生成モデルに基づくギター楽譜からの演奏難易度推定

吉原　拓海† 　加藤　誠†† 　吉川　正俊†††

† 京都大学工学部情報学科　〒 606–8501 京都府京都市左京区吉田本町
†† 京都大学国際高等教育院データ科学イノベーション教育研究センター
††† 京都大学大学院情報学研究科
E-mail: †yoshihara@db.soc.i.kyoto-u.ac.jp, ††kato@dl.kuis.kyoto-u.ac.jp, †††yoshikawa@i.kyoto-u.ac.jp

あらまし　本研究では，楽譜の生成モデルに基づいて，ギター楽譜の演奏難易度を推定する手法を提案する．楽譜の生成モデルには，音符を音程と長さの生成が独立であること，また，和音の生成にギター特有の生成過程を仮定した，Note n-gram と呼ばれるモデルを使用する．実験では，提案手法である Note n-gram を用いたモデルに加え，一般的な n-gram を用いたモデルや楽譜の特徴量に基づく教師有り学習手法と比較を行い，提案手法の有効性を示した．

1. はじめに

近年，ギターの売り上げが低下している (注1)．しかし，楽器の価格が低下していることやフェス文化の流行などから，楽器を新たに始める人は少なからずいると考えられる．それでも販売数が低迷していることの原因には，始めたもののすぐに止めてしまう人や，ある程度上達したものの行き詰まり止めてしまう人が多くいることもあると考えられる。

その理由の一つには，楽譜だけを見ても初心者には難易度を判別しづらいため，自分のレベルにあった練習曲を見つけるのが難しいということが挙げられる。演奏の難易度には，速いフレーズが含まれているというだけでなく，隣接する 2 音が離れているため弾きづらい，リズムが複雑であるために高いリズム感が要求される，特殊な奏法が要求されるなどの要素があり，初心者がそれらの要素を加味した上で自分が取り組むべき曲を選択するのは容易ではない。また，数年の経験があったとしても，楽譜を精査しなければわからない場合もある．そのため初心者に限らず演奏者にとって，指導者がいない中で自分のレベルにあった曲を見つけることは難しい問題である。加えて，ピアノの演奏難易度推定については松原ら [9] や，米林ら [15] などがあるが，ピアノに比べて独学で習得されることの多いギターに対しての研究は，あまり行われていない．

本研究では，楽譜の生成モデルに基づいて，ギター楽譜の演奏難易度を推定する手法を提案する．これによって，自分が演奏しようとしている曲の難易度がわかるだけでなく，難易度による楽譜の検索が可能となる．そのため，自分の習熟度に適した難易度の楽譜を見つけることが容易になり，初心者の挫折防止や中級者以上の上達に繋がると考える．

具体的には，言語モデルを応用した音符モデルを作成し，これに基づいて楽譜の難易度を推定する手法を提案する．ただし，楽譜のデータセットは文書と比べて数が少ないため，少ないデータでも精度の出るような手法が必要とされる．一方，音符は単語に比べて種類が少なく，少量のデータでも正しく判定できるように思われる．しかし，一つの音符には音程と音価という二つの情報が含まれる上に，和音では同時に複数の音符が演奏されるため，単語よりは種類が少ないとはいえ，手法に工夫が必要となる．

そこで，難易度の推定に必要な情報を残しつつも学習不足となるのを避けるために，Note n-gram と名付けた独自のモデルを提案する．この Note n-gram は，音符の情報を音程列と長さの 2 つに分割し，それぞれ別の確率モデルを用いて生成確率を求めるモデルである．これには，それぞれの間に交互作用がないことを仮定している．

音符の生成確率に含まれる音程列 h_i は複数の音程から構成されるため（一般に，和音と呼ばれる），データの種類数が大きくなりやすく，少数の学習データを用いた学習の際に正確に推定できない可能性がある．より正確には，音程列の定義域は，音程の全体集合 H と同時に弾かれる音程数 n に対して H^n となる．このように和音で学習不足となる問題を解決しつつ，よりギターに適した生成確率を求めるために，音程列の生成確率モデルにおいてはギター特有の生成過程を仮定した，Chord-shape 確率モデルを考案した．この Chord-shape 確率モデルは，音程列の情報を和音の音数と和音の基本形，及びそこからの差の 3 つの情報に分割する．これらのモデルの導入によって，データセットが少ない中でも高い精度が得られると考えられる．

実験では，Note n-gram の有効性を示すために，一般的な n-gram を用いたモデルや楽譜の特徴量に基づく教師有り学習手法との比較を行った．さらに，Chord-shape モデルの有効性を示すために，Chord-shape モデル以外の和音を扱う手法として，Chord-split モデルという手法を導入し，これとの比較を行った．これによって，Note n-gram そのものが楽譜の情報を適切に分割することで少ないデータでも効率的に学習できるようになると同時に，Chord-shape 確率モデルが，ギターを対象としている本研究において適したモデルであることを示した．

本研究の貢献としてはまず，楽譜の演奏難易度の推定に対して言語モデルを応用したと言う点がある．加えて，言語ではな

(注1)：http://www.meti.go.jp/statistics/tyo/seidou/result/ichiran/08_seidou.html

く楽譜であるという点から，言語に対して用いられるモデルをそのまま流用するのではなく，Note n-gram という新たなモデルを提案したという点が挙げられる．さらに，その際に音符の情報をそのまま用いて生成確率を算出するのではなく，Chord-shape モデルという独自の生成確率を定義して，データセットが少量である場合に対応できるようにしたという点も挙げられる．

以下，本稿では 2 節で関連研究について述べ，本研究の位置づけを明らかにする．3 節では，音符モデル及び n-gram について具体的なアイデアとアルゴリズムについて記述し，4 節で実際に実験を行い，その性能を評価する．最後に 5 節で本稿をまとめる．

2. 関 連 研 究

本節では本研究に関連する過去の研究について述べる．関連する過去の研究事例としては，生成モデルや識別モデルを用いて文書の読解難易度を推定する研究や，生成モデルを用いた自動作曲についての研究，また，ピアノを対象として，楽譜の特徴量から難易度を識別する研究などに加えて，ギター楽譜の演奏難易度についての研究などがある．

