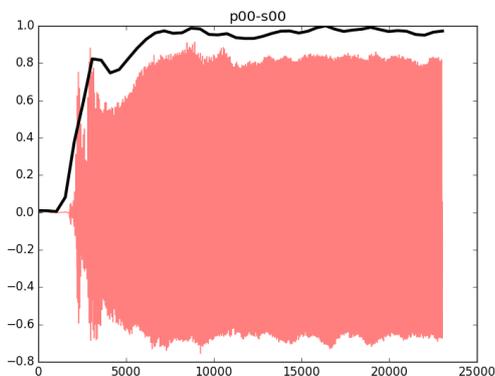
複数回試行してみたところ、安定した結果は得られなかったが、30フレームから55フレーム程度の範囲で比較的良好な結果が得られたため、本実験では今後、音の初めから45フレーム分(約0.5秒間)の波形の二乗平均平方根を用いて学習を行う。

本実験で使用した45フレーム分の二乗平均平方根をグラフ1とグラフ2で示した。赤色でプロットされているのが正規化された音源データの波形、黒の太線でプロットされているのが二乗平均平方根の値である。グラフ1は遅滞がないと評価されたデータで、二乗平均平方根の値が最大

値を迎えたとしばらく安定しているが、遅滞があると評価されたグラフ2では、最大値に達してから値が安定せず、黒い線が波打っているという特徴が見受けられた。



グラフ 1 遅滞のない音の出だし部分



グラフ 2 遅滞のある音の出だし部分

4.3 分類器の検討

次に、分割数 5 の交差検定を 6 回試行し、auc の値を比較する。結果は表 2 に示した。なお、表中の all はアンサンブル学習を行った結果である。表 2 において最も順位の悪かった 2 分類器、GradientBoostingClassifier と AdaBoostClassifier を除き、残る 4 分類器で同様に 6 回試行すると、アンサンブル学習による予測が各分類器単体での予測を上回る確率が上がった (表 3)。

表 2 分割数 5 の交差検定結果①

| 順位 | 分類器 | auc | 順位 | 分類器 | auc | 順位 | 分類器 | auc |
|----|------|--------|----|------|--------|----|------|--------|
| 1 | BC : | 0.63 | 1 | ETC: | 0.6444 | 1 | ETC: | 0.6369 |
| 2 | all: | 0.6136 | 2 | all: | 0.6415 | 2 | all: | 0.6221 |
| 3 | SVC: | 0.607 | 3 | SVC: | 0.6187 | 3 | SVC: | 0.6052 |
| 4 | RFC: | 0.5958 | 4 | BC : | 0.6177 | 4 | RFC: | 0.5972 |
| 5 | ETC: | 0.5915 | 5 | RFC: | 0.6162 | 5 | BC : | 0.5841 |
| 6 | ABC: | 0.5711 | 6 | GBC: | 0.6046 | 6 | ABC: | 0.5792 |
| 7 | GBC: | 0.5606 | 7 | ABC: | 0.5529 | 7 | GBC: | 0.5672 |

| 順位 | 分類器 | auc | 順位 | 分類器 | auc | 順位 | 分類器 | auc |
|----|------|--------|----|------|--------|----|------|--------|
| 1 | ETC: | 0.6421 | 1 | all: | 0.6175 | 1 | ETC: | 0.6352 |
| 2 | all: | 0.6161 | 2 | ETC: | 0.6132 | 2 | GBC: | 0.6311 |
| 3 | SVC: | 0.613 | 3 | BC : | 0.609 | 3 | all: | 0.6222 |
| 4 | RFC: | 0.6094 | 4 | SVC: | 0.6046 | 4 | RFC: | 0.6194 |
| 5 | BC : | 0.597 | 5 | GBC: | 0.5899 | 5 | BC : | 0.6166 |
| 6 | GBC: | 0.5766 | 6 | RFC: | 0.5865 | 6 | SVC: | 0.6129 |
| 7 | ABC: | 0.5467 | 7 | ABC: | 0.5577 | 7 | ABC: | 0.5596 |

