

楽譜簡略化と自動補完伴奏による ピアノ演奏練習支援システム

福田 翼^{1,a)} 中村 栄太^{1,b)} 糸山 克寿^{1,c)} 吉井 和佳^{1,d)}

概要：本稿では、楽譜簡略化と自動補完伴奏によるピアノ演奏練習支援システムについて報告する。近年、楽器演奏の支援を行うアプリケーションの開発が活発になってきており、特にユーザひとりひとりにパーソナライズされたものが求められてきている。本システムでは、あらかじめ手本となる楽譜を与えてユーザの演奏ミスのデータを蓄積することで、ユーザが苦手とする楽譜の傾向を判断する。該当箇所の楽譜をユーザの演奏レベルに合わせて簡略化を行うことによって、ユーザの好みの楽曲を自分の実力に合った難易度の楽譜として練習することを可能とする。さらに、簡略化によって削除された音をシステムに自動伴奏させることによって、簡略化後の楽譜が不自然なものになったとしても、奏者に違和感を感じさせることのないようにする。また、ユーザの演奏ミスのデータを蓄積することができるため、そのデータと楽譜の特徴量を合わせたデータを訓練サンプルとすることで、学習モデルを得ることができる。こうして得られたモデルに楽譜の特徴量を与えることで、対象ユーザがその楽譜に対して演奏ミスしやすい箇所を予測し、ユーザが一度も演奏したことがない楽曲に対しても、そのユーザに合わせて楽譜の簡略化を行うことができる。

1. はじめに

楽器の中でもピアノは特に演奏人口が多く、奏者の演奏レベルも非常に幅広いことが知られている。その中でも、ある程度演奏ができる中級者は自分の好きな楽曲で練習する傾向がある。これは更なる上達へのモチベーションとなるが、一方でそのような楽譜は演奏難度が高いことが多いため、奏者の練習意欲を低下させることになるという問題がしばしば発生する。演奏難度の高い楽譜を奏者の演奏レベルに合わせて簡略化することができれば、この事態を避け、より効率的に演奏練習を促進することができる。

現在までに、楽譜の簡略化に関する研究として、ユーザが実際に演奏した音響信号を元に演奏レベルに合わせた難易度のタブ譜を生成するシステムが、ピアノの五線譜やギターのタブ譜を対象に開発されてきた [1][2]。これらのアプリケーションでは、楽譜が持つ演奏の難しさを数値化することで楽譜全体を編集するという手法を用いている。

本稿ではピアノの演奏練習を対象とし、演奏楽曲の楽譜とユーザによる実際の演奏データ (MIDI データ) を入力とした、奏者の技術に合わせたピアノ演奏練習支援システムを提案する。本システムでは、入力として与えられた楽譜

と演奏のデータを比較することでユーザの演奏ミスを抽出し蓄積する。このデータを元にしてユーザが苦手と思われるパートを判断し該当箇所の簡略化を行うことで、楽譜をユーザの演奏レベルに合ったものにするのができ、ユーザの得手不得手に細かく対応した演奏練習を実現することができる。また、著者らが過去に開発したシステム [3] で挙げられた、簡略化後の演奏が音楽的に不自然なものになることがあるという問題点を解決するため、任意箇所への弾き直し・弾き飛ばしを含む演奏に追従が可能な楽譜追跡と自動伴奏が可能な Eurydice[4] を用いて、簡略化を行った後の楽譜を演奏する際には削除された音符をシステムが補完して伴奏することとした。さらに、音符の密度や音域などといった楽譜の特徴量を各パートおよび小節ごとに抽出し、蓄積した演奏ミスのデータと併せたサンプルを教師データとして用いることにより、ある特徴量を持つ楽譜がそのユーザにとって簡略化されるべきかどうかを判断する学習モデルを得ることができる。このモデルに従う分類器を利用することによって、ユーザがシステムを通じて一度も演奏をしたことのないような楽譜に対しても、対象のユーザの演奏レベルに適した簡略化を実行することが可能となる。