2.1 文書の難易度推定

生成モデルを用いて文書の読解難易度を推定する研究としては，Thompson らの [3] や [2] などがある．これらの研究は共に，Smoothed Unigram Model と名付けられた独自のモデルに基づく言語モデルを使用して，Web 上の文書の対象学年を推定する．この Smoothed Unigram Model は，Uni-gram をベースとした統計的言語モデルである．このモデルにおいては，ある単語列 T がモデル G_i から生成される確率 $P(T|G_i)$ を，T の単語数 L，モデル G_i における単語 w の出現確率 $P(w|G_i)$，モデル G_i における単語 w の出現回数 $C(w)$，を用いて，

$$P(T|G_i) = P(L|G_i) \cdot L! \prod_{w \in T} \frac{P(w|G_i)^{C(w)}}{C(w)!}$$

と定義する．この $P(T|G_i)$ に対してベイズ則を適用した上で，G_i の出現する確率は i によらず一定であるため $P(G_i) = \frac{1}{N_G}$(ただし N_G はモデルの数) と表されるという仮定と，$P(L|G_i)$ は i によらず一定であるという仮定を置く．これによって，与えられた単語列 T を生成したモデルが G_i である確率 $P(G_i|T)$ の log を取った $L(G_i|T)$ は

$$L(G_i|T) = \sum_{w \in T} c(w) \log P(w|G_i) + \log Z$$

と表せるようになる．ここで，$\log Z$ は i によらない定数項であることから，Smoothed Unigram Model 同士での比較においては無視することのできる値である．全モデル G_i の中で，この $L(G_i|T)$ の値を最大化するモデル G_i が，単語列 T を生成したモデルとされる．

また，識別モデルを用いて文書の難易度を推定する研究としては，Petersen らの [5] などがある．この研究では，入力の文章から単語ごとの音節の平均値や文の長さの平均値などの特徴量を用いて，サポートベクトルマシンによって難易度を分類する．

これらの研究は，難易度についてのものではあるが対象が言語であるため，言語に限定された手法が用いられており，そのままでは楽譜への利用は難しいと考えられる．

2.2 生成モデルを用いた自動作曲

生成モデルを用いた自動作曲についての研究としては，白井らの [14] や田村らの [12]，Yang らの [7]，川村らの [11] などがある．[14] では，学習を行う際に確率的に接尾辞木の一つ上の文脈をカウントすることでスムージングを行う Hierarchical Pitman-Yor Language Model というモデルを可変長 n-gram に拡張した VPYLM というモデルを用いる．この VPYLM に対して，歌詞とコードを入力として与え，出力としてメロディを得る．[12] では，音の長さと高さを分けて Bi-gram 確率を計算し，それを HMM の遷移確率として用いることで楽曲を生成する．[11] では， 音の長さの列，すなわちリズムが音程の生成に制約を与えるとして，リズムを生成した上でそのリズムに適した音程列を生成する．

これらの研究では，楽曲を生成モデルを用いてモデル化している．しかしながら，ここで用いられているモデルは自動作曲を目的としたモデルであるため，コード進行や歌詞のモーラ数 (注2) を制約として持つなど，作曲に特化したものとなっている．そのため，そのままでは難易度推定への利用は難しい．

2.3 ピアノ楽譜の難易度推定

ピアノ楽譜の演奏難易度の推定を含む研究としては，松原らの [9] や Chiu らの [1]，Holder らの [4]，藤田らの [13]，宮川の [8] などがある．[9] では，ピアノを対象としていることから，隣接する 2 音間の鍵盤上での距離を求め，その各距離の出現頻度を楽譜の特徴量としていた．これは，隣接する 2 音間の関係から難易度を推定する手法ではあるが，テンポや音符の長さなどのリズムの要素を無視している．また，ギターはピアノのように音程に対して線形に物理的距離が離れるものではない．このため，この研究で用いられている特徴量はピアノに固有のものであり，ギターへの応用は難しい．

[1] では，楽譜から得られる特徴量として，楽譜に用いられている音程の平均値やテンポなどの従来の特徴量 10 個と，テンポと音符の種類から割り出される実際の音符の速さの平均である Playing speed や，左右の手がどれだけ離れているかを表す Hand stretch などの [6] で提案されていた特徴量のうち，難易度推定で用いることができると考えられた 8 個の計 18 個用いて，回帰モデルを用いて分類する．これは，楽譜から取れる様々な特徴量を用いようとしてはいるが，連続する音の関係をとる特徴量がほとんど存在していない．そのため，同じテンポで同じ音符を用いているものの並びが違うことで難易度が異なるような楽譜に対しても，近い値を出してしまいうる．また，ピアノ固有の特徴量が多く含まれることから，そのままでのギターへの応用は難しいと考えられるが，ピアノに対しては有効

(注2)：音節数に，長音，促音，撥音の数を加えたもの．歌詞に対して割り当てられる音の数である．

な手法であった．そこで，この手法をベースラインとして全特徴量のうちギターでも利用可能なもののみを用いて，提案手法と比較するとことする．

2.4 ギターの演奏難易度推定

ギターを対象として楽譜の演奏難易度を扱う研究としては，森田らの [10] や，矢澤らの [16] などがある．

[10] では，推定対象の楽譜をギターの経験のある被験者に演奏させ，その演奏と楽譜とのズレを入力として用いて，難易度を決定していた．

[16] では演奏時の運指推定の一部として，弦を押さえる際の難易度を推定していた．この際には楽譜の特徴量として，指を広げる幅を示すフレット幅，同時に使用指の数，ギターのどの位置を押さえているかを示すフレット位置，一本の指で複数弦を押さえるバレーという技法の有無の 4 つを用いて計算を行なっていた．

[10] における手法については，楽譜のみからの難易度の推定を目的とする本研究では利用することが難しい．また, [16] における手法については，弦を押さえる際の難易度についてのみの手法であることに加えて，五線譜に含まれていない情報も用いているため，そのままの形での本研究での利用は難しいと考えた．

これらの研究を元に本研究では，連続する音によって難易度が変化するような場合に適切に対応でき，かつギターに適した手法を提案することを目的とする．

3. 提 案 手 法

本節では，まず 3.1 節節で本論文における楽譜に関する定義を述べ，その後 3.2 節節で問題設定を行う．その上で 3.3 節節で設定した問題に対するアプローチを説明した後に，3.4 節節で比較手法である n-gram モデルについて述べ，3.5 節節，3.6 節節で提案手法である Note n-gram モデルについて述べる．