表 3 分割数 5 の交差検定結果②

| 順位 | 分類器 | auc | 順位 | 分類器 | auc | 順位 | 分類器 | auc |
|----|------|--------|----|------|--------|----|------|--------|
| 1 | ETC: | 0.6179 | 1 | SVC: | 0.6178 | 1 | all: | 0.6429 |
| 2 | all: | 0.6165 | 2 | all: | 0.6145 | 2 | ETC: | 0.6219 |
| 3 | RFC: | 0.604 | 3 | ETC: | 0.6145 | 3 | RFC: | 0.6114 |
| 4 | SVC: | 0.597 | 4 | RFC: | 0.6018 | 4 | SVC: | 0.6109 |
| 5 | BC : | 0.5872 | 5 | BC : | 0.5878 | 5 | BC : | 0.608 |

| 順位 | 分類器 | auc | 順位 | 分類器 | auc | 順位 | 分類器 | auc |
|----|------|--------|----|------|--------|----|------|--------|
| 1 | all: | 0.6358 | 1 | all: | 0.6466 | 1 | all: | 0.6616 |
| 2 | BC : | 0.6252 | 2 | RFC: | 0.6359 | 2 | RFC: | 0.6534 |
| 3 | ETC: | 0.6188 | 3 | ETC: | 0.6333 | 3 | ETC: | 0.6419 |
| 4 | SVC: | 0.6152 | 4 | SVC: | 0.6203 | 4 | BC : | 0.6338 |
| 5 | RFC: | 0.5899 | 5 | BC : | 0.6183 | 5 | SVC: | 0.623 |

4.4 交差検定時の分割数の検討

分割数を 100 に変更し、同様に交差検定を 6 回試行した (表 4)。分割数を大きくしても、以前として GradientBoostingClassifier と AdaBoostClassifier の精度は低かったが、この 2 分類器を除いた 4 つの分類器のみを用いたアンサンブル学習では、安定して 0.6 以上の auc 値を出力するようになった。

次に分割数を 50 に変更し、4 分類器のみで同様に試行したところ、分割数 100 の時よりもアンサンブル学習の精度が向上した。

表 4 分割数 100 の交差検定結果

| 順位 | 分類器 | auc | 順位 | 分類器 | auc | 順位 | 分類器 | auc |
|----|------|--------|----|------|--------|----|------|--------|
| 1 | all: | 0.6311 | 1 | all: | 0.6284 | 1 | all: | 0.6226 |
| 2 | RFC: | 0.6265 | 2 | ETC: | 0.6185 | 2 | RFC: | 0.6085 |
| 3 | SVC: | 0.6125 | 3 | SVC: | 0.6046 | 3 | ETC: | 0.6049 |
| 4 | BC : | 0.6069 | 4 | BC : | 0.6025 | 4 | SVC: | 0.5942 |
| 5 | ETC: | 0.5988 | 5 | RFC: | 0.6003 | 5 | BC : | 0.5886 |

| 順位 | 分類器 | auc | 順位 | 分類器 | auc | 順位 | 分類器 | auc |
|----|------|--------|----|------|--------|----|------|--------|
| 1 | all: | 0.6392 | 1 | all: | 0.6433 | 1 | all: | 0.6235 |
| 2 | RFC: | 0.6272 | 2 | SVC: | 0.6372 | 2 | BC : | 0.6174 |
| 3 | ETC: | 0.6187 | 3 | RFC: | 0.6371 | 3 | SVC: | 0.6166 |
| 4 | BC : | 0.6172 | 4 | BC : | 0.619 | 4 | ETC: | 0.6068 |
| 5 | SVC: | 0.6127 | 5 | ETC: | 0.6151 | 5 | RFC: | 0.5993 |