2. ピアノ演奏練習支援システム

本稿で提案するピアノ演奏練習システムは、楽譜情報と

¹ 京都大学

a) tfukuda@sap.ist.i.kyoto-u.ac.jp

b) enakamura@sap.ist.i.kyoto-u.ac.jp

c) itoyama@kuis.kyoto-u.ac.jp

d) yoshii@kuis.kyoto-u.ac.jp



図 1 ピアノ演奏練習支援システムの概要図

MIDI ピアノを用いてユーザの実際の演奏から得られた演奏情報に基づいてユーザの演奏ミスを抽出し、それを利用することでユーザの苦手な楽譜の傾向を判断する。苦手であると判断された箇所については楽譜の簡略化を行い、演奏難度が高い楽譜をそのユーザの演奏レベルに適したものにす (図 1)。

入力 楽譜情報 (MusicXML)
演奏情報 (SMF)
出力 簡略化された楽譜
ユーザの演奏ミスデータ

なお、簡略化を行った後の楽譜をユーザに提示する際には、一般的なピアノ奏者が用いる五線譜形式であることが必須であるが、SMF (Standard MIDI File) 形式のファイルから五線譜上へ音符を自然に配置することは容易ではない。そこで、音符が五線譜上にどのように配置されているかが記載されている MusicXML 形式のファイルを入力とし、これを編集することでこの問題を回避することとした。

2.1 演奏ミスの抽出

本稿では、ユーザの演奏ミスとして次の 3 つを定義し、入力となる楽譜情報と演奏情報を用いることによって演奏ミスの抽出と分類を行う。

| | |
|---------------------|---------------------|
| extra note | 楽譜にない音符を演奏する誤り |
| missing note | 楽譜にある音符を演奏しない誤り |
| pitch error | 楽譜にある音符と異なる音を演奏する誤り |

具体的には、楽譜上でオンセット時刻が同じ演奏音符をまとめることでクラスタを生成し、クラスタ単位および音符単位の 2 つのレベルで left-to-right HMM (LR-HMM) を用いたアラインメントを行うことで演奏ミスの自動認識を行う。まず、クラスタ単位で弾き直しも考慮した従来法 [4] によるアラインメントを行う。これとは別に left-to-right HMM (LR-HMM) を用いたアラインメントも行う。これらのアラインメントの結果のクラスタラベルが異なる場合にはクラスタごと挿入誤りと考えられるので、含まれる音符を全て extra note とする。クラスタ単位のエラーがなかった場合には、そのクラスタ内の音符ごとでのエラー判定を行う。各クラスタの演奏音符に対して、楽譜音符と同じ音高で一番オンセット時刻が早いものを正解音符と判

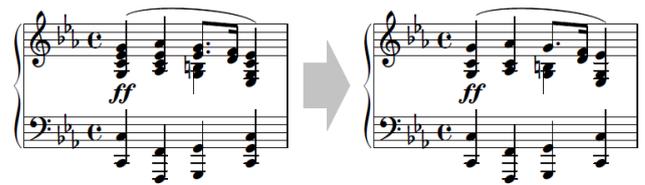


図 2 パターン 1 の例



図 3 パターン 2 の例

定し、それ以外の場合は以下の要領で extra note か pitch error かを判定する。まず、楽譜音符に同じ音高があり、正解音符とならなかったものは extra note とする。次に、1 オクターブと完全 4 度以内に楽譜音符がある場合には pitch error の候補とし、候補の空間で pitch 方向に関して LR-HMM によるアラインメントを行う。この際、音高誤りの確率を出力確率として用いる。この結果パス内にあるものは pitch error、それ以外の場合は extra note とする。最後に、演奏音符に対応する音符がない楽譜音符を missing note とする。

2.2 楽譜の簡略化

演奏ミスのデータからユーザが苦手なパートと判断された箇所について、該当楽譜をパターン分けすることで簡略化を行う。具体的には以下に挙げる 4 つのパターンに分類し、各パターンごとにルールを与えることで楽譜の簡略化を行う。今回のシステムでは、簡略化の範囲の最小単位を 1 小節としたため、簡略化を行う際には該当する小節全体を簡略化することとする。