3.1 定　　義

以降で用いる記号を以下のように定義する

- $\boldsymbol{s} = (w_1, w_2, ...)$: 音符列
- $w_i = (\boldsymbol{h}_i, l_i)$: 音符

音程列 $\boldsymbol{h}_i$ と長さ l_i からなるタプルとする．

- $\boldsymbol{h}_i = (h_{i,1}, h_{i,2}, ...)$: 音程列

同時に鳴らす音程を昇順に並べた列とする．

- $|\boldsymbol{h}_i|$: $\boldsymbol{h}_i$ に含まれる音の数
- $h_{i,j}$: 音程 midi のノートナンバーで表された音程とする．0 以上 127 以下の整数であり，各音程ごとに値が定められている．五線譜との対応表の一部を図 1 に示す．これは，五線譜に書かれた各音程とそれに対応する値が書かれているものであり，例えばト音記号の下のドには 48，ファには 53 などとなっている．
- v_i: 音価

四分音符を 12 としたときのその音符の相対的な長さとする．主要な各音符の値を図 2 に示す．この図に示した通り，四分音符の半分の長さである八分音符の音価は 6，三分の一の長さである三連符の音価は 4 などとなっている．

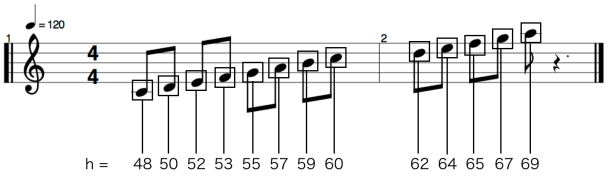

図 1: midi のノートナンバーと五線譜との対応

名称	全音符	半音符	四分音符	八分音符	十六分音符	三連符
図						
長さ	48	24	12	6	3	4

図 2: 音価の定義

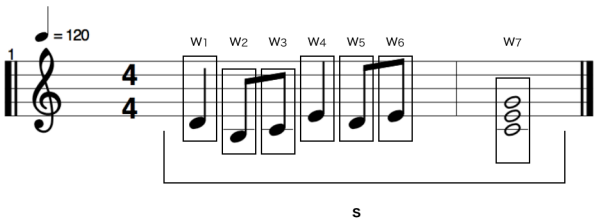

図 3: 定義の例

- l_i: 長さ

音の継続時間であり，単位は秒である．曲のテンポ t と音価 v_i から，以下の式で定義される．

$$l_i = \frac{60}{t} \times \frac{v_i}{12} = \frac{5v_i}{t}$$

- M: 音符モデル
- $C(w)$: コーパス内での音符 w の出現回数
- $P(\boldsymbol{s}|M)$: 音符列 $\boldsymbol{s}$ がモデル M から生成される確率
- $P(w|M)$: 音符 w がモデル M から生成される確率

図 3 に例を挙げた．音符の列全体が $\boldsymbol{s}$，縦に並んでいる音符の各々が w_i である．ここで w_1 を例にとると，$\boldsymbol{h}_1 = (50)$ となり，$l_1 = \frac{5v}{t} = \frac{5\times12}{120} = 0.5$ となる．また，w_7 では，$\boldsymbol{h}_7 = (48, 52, 55)$ となり，$l_7 = \frac{5v}{t} = \frac{5\times48}{120} = 2$ となる．

3.2 問 題 設 定

本研究における問題を以下のように設定する．

入力 ： 音符列 $\boldsymbol{s}$, 楽譜モデル集合 $\mathcal{M} = \{M_1, M_2, ...\}$

出力 ： $\boldsymbol{s}$ を生成した確率のもっとも高い楽譜モデル M^*

これは以下のように定式化できる．

$$M^* = \underset{M_i \in \mathcal{M}}{\operatorname{argmax}} P(M_i|\boldsymbol{s}) \tag{1}$$

3.3 生成モデルに基づく楽譜の難易度推定手法

音符列 $\boldsymbol{s}$ が与えられた時に，モデル M_i がそれを生成した確率 $P(M_i|\boldsymbol{s})$ はベイズの法則から，

$$P(M_i|\boldsymbol{s}) = \frac{P(\boldsymbol{s}|M_i)P(M_i)}{P(\boldsymbol{s})} \tag{2}$$

と表される．

ここで，$P(\boldsymbol{s})$ は $\boldsymbol{s}$ によらず一定であるという仮定をおくこ

とでこの項は無視することができる様になり，確率の大小の順序は log を取っても変化しないため，式 1 は

$$M^* = \underset{M_i \in \mathcal{M}}{\operatorname{argmax}}(\log P(M_i|\boldsymbol{s})) = \underset{M_i \in \mathcal{M}}{\operatorname{argmax}}(\log P(\boldsymbol{s}|M_i) + \log P(M_i)) \quad (3)$$

と変形できる．これによって，$P(\boldsymbol{s}|M_i)$ が求められれば問題は解くことができるようになる．ただし，この確率 $P(\boldsymbol{s}|M_i)$ にはさまざまなものが考えられ，これについて以下の 3.4 節節，及び 3.5 節節で述べる．また，一般に生成モデルを用いる際には，未知の単語，すなわち未知語の出現によって確率が 0 になるのを避けるため，未知語に対して一定の確率を与えるスムージングという手法が用いられる．本研究においては未知の音符全てに対して 0.01%の確率を与えることとする．また，自然言語を用いた研究では，英語での冠詞や be 動詞，日本語での語尾のような，全モデルに一様に出現する単語，すなわちストップワードを事前に削除するが，楽譜における音符の出現に関しては，無視することのできる出現は存在しないものとし，ストップワードの削除は行わないこととする．