表 5 分割数 50 の交差検定結果

| 順位 | 分類器 | auc | 順位 | 分類器 | auc | 順位 | 分類器 | auc |
|----|------|---------|----|------|---------|----|------|---------|
| 1 | all: | 0.64819 | 1 | all: | 0.65408 | 1 | all: | 0.64121 |
| 2 | ETC: | 0.62525 | 2 | BC : | 0.61988 | 2 | ETC: | 0.62211 |
| 3 | BC : | 0.61518 | 3 | ETC: | 0.60371 | 3 | RFC: | 0.61431 |
| 4 | RFC: | 0.59824 | 4 | RFC: | 0.60288 | 4 | SVC: | 0.58941 |
| 5 | SVC: | 0.59371 | 5 | SVC: | 0.59519 | 5 | BC : | 0.58698 |

| 順位 | 分類器 | auc | 順位 | 分類器 | auc | 順位 | 分類器 | auc |
|----|------|---------|----|------|---------|----|------|---------|
| 1 | all: | 0.63738 | 1 | all: | 0.65294 | 1 | all: | 0.64474 |
| 2 | RFC: | 0.61719 | 2 | ETC: | 0.62245 | 2 | RFC: | 0.6174 |
| 3 | BC : | 0.60954 | 3 | RFC: | 0.61501 | 3 | ETC: | 0.60823 |
| 4 | SVC: | 0.60646 | 4 | SVC: | 0.6032 | 4 | BC : | 0.60133 |
| 5 | ETC: | 0.5912 | 5 | BC : | 0.59302 | 5 | SVC: | 0.5889 |

5. おわりに

本研究では、トランペットの基礎練習に対して自動でフィードバックを与えるにあたり、音が遅滞なく発音されているかどうかを機械学習によって評価することを試みた。特徴量には音の出だし部分の波形の二乗平均平方根を使用し、AdaBoostClassifier, BaggingClassifier, ExtraTreesClassifier, GradientBoostingClassifier, SVC, RandomForestClassifier という 6 つの分類器を用いて予測を行い、auc の値を用いて精度を比較した。結果は BaggingClassifier, ExtraTreesClassifier, SVC, RandomForestClassifier の 4 つの分類器の精度が高かったため、それらを用いてアンサンブル学習を行うことでより精度を向上させることができた。また、交差検定を行

う際の分割数を増やすことによっても精度は向上した。

しかしながら、auc の値は依然として 0.6 前後であるため、特徴量や分類器を改善する余地がある。本研究では、特徴量を一貫して音の出だし部の二乗平均平方根を使用した。他の特徴量については検討していない。音響分析でよく使用されるフーリエ変換や、音名やコードを分析するのに便利なクロマグラム等を使用しようかと試みたが、音名が異なることによる特徴の差が生じてしまわないようにする正規化手法が検討できていない。また、分類器については、sklearn ライブラリでサポートされている分類器のすべては試行できておらず、パラメータに関しても大雑把な調整しか行っていないため、同じ分類器でもより精度の高いパラメータを発見できていない可能性もある。特に、精度が比較的悪かった GradientBoostingClassifier と AdaBoostClassifier については、パラメータの調整前後で精度の向上がそれほど大きくなかったため、改善の余地があると言える。交差検定の分割数についても、5, 50, 100 のみで試行したため、適切な値を使用できているとは断言できない。

また、本研究の最終目標は、基礎練習に対してフィードバックを与えることであるが、どのようなフィードバックを与えるかについては検討できていない。今後は、分類器の学習精度を高めるとともに、具体的なフィードバックを検討していきたい。

謝辞 トランペットの演奏データを提供してくださった被験者 9 名はじめ、本研究にご協力頂いた皆様に、謹んで感謝の意を表す。

参考文献

- [1] 曾我部清典. アーバン:金管教則本. 株式会社全音楽譜出版社, 2009, p. 6-16.
- [2] O.Romani et al.. A real-time system for measuring sound goodness in instrumental sounds. AES 138th Convention Paper, 2015.