2.2.1 パターン 1: 同時に弾く音数が多い楽譜

同時に弾く音数が多いパターンの楽譜では、和音の中で音符ごとに優先順位を設定し、順位の低いものを取り除いていくことによって簡略化を行う。具体的には、まず和音内の最も高い音は基本的にメロディラインに含まれているため音楽的に最も重要な音であり、次いで和音のコードを構成する根音が重要な音となる。和音の中でこの 2 音は特に重要な音で、原則として省略してはならないということが過去の研究 [5] でも述べられている。音符の優先順位は根音の 3 度上の音、5 度上の音と続いていくが、u これら

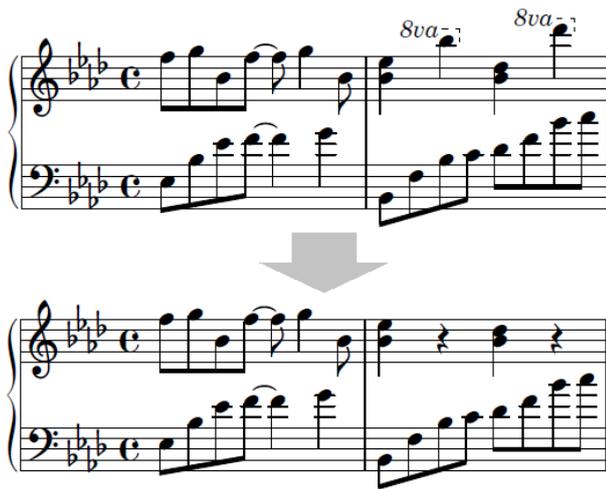


図 4 パターン 3-A の例

はいずれも上記で述べた 2 音ほど重要な音ではないため、必要に応じて省略することができる。このような優先度を設定することによって、演奏の難しい和音を比較的演奏の容易な 2 和音や 3 和音に簡略化することができる。パターン 1 の簡略化の例を図 2 に示す。

2.2.2 パターン 2：速い指の動きを要求する楽譜

速い指の動きを要求するパターンの楽譜については、一定以上のしきい値を越える速さの連続する音符を削除することで簡略化を行う。ここで用いるしきい値は、簡略化を行う箇所の楽譜情報を元に決定する。このパターンの簡略化の例を図 3 に示す。

2.2.3 パターン 3：離れた鍵盤を弾く楽譜

離れた鍵盤を弾く、いわゆる「跳躍」と呼ばれるパターンの楽譜については、さらに 2 通りの場合分けをすることができる。すなわち、

3-A 跳躍後すぐに跳躍前に演奏していた辺りに戻る場合

3-B 跳躍後にしばらく跳躍がない場合

と分類することができる。3-A のように離れた鍵盤を 1 つだけ弾かせるような場合には、その音を単純に削除し、3-B のように離れた音符の近くにしばらく音符があるような場合には、跳躍の周辺の音を簡略化することで対処する。この際の簡略化のルールはパターン 1 とパターン 2 の方法に従う。ここで、3-A のパターンの簡略化の例を図 4 に示す。

2.2.4 パターン 4：上記以外の楽譜

演奏ミスの多くは上記の 3 パターンによって分類することができると考えられるが、このどれにも当てはまらないようなものも存在し得る。例えば、演奏難度の低い簡単な楽譜はこれらのどれにも上手く分類することができない。このような場合には、パターン 1 やパターン 2 の手法をしきい値を低く設定した上で適用することによって対応する。すなわち、2 和音を単音にしたり、4 分音符以上の速さの音を削除したりすることによって多少無理にでも簡略化を行うことができる。