3.4 n-gram

本節では，本研究における n-gram の定義と，その限界について説明する．

n-gram では，音符 w を音程列 $\boldsymbol{h}$ と長さ l のタプルのままで各モデル M での出現頻度を求め，それを生成確率とする．よって，n-gram に基づくモデル M_i における音符 w_j の生成確率 $P(w_j|w_{j-1}, w_{j-2}, ..., w_{j-n+1}, M_i)$ は，モデル M_i における $w_{j-n+1}, ..., w_{j-2}, w_{j-1}, w_j$ の連続した出現の回数 $C(w_{j-n+1}, ..., w_{j-2}, w_{j-1}, w_j|M_i)$ を用いて，

$$P(w_j|w_{j-1}, w_{j-2}, ..., w_{j-n+1}, M_i) = \frac{C(w_{j-n+1}, ..., w_{j-2}, w_{j-1}, w_j|M_i)}{C(w_{j-n+1}, ..., w_{j-2}, w_{j-1}|M_i)}$$

と定義され，これを用いてモデル M_i での音符列 $\boldsymbol{s} = [w_1, ..., w_{|\boldsymbol{s}|}]$ の生成確率 $P(\boldsymbol{s})$ は以下のように求められる．

$$P(\boldsymbol{s}|M_i) = \prod_{j=1}^{|\boldsymbol{s}|} P(w_j|w_{j-1}, w_{j-2}, ..., w_{j-n+1}, M_i)$$

これと式 3 から，n-gram において問題は以下のように変形できる．

$$M^* = \underset{M_i \in \mathcal{M}}{\operatorname{argmax}}(\sum_{j=1}^{|\boldsymbol{s}|} (\log P(w_j|w_{j-1}, w_{j-2}, ..., w_{j-n+1}, M_i)) + \log P(M_i))$$

この n-gram というモデルは，楽譜を演奏する際の難易度は，各々の音符のみではなくそれらの組み合わせ，すなわち前後の音符との関係によって決定されるということに着目している．これによって，それぞれ別々で弾いているときには簡単でも，連続して弾いた場合に難易度が上がるという状況に対応できると考えた．

しかし，このモデルでは，音符を音程列と長さのタプルのままで扱うため，少ないデータセットに対しては各データの出現回数が十分に取れない可能性がある．これは，全く同じ音程列と長さでなければ同じ音符の出現とされないことによる．加えて，例えば音程よりも長さによって難易度が上がっている，などという状況にも対応しづらい．そこで，こういった問題を解決することを目的として考案した Note n-gram について 3.5 節節で述べる．

3.5 Note n-gram

本節では，提案手法である Note n-gram のアイデア及び定義を説明する．

Note n-gram は，音符の情報を音程列と長さに分割した上で，その各々の生成確率を独立に求めるモデルである．これは，音程列と長さの二つの要素の間に交互作用が存在しないことを仮定する代わりに，データセットが自然言語と比べて少ない楽譜の学習において，学習不足を軽減することを目的としたモデルである．すなわち，この Note n-gram というモデルにおける i 番目の音符 w_i の生成確率 $P(w_i|w_{i-1}, w_{i-2}, ..., w_{i-n+1})$ は，音程列と長さの生成確率 $P(\boldsymbol{h}_i|\boldsymbol{h}_{i-1}, \boldsymbol{h}_{i-2}, ..., \boldsymbol{h}_{i-n+1})$, $P(l_i|l_{i-1}, l_{i-2}, ..., l_{i-n+1})$ を用いて，

$$P(w_i|w_{i-1}, w_{i-2}, ..., w_{i-n+1}) = P(\boldsymbol{h}_i|\boldsymbol{h}_{i-1}, \boldsymbol{h}_{i-2}, ..., \boldsymbol{h}_{i-n+1}) \cdot P(l_i|l_{i-1}, l_{i-2}, ..., l_{i-n+1})$$

となる．これによって，3.4 節節で述べたような，データが足りず学習不足になりやすいという問題や，音程あるいは長さの一方のみに大きく依存した難易度にも対応できるようになると考える．

3.6 Note n-gram における音程列の生成確率

本節では，Note n-gram における音程列の生成確率 $P(\boldsymbol{h}_i|\boldsymbol{h}_{i-1}, \boldsymbol{h}_{i-2}, ..., \boldsymbol{h}_{i-n+1})$ について述べる．Note n-gram における音程列の生成確率を定義する際に，仮に単純に出現頻度を用いた場合，全く同じ和音しか出現として扱うことができない．このとき，各和音の出現数が十分に得られず，学習不足となると考えられる．これを避けるために，単純な出現頻度を用いるのではなく，独自の生成確率を用いることとする．具体的には，二つの和音の間で遷移する際の遷移確率を求める際に，構成する単音に分離した上でそれぞれの間の遷移確率を用いる Chord-split モデル，及びギター特有の生成過程を用いる Chord-shape モデルの二種類を考える．

3.6.1 Chord-split モデル

本節では Chord-split モデルの 2 つのモデルについて述べる．Chord-split モデルにおいては，前の和音の構成音の中で最も遷移確率が高いものから遷移したと仮定する Chord-split-maximum モデルと，前の和音の構成音からの遷移確率の平均を用いる Chord-split-average モデルの二つを用意する．それぞれにおける生成確率を，$\boldsymbol{h}_{i-n+1}, ..., \boldsymbol{h}_{i-1}, \boldsymbol{h}_i$ という音程列の連続した出現があったときに，Chord-split-maximum モデルでは

$$P(\boldsymbol{h}_i|\boldsymbol{h}_{i-1}, ..., \boldsymbol{h}_{i-n+1}) = \prod_{n_i=1}^{|\boldsymbol{h}_i|} \max_{n_{i-n+1}=1}^{|\boldsymbol{h}_{i-n+1}|} ... \max_{n_{i-1}=1}^{|\boldsymbol{h}_{i-1}|} P(h_{i,n_i}|h_{(i-1),n_{i-1}}, ..., h_{(i-n+1),n_{i-n+1}})$$

とし，Chord-split-average モデルでは

$$P(\boldsymbol{h}_i|\boldsymbol{h}_{i-1},...,\boldsymbol{h}_{i-n+1}) = \prod_{n_i=1}^{|\boldsymbol{h}_i|}\frac{1}{|\boldsymbol{h}_{i-n+1}|}\sum_{n_{i-n+1}=1}^{|\boldsymbol{h}_{i-n+1}|}...\frac{1}{|\boldsymbol{h}_{i-1}|}\sum_{n_{i-1}=1}^{|\boldsymbol{h}_{i-1}|} P(h_{i,n_i}|h_{(i-1),n_{i-1}},...,h_{(i-n+1),n_{i-n+1}})$$

とする．

しかしながら，このモデルはギターにおける音程列の生成過程に適していないため，ある音程列の生成確率を求める際に，近しい難易度にある別の出現を利用することができない．例えば，$\boldsymbol{h}_{i-1} = (60, 64, 67)$ から $\boldsymbol{h}_i = (64, 67, 71)$ へ移る難易度と $\boldsymbol{h}_{j-1} = (61, 65, 68)$ から $\boldsymbol{h}_j = (65, 68, 72)$ へ移る難易度は，ギターにおいては近い関係にある．しかしながら，Chord-split モデルを用いた場合，Note Bi-gram におけるこれらの音程列の連続した出現 $\boldsymbol{h}_{i-1}, \boldsymbol{h}_i$ と $\boldsymbol{h}_{j-1}, \boldsymbol{h}_j$ の生成確率を求める際に用いられる確率は以下の通りとなる．$\boldsymbol{h}_{i-1}, \boldsymbol{h}_i$ に対しては $P(60, 64)$，$P(60, 67)$，$P(60, 71)$，$P(64, 64)$，$P(64, 67)$，$P(64, 71)$，$P(67, 64)$，$P(67, 67)$，$P(67, 71)$ が用いられ，$\boldsymbol{h}_{j-1}, \boldsymbol{h}_j$ に対しては，$P(61, 65)$，$P(61, 68)$，$P(61, 72)$，$P(65, 65)$，$P(65, 68)$，$P(65, 72)$，$P(68, 65)$，$P(68, 68)$，$P(68, 72)$ が用いられる．これらの間に共通するものがないことからわかる通り，Chord-split モデルにおいては $P(\boldsymbol{h}_i|\boldsymbol{h}_{i-1})$ と $P(\boldsymbol{h}_j|\boldsymbol{h}_{j-1})$ は相関のない確率となる．そのため，一方の出現をもう一方の生成確率を求めるときに用いることはできない．この問題を解決するために考案した Chord-shape モデルについて 3.62 節節で述べる

3.6.2 Chord-shape モデル

本節では Chord-shape モデルについて述べる．このモデルは，ギターにおいて，同じ手の形のままで移動平行移動させることによって異なる和音を演奏することが多いという特徴に着目したモデルである．ピアノでは音程から弾く鍵盤が一意に定まるため，五線譜の情報で運指が推定できるが，ギターでは同じ音程を出すことのできる部分が複数ある．そのため，本研究では以下のようなモデル化を行い，運指の情報を擬似的に表すこととする．

まず，和音の基本形というものを導入する．これは，ギターで弦を押さえる際の手の形を擬似的に表すことを目的としたものである．この基本形 $\boldsymbol{b}$ の全体集合 $\mathcal{B}$ は以下のように示される．

$$\mathcal{B} = \cup_{i=1}^{6}\mathcal{B}_i$$

ただし，$\mathcal{B}_i$ は，音程の全体集合 H と，$\boldsymbol{b}$ の最小値 $\min(\boldsymbol{b})$ を用いて，以下のように表される集合である．

$$\mathcal{B}_i = \{\boldsymbol{b}|\boldsymbol{b} \in H^i \wedge \min(\boldsymbol{b}) = 0\}$$

このとき，$\boldsymbol{h}_i$ は，これに対応する基本形 $\boldsymbol{b}_i$ と，次元数 $|\boldsymbol{h}_i|$ で全次元が 1 のベクトル $\boldsymbol{e}_i$，$d_i = \min(\boldsymbol{h}_i)$ を用いて以下のように表される．

$$\boldsymbol{h}_i = \boldsymbol{b}_i + d_i \cdot \boldsymbol{e}_i$$

ここでさらに，$n_i = |\boldsymbol{h}_i|$ をおく．各々の生成確率を，$P(n_i)$，$P(\boldsymbol{b}_i|n_i, \boldsymbol{b}_{i-1}, \boldsymbol{b}_{i-2}, ..., \boldsymbol{b}_{i-n+1})$，$P(d_i|d_{i-1}, d_{i-2}, ..., d_{i-n+1})$ と定義する．これは基本形 $\boldsymbol{b}_i$ が n_i に依存することと，n_i はそれより前の $n_{i-1}, n_{i-2}, ...$ には依存しないという仮定をおいていることによる．

これらを用いて，$\boldsymbol{h}_{i-n+1}, ..., \boldsymbol{h}_{i-1}, \boldsymbol{h}_i$ という音程列の連続した出現があったときに，その生成確率を Chord-shape モデルでは，

$$P(\boldsymbol{h}_i|\boldsymbol{h}_{i-1},...,\boldsymbol{h}_{i-n+1}) = P(n_i)\cdot P(\boldsymbol{b}_i|n_i, \boldsymbol{b}_{i-1}, \boldsymbol{b}_{i-2}, ..., \boldsymbol{b}_{i-n+1}) \cdot P(d_i|d_{i-1}, d_{i-2}, ..., d_{i-n+1})$$

とする．これは，n と d，$\boldsymbol{b}$ と d それぞれに相互作用が存在しないという仮定を置いていることによる．

4. 実　　験

本節では，まず 4.1 節節で実験で用いたデータセットについて説明し，4.2 節節で提案手法と比較手法について説明する．その後，4.3 節節で実験方法について説明した上で，4.4 節節で実験結果とその結果に対しての考察を述べる．

4.1 データセット

本節では，実験に用いたデータとその処理について説明する．

本研究では学習用データとして，ギターの教則本を執筆している小林 信一氏の著書から，「地獄のベーシック・トレーニング・フレーズ」(注3) (以下ベーシックトレーニング) と，「地獄のメカニカル・トレーニング・フレーズ」(注4) (以下メカニカルトレーニング) の 2 冊を用いた．この 2 冊は，フレーズごとに難易度付けされており，同一のシリーズで複数出版されている上に知名度もあるため，信頼がおけ，基礎的なフレーズをカバーしている教則本が少ない中で，地獄のベーシック・トレーニング・フレーズには，基礎的なフレーズから含まれていたため，この 2 冊を学習用データに用いた．