2.3 システムによる自動補完伴奏

楽譜の簡略化によって削除された音符をシステムに補完して演奏させるために、任意箇所への弾き直し・弾き飛ばしを含む演奏に追従可能な楽譜追跡と自動伴奏が可能なアプリケーション Eurydice を用いる。このアプリケーションは本来、複数のパートに分かれている音楽の一部のパートだけを演奏し、残りのパートをシステムが自動で追追しながら演奏をしてくれるというものである。本稿では、簡略化後の楽譜からなるパートと簡略化によって削除された音のみで構成されたパートからなる楽譜を Eurydice に与えることで、簡略化された箇所のシステムによる自動伴奏を実現した。これによって、ユーザが演奏するのは簡略化された楽譜だが、全体の演奏としては元の楽曲をそのまま演奏したかのように聞くことができるため、特定の楽曲でピアノの練習をしたいというユーザの意欲を保ったまま、演奏の練習を続けることができると考えられる。

3. 楽譜の事前簡略化

2 章で述べたように、本システムは楽譜情報と演奏情報を入力とすることで、簡略化された楽譜とユーザの演奏ミスデータを取得することができる。そこで、蓄積された演奏ミスのデータを楽譜の特徴量と併せて訓練サンプルとし、そのサンプルを教師データとして用いることにより、ある特徴量を持つ楽譜がそのユーザにとって簡略化されるべきかどうかを判断する学習モデルを得ることができる。さらに、このモデルに従う分類器を利用することによって、ユーザがシステムを通じて一度も演奏をしたことのないような楽譜に対しても、対象のユーザの演奏レベルに適した簡略化を事前に実行することが可能となる。

3.1 楽譜特徴量の抽出

学習に必要な訓練サンプルを作成するために、一般的なピアノの楽譜である五線譜の特徴量をいくつか定義した。今回定義した特徴量は縦の多重度 D_v 、横の多重度 D_h 、音域 E 、最も広い和音の広さ W_{max} 、パートの音符密度 N_b 、パート内の黒鍵が占める割合 R_{black} の 6 つであり、以下のように定義される。

$$D_v = \sqrt{\frac{1}{N_h} \sum_{k=1}^{N_h} n_k^2}$$

$$D_h = N_h$$

$$E = P_{highest} - P_{lowest} + 1$$

$$W_{max} = \max(P_{highest}^{(k)} - P_{lowest}^{(k)}, k = 1, \dots, N_h)$$

$$N_b = \frac{N}{b}$$

$$R_{black} = \frac{N_{black}}{N}$$

ここで、 N_h は小節内の (単音を 1 和音とみなした) 和音の

表 1 楽譜の特徴量抽出例

| パート | D_v | D_h | E | W_{max} | N_b | R_{black} |
|---------|-------|-------|-----|-----------|-------|-------------|
| 例 1(右手) | 3.69 | 1.25 | 18 | 9 | 4.50 | 0.39 |
| 例 1(左手) | 2.00 | 1.00 | 20 | 13 | 2.00 | 0.00 |
| 例 2(右手) | 1.00 | 2.33 | 5 | 4 | 2.33 | 0.43 |
| 例 2(左手) | 2.52 | 1.00 | 19 | 19 | 2.33 | 0.29 |



図 5 例 1



図 6 例 2

総数, n_k は k 番目の和音の中に含まれる音符の数, $P_{highest}$ と P_{lowest} および $P_{highest}^{(k)}$ と $P_{lowest}^{(k)}$ はそれぞれ小節内の最大音高と最小音高および k 番目の和音内の最大音高と最小音高, N はパート内の総音符数, b は小節の拍数, N_{black} は小節内の黒鍵の音符の数を表す. 例として, 図 5 と図 6 の各パートごとの楽譜から得られる特徴量を表 1 に示す. また, これらの特徴量を入力 \mathbf{x} として,

$$\mathbf{x} = (D_v, D_h, E, W_{max}, N_b, R_{black})$$

と表すこととする.