この 2 冊をスキャンして PDF 化した後，KAWAI 社のスコアメーカー Plutinum の楽譜認識機能を用いて MusicXML に変換した．この際に，小節線のずれや拍子の誤認識等は排除したものの，調合の誤検出や五線の誤検出による音程の誤認識や，音符の種類の誤認識，特殊な省略記号が用いられていることによる誤認識は排除されずに残った．

こうして得られた MusicXML のデータをパースし，以下の処理を施した．

（1） 0 番目のパート，すなわち五線譜の音符の情報を抜き出す

（2） 小節ごとで 4 拍丁度，すなわち音価の総和が 48 でないものについては，状況に応じて以下の処理を施す

- 音符が一切ない場合は，誤検出された小節として無視する
- 4 拍に満たないものについては，最後の音符が省略記号によって省略されたものとして，4 拍丁度になるように最後の

(注3)：https://www.rittor-music.co.jp/product/detail/3113217114/
(注4)：https://www.rittor-music.co.jp/product/detail/3103217101/

表 1: ベーシックトレーニングのデータ量

難易度	フレーズ数	小節数	音符数
1	40	160	1471
2	28	112	773
3	16	64	670
4	15	60	845

表 2: メカニカルトレーニングのデータ量

難易度	フレーズ数	小節数	音符数
1	37	148	1768
2	41	164	2084
3	70	280	3744
4	52	208	2752
5	40	160	2256

音符を加える．

- 4 拍を超えるものについては，音符の誤検出によるものと考えられるが，そのままでもさほど大きな問題にならないと考えてそのままとした

（3） 4 小節を 1 フレーズとしてフレーズ単位にまとめ，各フレーズにメタデータとして，楽譜認識で読み取れなかったテンポと，難易度の情報を加える

以上の処理を行った後の各教則本の，難易度ごとのデータ量は表 1,2 に示す通りである．

この 2 冊の難易度を一つに扱うために，5 年以上のギターの経験者 5 人を対象に，2 冊の難易度の差についてアンケートをとったところ平均が 3 となったため，メカニカルトレーニング由来のフレーズの難易度に 3 を加えて扱うこととした．また，難易度の数値のまま扱うのではなく，全体を難フレーズと易フレーズとに分けることにした．これは，各難易度ごとにモデルを作成した時にデータ量が少ないことで学習不足になるのを防ぐ為である．これについても同じく 5 年以上のギターの経験者に難フレーズと易フレーズの境界についてアンケートをとったところ平均が 6 となったので，難易度 6 未満のフレーズを易フレーズ，それ以上のフレーズを難フレーズとすることにした．この様な処理を行った結果，難フレーズが 162 フレーズ，易フレーズが 177 フレーズとなり，これらのフレーズから生成される，易フレーズモデルと難フレーズモデルの 2 モデルを用いて以下の実験を行うこととする．

4.2 比較手法

提案手法である Note n-gram に対する比較手法として，通常の n-gram，及び [1] で提案されている特徴量に基づく教師有り学習を用いる．n-gram では Uni-gram，Bi-gram，Tri-gram の 3 つを用い，Note n-gram では，Note Uni-gram(以下 NUG)，Note Bi-gram(以下 NBG)，Note Tri-gram(以下 NTG)，の 3 つに対してそれぞれ Chord-shape モデル，Chord-split-average モデル，Chord-split-maximum モデル，を適用した全 9 モデルを用いる．

教師あり学習については, [1] で提案されている特徴量のうち，ピアノ固有のものをのぞいた全 11 個の特徴量に基づくサポートベクトルマシンを用いることとし，全ての特徴量を用いて分類するモデル 1 個と，各特徴量 1 つのみを用いて分類するモデル 11 個の全 12 モデルを用いた．実際に用いた特徴量を表 3 に示す．

表 3: 特徴量

名称	説明	定義
Playing speed (PS)	各音符の長さの平均．長さは 3 節で定義したものを用いる．	$PS = \frac{1}{\lvert \boldsymbol{s} \rvert} \frac{5}{tempo} \sum_{i=i}^{\lvert \boldsymbol{s} \rvert} v_i$
Pitch entropy (PE)	各音符の音程のエントロピー	$PE = -\sum_{i=1}^{\lvert \boldsymbol{s} \rvert} \sum_{j=1}^{\lvert \boldsymbol{h}_i \rvert} p(h_{i,j}) \log p(h_{i,j})$
Hand displacement rate (HDR)	演奏時に手を離す割合．ここでは，ある音符が直前の音符と異なる音程である割合	$HDR = \frac{1}{\lvert \boldsymbol{s} \rvert} \sum_{i=1}^{\lvert \boldsymbol{s} \rvert} a_i$ ただし， $a_i = 1(\boldsymbol{h}_i \neq \boldsymbol{h}_{i-1})$ $= 0(\boldsymbol{h}_i = \boldsymbol{h}_{i-1})$ とする．
Polyphony rate(PR)	全ての音符における，和音の割合	$PR = \frac{1}{\lvert \boldsymbol{s} \rvert} \sum_{i=1}^{\lvert \boldsymbol{s} \rvert} p_i$ ただし， $p_i = 1(\lvert \boldsymbol{h}_i \rvert > 1)$ $= 0(\lvert \boldsymbol{h}_i \rvert \leqq 1)$ とする．
Average pitch value (APV)	全ての音符の音程の平均．音程は 3 節で定義したものを用い，和音は各音に分けて用いることとする．	$APV = \frac{1}{\sum_{i=1}^{\lvert \boldsymbol{s} \rvert} \lvert \boldsymbol{h}_i \rvert} \sum_{i=1}^{\lvert \boldsymbol{s} \rvert} \sum_{j=1}^{\lvert \boldsymbol{h}_i \rvert} h_{i,j}$
Average note duration (AND)	全ての音符の音価の平均	$AND = \frac{1}{\lvert \boldsymbol{s} \rvert} \sum_{i=1}^{\lvert \boldsymbol{s} \rvert} v_i$
Deviation of pitch value (DPV)	全ての音符の音程の標準偏差．音程の扱いについては APV と同様にする．	$DPV = \sqrt{\frac{1}{\sum_{i=1}^{\lvert \boldsymbol{s} \rvert} \lvert \boldsymbol{h}_i \rvert} \sum_{i=1}^{\lvert \boldsymbol{s} \rvert} \sum_{j=1}^{\lvert \boldsymbol{h}_i \rvert} (h_{i,j} - APV)^2}$
Deviation of note duration (DND)	全ての音符の音価の標準偏差	$DND = \sqrt{\frac{1}{\lvert \boldsymbol{s} \rvert} \sum_{i=1}^{\lvert \boldsymbol{s} \rvert} (v_i - AND)^2}$
Pitch range (PRG)	最高音と最低音との差	$PRG = \max_{i=1}^{\lvert \boldsymbol{s} \rvert} \max_{j=1}^{\lvert \boldsymbol{h}_i \rvert} h_{i,j} - \min_{i=1}^{\lvert \boldsymbol{s} \rvert} \min_{j=1}^{\lvert \boldsymbol{h}_i \rvert} h_{i,j}$
Shortest rhythm value (SRV)	もっとも短い音価	$\min_{i=1}^{\lvert \boldsymbol{s} \rvert} v_i$
Tempo	テンポ	(楽譜から抽出する)

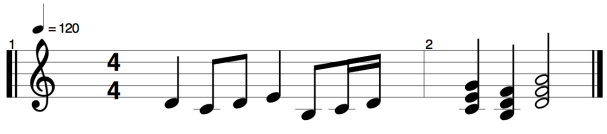

図 4: 特徴量の例に用いた楽譜

4.3 実験方法

本節では，実験の方法について述べる．

4.1 節節で設定したデータセットを用いて 5-fold cross-validation を行う．これはデータの 80%を学習用データとして学習し，残りの 20%をテストデータとしてテストを行うという検証を 5 回，テストデータに重複が起こらないように行うというものである．

4.2 節節で挙げた各モデルにおいて，学習データを用いて学習したのち，テストデータの各音符列に対して，易フレーズモデルと難フレーズモデル各々からの生成確率を求め，そのオッズ比の log をとったものを出力 $O(\boldsymbol{s})$ とする．すなわち，入力を $\boldsymbol{s}$，易フレーズモデルを M_e，難フレーズモデルを M_d とすると，出力 $O(\boldsymbol{s})$ は

$$\begin{aligned} O(\boldsymbol{s}) &= \log \frac{P(\boldsymbol{s}|M_d)}{P(\boldsymbol{s}|M_e)} \\ &= \log P(\boldsymbol{s}|M_d) - \log P(\boldsymbol{s}|M_e) \end{aligned}$$

となる．

この $O(\boldsymbol{s})$ に対して，$O(\boldsymbol{s}) > 0$ であるような音符列 $\boldsymbol{s}$ は難フレーズ，$O(\boldsymbol{s}) \leqq 0$ であるものは易フレーズとして推定し，この正確性すなわち Accuracy での比較をすると同時に，$O(\boldsymbol{s})$ を用いて AUC を求め，これら二つの値を用いて各モデルを比較する．

4.4 結果と考察

本節では，実験結果とその結果からの考察を述べる．結果をまとめたものが表 4 である．また，実験した際の，Chord-shape モデルに基づいた NBG における，易フレーズモデルと難フレーズモデルでの $P(d)$ の分布を図**??**，**??**に示す．

表 4 から，n-gram より Note n-gram の方が Accuracy と AUC どちらの値においても優位であることが読み取れる．ここから Note n-gram において音符の情報を分割することでデータの種類数が減り，教則本 2 冊のみという少ないデータセットでも効果を発揮できるようになったと考えられる．これは，n-gram においては Bi-gram の性能が最高であるのに対して，Note n-gram においては NTG すなわち Tri-gram で最高値を出していることからも読み取れる．

さらに，Chord-split モデルより Chord-shape モデルの方が Accuracy と AUC どちらの値においても優位であることが読み取れる．これは，Chord-shape モデルにおいてギター特有の生成過程を用いたことで，よりギターにおける難易度推定に適したモデルとなったことによると考えられる．ここで，図**??**，**??**に示した $P(d)$ の分布からも Chord-shape モデルについて考察する．図**??**，すなわち難フレーズモデルでは d_{i-1}，d_i が共に比較的大きい数字である部分に多く分布しており，加えて，同じ数字同士での分布は多くない．対して，図**??**，すなわち易フレーズモデルでは d_{i-1}，d_i が共に比較的小さい数字である部分に多く分布しており，加えて，同じ数字同士での分布が多い．この結果は，d の定義から音楽的に考察すると，難フレーズにおいては高い音が多く分布しているのに加え，同じ音同士での遷移がさほど多くないのに対して，易フレーズにおいては低い音が多く分布しているのに加え，同じ音同士での遷移が多いことを表している．これは，ギター楽譜の難易度に対する直感的な理解と合致する．このことからも，Chord-shape モデルがギターにおける音程列の生成過程に適していることがわかる．

表 4: 提案手法及び比較手法による実験結果

Model		Accuracy	AUC
Uni-gram		64.3%	0.721
Bi-gram		70.5%	0.755
Tri-gram		66.5%	0.731
NUG	Chord-shape	70.5%	0.744
	Chord-split-average	70.2%	0.727
	Chord-split-maximum	70.2%	0.727
NBG	Chord-shape	75.4%	0.805
	Chord-split-maximum	72.6%	0.784
	Chord-split-maximum	72.3%	0.781
NTG	Chord-shape	77.2%	0.842
	Chord-split-maximum	75.1%	0.820
	Chord-split-maximum	75.1%	0.819
Feature Based Classifier		56.6%	0.626

表 5: 各特徴量のみを用いた分類器による実験結果

Feature	Accuracy	AUC
Playing speed (PS)	55.1%	0.598
Pitch entropy (PE)	60.0%	0.689
Hand displacement rate (HDR)	55.1%	0.649
Polyphony rate (PR)	44.0%	0.488
Average pitch value (APV)	51.7%	0.565
Average note duration (AND)	52.3%	0.631
Deviation of pitch value (DPV)	51.1%	0.580
Deviation of note duration (DND)	53.2%	0.512
Pitch range (PRG)	63.1%	0.642
Shortest rhythm value (SRV)	66.2%	0.667
Tempo	52.9%	0.558