3.2 訓練サンプルを用いた教師あり学習

MIDI ピアノを用いて実際に演奏を繰り返し, 楽譜の特徴量と演奏ミスの有無をペアにしたサンプルの集合 $\{(\mathbf{x}_i, u_i) | i = 1, \dots, M\}$ を十分な数用意することによって, 学習モデルを得ることができる. ここで, u_i は演奏ミスが存在するときに 1, そうでないときに 0 が与えられる教師信号である. 本システムでは, 最適化法として確率的勾配降下法を採用したロジスティック回帰 [6] を用いて学習を行うこととする. すなわち, 入力 \mathbf{x} を与えた時の予測確率 y は,

$$y = \sigma(\mathbf{w}^T \phi(\mathbf{x}))$$

と表されるものとする. ここで $\sigma(z)$ は以下で定義されるシグモイド関数である.

$$\sigma(z) = \frac{1}{1 + e^{-z}}$$

予測確率が得られると, 次に示す確率的勾配降下法の更新式に従ってパラメータ \mathbf{w} を更新する.

$$\mathbf{w}^{(i+1)} = \mathbf{w}^{(i)} - \eta_i \cdot (y_n - u_n) \phi(\mathbf{x}) \quad (n = 1, \dots, M)$$

ここで, η_i は学習率を表し, 以下のように与えた.

$$\begin{cases} \eta_0 = 0.1 \\ \eta_i = \eta_{i-1} \times 0.95 (i = 1, \dots, M) \end{cases}$$

4. おわりに

本稿では, 楽譜情報とユーザの実際の演奏情報を基に演奏ミスがあった箇所を判定し, 該当箇所の簡略化を行うことにより, 自分の好みの楽曲でピアノの練習をすることを可能とするシステムを開発した. さらに, 本システムでは蓄積した演奏ミスのデータを楽譜の特徴量と共に用いることで, 対象ユーザがシステムを通じて一度も演奏したことがないような楽曲に対しても, そのユーザの演奏レベルに適した簡略化を行うことを可能とした.

今後の課題としては, 楽曲推薦機能の追加, 評価実験の実施を行っていきたいと考えている. まず, 楽曲推薦機能については, 3 章で述べたようにユーザの学習モデルを生成することができるため, 向上心のあるユーザに対しては, そのユーザの苦手なパートを多く含む楽曲をシステムが推薦してくれるという機能考えられる. 楽譜の解析や簡略化に必要な MusicXML 形式のファイルは, MuseScore*1 等ウェブ上で公開されているため, これらを集めて楽譜情報のデータベースを作成することでこのような機能を実現できると考えている. 評価実験に関しては, 演奏ミスのデータと特徴量から学習モデルを生成する手法は確立したが, 実際のデータを用いての評価は十分に行えていないため, 演奏データを集めて評価実験を行うことが急務である.

謝辞 本研究の一部は, JSPS 科研費 24220006, 26700020, 26280089, 16H01744, JST CREST, JST ACCEL の支援を受けた. また, Eurydice や演奏ミスデータの抽出に関する中村栄太氏の多大なるご協力に感謝します.

参考文献

- [1] 藤田 他: 習熟度を考慮した複数楽譜からのピアノ譜生成手法の提案, 情報処理学会研究報告音楽情報科学 (MUS) (2008).
- [2] 矢澤 他: ギター演奏者の習熟度に合わせた音響信号からのタブ譜自動生成, 情報研報, Vol. 2013-MUS-100, No. 17, pp. 1-6 (2013).
- [3] Fukuda T., et al.: A Score-Informed Piano Tutoring System with Mistake detection and Score Simplification, *Sound and Music Computing Conference* (2015).
- [4] 中村 他: 任意箇所への弾き直し・弾き飛ばしを含む演奏に追従可能な楽譜追跡と自動伴奏, 情報処理学会論文誌, Vol. 54, No. 4, pp. 1338-1349 (2013).
- [5] Hori G., et al.: Input-Output HMM Applied to Automatic Arrangement for Guitars, *JIP*, Vol. 21, No. 2, pp. 264-271 (2013).
- [6] Efon, B.: The efficiency of logistic regression compared to normal discriminant analysis, *Journal of the American Statistical Association* 70.352 (1975).

*1 <https://musescore.com/sheetmusic>