一方で，特徴量に基づいた分類は，生成モデルによるものと比較して精度が落ちることがわかる．これは，もともとピアノに対する学習で用いられていた手法であるため，ギターに適していないことや，連続する 2 音間の関係を取れる特徴量が少なかったことに起因すると考えられる．

表 5 に各特徴量それぞれ一つのみを用いた分類器での結果を示す．

一般に難易度というと，テンポや音符の細かさや音程のばらつきなどでほとんどわかるのではないかと思われがちであるが，実際に Accuracy を見ると，テンポのみを用いた分類器は 52.9%，最も短い音符の音価を用いた分類器は 66.2%，音程のばらつきを意味する Pitch entropy を用いた分類器は 60.0%と，他の特徴量と比べれば有効であるが，提案手法である Chord-shape に基づく Note n-gram よりは劣ることがわかる．以上の結果から，ギター楽譜からの難易度の推定において，Note n-gram が有効であることが示された．

つぎに，最も結果の良かった Chord-shape モデルを用いた NTG で分類に失敗したフレーズから，本手法の限界について考える．難フレーズと誤分類された易フレーズには，スウィー

プ等の特別な奏法を習得していれば容易に弾くことのできるものや，フレーズ自体は難しくないものの，早いテンポで高い音を弾いているものなどがあった．前者については，その奏法が音符列から読み取れるものであれば，それを用いるフレーズを複数データセットに用意することができれば，正しく検出することが可能になると考える．また，後者については，$P(l)$ と $P(\boldsymbol{b})$ が強く働いたことで $P(d)$ が強く出なかったためであると考えられる．そこで，各確率に対してパラメータを用意し，それぞれに重み付けをすることでこの問題は解決するのではないかと考える．

また，易フレーズと誤分類された難フレーズには，トリルやビブラートと言った奏法が入っているものや，フレーズ自体は難しいものの，低い音でゆっくりと弾いているものなどがあった．前者については，本研究における音符列の定義では取ることのできない情報による難易度であるため，これを取ることができるようなモデル作成が必要となる．後者については，易フレーズにおける問題と同様，各確率に対するパラメータが効果的であると考える．

5. 結　　論

本研究では，楽譜の生成モデルに基づいて，ギター楽譜の演奏難易度を推定する手法を提案した．楽譜の生成モデルには，音符の音程列と長さの生成が独立であることを仮定した Note n-gram と呼ばれるモデルを用い，和音の生成には，ギター特有の生成過程を仮定した Chord-shape モデルと呼ばれるモデルを使用した．実験では，提案手法である Note n-gram を用いたモデルに加え，一般的な n-gram を用いたモデルや楽譜の特徴量に基づく教師有り学習手法と比較を行い，提案手法の有効性を示した．また，今後の課題としては，特殊な奏法や一部の情報による誤推定があったため，これらへの対応が挙げられた．

謝辞　本研究は JSPS 科研費 JP26700009 の助成を受けたものです．ここに記して謝意を表します．

文　　献

[1] S.-C. Chiu and M.-S. Chen. A study on difficulty level recognition of piano sheet music. In *2012 IEEE International Symposium on Multimedia (ISM)*, pages 17–23. IEEE, 2012.

[2] K. Collins-Thompson and J. Callan. Predicting reading difficulty with statistical language models. *Journal of the American Society for Information Science and Technology*, 56(13):1448–1462, 2005.

[3] K. Collins-Thompson and J. P. Callan. A language modeling approach to predicting reading difficulty. In *HLT-NAACL*, pages 193–200, 2004.

[4] E. Holder, E. Tilevich, and A. Gillick. Musiplectics: computational assessment of the complexity of music scores. In *2015 ACM International Symposium on New Ideas, New Paradigms, and Reflections on Programming and Software (Onward!)*, pages 107–120. ACM, 2015.

[5] S. E. Petersen and M. Ostendorf. A machine learning approach to reading level assessment. *Computer speech & language*, 23(1):89–106, 2009.

[6] V. Sébastien, D. Sébastien, and N. Conruyt. Dynamic music lessons on a collaborative score annotation platform. In *The Sixth International Conference on Internet and Web Applications and Services, ICIW*, pages 178–183, 2011.

[7] L.-C. Yang, S.-Y. Chou, and Y.-H. Yang. Midinet: A convolutional generative adversarial network for symbolic-domain music generation. In *Proceedings of the 18th International Society for Music Information Retrieval Conference (ISMIR '2017), Suzhou, China*, 2017.

[8] 宮川洋平. ピアノ用楽譜の難易度評価手法の研究. 2003.

[9] 松原正樹, 遠山紀子, and 斎藤博昭. ピアノ初級者のための独習支援システムの提案. **情報処理学会研究報告音楽情報科学** *(MUS)*, 2006(19 (2006-MUS-064)):79–84, 2006.

[10] 森田花野, 小泉悠馬, and 伊藤克亘. 教則本を利用したギターフレーズの難易度推定. **第 *75* 回全国大会講演論文集**, 2013(1):267–268, 2013.

[11] 川村修, 大圃忠親, 伊藤孝行, and 新谷虎松. 逐次的リズム音程生成モデルに基づく自動作曲システム. **情報処理学会研究報告音楽情報科学** *(MUS)*, 2005(129 (2005-MUS-063)):19–24, 2005.

[12] 田村理遊, 但馬康宏, and 小谷善行. 音高と音価の隠れマルコフモデルを用いた自動副旋律生成. **情報処理学会研究報告音楽情報科学** *(MUS)*, 2007(15 (2007-MUS-069)):7–12, 2007.

[13] 藤田顕次, 大野博之, and 稲積宏誠. 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(89 (2008-MUS-077)):47–52, 2008.

[14] 白井亨 and 谷口忠大. 階層 pitman-yor 言語モデルを用いたメロディー生成手法の提案. **研究報告音楽情報科学** *(MUS)*, 2011(3):1–6, 2011.

[15] 米林裕一郎, 亀岡弘和, and 嵯峨山茂樹. 隠れマルコフモデルに基づくピアノ運指の自動決定. **情報処理学会研究報告音楽情報科学** *(MUS)*, 2006(45 (2006-MUS-065)):7–12, 2006.

[16] 矢澤一樹, 糸山克寿, and 奥乃博. 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. **研究報告音楽情報科学** *(MUS)*, 2013(17):1–6, 2013.