

PCFG の文法拡大による音列パターン解析

丹治 信 安藤 大地 伊庭 齊志

東京大学大学院工学系研究科
新領域創成科学研究科

本稿では、確率文脈自由文法 (PCFG) をベースに拍節構造モデルを構築し、その文法を拡大する手法を提案する。PCFG を音楽モデルとして仮定することで、確率的に最尤な構造を数理的に考える手法であるが、最初に与える文法はシンプルな文法なため学習データ中のパターンを捉えることができない。本研究では、PCFG 文法の拡大とその後の EM アルゴリズムによるパラメータ推定によって、学習データ中のパターンに特化した非終端記号とルールが学習されることを示す。また、拡大する非終端記号の選択基準の違いによる性能の変化について実験を行う。

Musical Pattern Analysis by Expansion of PCFG Grammar

Makoto Tanji Daichi Ando Hitoshi Iba

Graduate School of Engineering and Graduate School of Frontier Sciences, The University of Tokyo.

We propose a metrical grammar to analyze metrical structure of music and its improving method. At first, the model is given as simple PCFG grammar that represent metrical structure by derivation just like parse tree in natural language processing. The expansion operator duplicates a symbol and its rule in the grammar. Then EM algorithm is used to estimate parameters. We present experiments that show our grammar expansion method specialize rules and symbols to rhythmic pattern in training music. And it increases accuracy of prediction for new piece compared with original grammar.

1 Introduction

音楽が単なる音の列のみから成るのではなく、その中に様々な楽曲構造と呼ばれる構造を含むことは広く認められている。人間の聴者は、音楽を聴く際に意識的・無意識的にこれらの高次の情報を知覚する。和声進行やグルーピング構造、拍節構造などの楽曲構造を計算機により解析することは、音楽情報処理の分野での応用のみならず、人間の音楽に対する知覚に関する知見を得るという意味でも重要である。

本稿では、拍節構造を対象に、確率モデルによる楽曲解析手法について述べる。拍節構造は GTTM [6] の理論などで指摘されているように、階層構造を形成すると言われる。人間の聴者は事前の経験から、この拍節構造に対する何らかのモデルを持っており、そのため経験を積みば楽譜が与えられなくても、音列から楽曲構造を知覚することができると考えられる。本研究の目的は、適切な計算機モデルを構築し、拍節構造解析手法を提案することである。これには前提として、音楽的な経験を持つ人間の頭の中では、音楽に対する一般的な事前知識が学習されており、それによって楽曲構造を知覚し、ある程度の音楽の予測や裏切りが成立

するという考え方がある。筆者らは PCFG(確率文脈自由文法) をベースとした確率モデルで楽曲の拍節構造を解析手法を研究しており [10]、本稿ではより大局的な音列のパターンをモデルで表現する為に、文法の拡大によるモデルの生成規則を含めた推定手法について述べる。

本稿では、まず、対象とする拍節構造について第 2 章で簡単に説明する。次に、第 3 章で PCFG を使った、我々の手法について述べ、第 4 章でその文法を学習データに適合するように拡大する手法について述べる。5 章で行なった実験について報告し、最後に、まとめとする。

2 拍節構造

人間の聴者は音楽を単なる個々の音の並びとしてではなく、まとまりや同時に鳴っている音の関係性や周期性といった高次の情報を知覚する。特に拍節構造はメロディの解析に対して、重要な位置を占める。拍節構造は階層的な表現で表され、特に並列性や繰り返しのなどの大きな構造も存在する [6] [8]。

本研究の目的は、拍節構造の解析のための適切な計算機モデルを構築することにある。図 1 に見られるよ

うに、一つの音列に対していくつかの可能な拍節構造が存在する。複数の拍子の種類が存在し、また音列の開始位置によって様々な解釈が存在する。人間の聴者は容易にこれらの複数の可能性から一つのありそうな(作者や演奏者によって意図された)拍節構造を推定することができる。

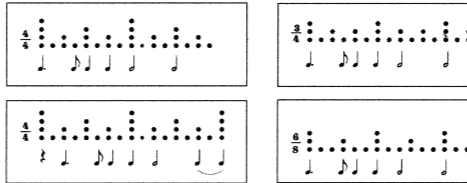


図1 Metrical Structures for the same note sequence

拍節構造の解析は、音楽情報処理の前処理や基盤的な技術として使われる。和声解析や演奏音列からの意図した楽譜の推定問題などに応用が考えられる。

3 拍節 PCFG モデル

初めに、我々の仮定する音楽モデルを図2に示す。楽曲は音楽モデル G から条件付き確率 $P(\mathbf{T}|\mathbf{G})$ で生成される。この時点で楽曲構造と音符列は保持されているが、我々は観測できない。これは作者の頭の中にある出来上がった楽曲などに相当する。我々が観測できるのは、楽譜のような“音符列と少しの楽曲構造”という形か、演奏による“音符列”という形である。

以上のモデルから、 $P(\mathbf{T}|\mathbf{G})$, $P(\mathbf{T}|\mathbf{S})$ など表現する適切な音楽モデルを仮定することで、効率的な計算機による逆問題としての解析が可能になると考えられる。

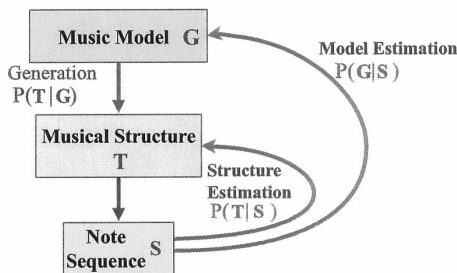


図2 Overview of the method

我々は、音楽モデルとして PCFG を仮定し、拍節 PCFG モデルを提案し [10]、モデルのパラメータ推定と拍節構造の推定手法を提案した。このモデルでは楽曲は拍節 PCFG モデルから確率 $P_G(\mathbf{T})$ で生成され、音符列 $\mathbf{S}$ は $P(\mathbf{S}|\mathbf{T}) = 1$ で観測される。よって、拍節

構造推定問題は以下のように、音符列 $\mathbf{S}$ を観測したときに条件付き確率を最大化する $\mathbf{T}$ を求める問題として定式化される。

$$\hat{\mathbf{T}} = \underset{\mathbf{T}}{\operatorname{argmax}} P(\mathbf{T}|\mathbf{S}) = \underset{\mathbf{T}}{\operatorname{argmax}} \frac{P(\mathbf{S}, \mathbf{T})}{P(\mathbf{S})}$$

$$\hat{\mathbf{T}} = \underset{\mathbf{T}}{\operatorname{argmax}} P(\mathbf{S}, \mathbf{T})$$

3.1 PCFG(Probabilistic Context-Free Grammar)

PCFG (Probabilistic Context-Free Grammar) は CFG (Context-Free Grammar) の生成規則に確率を付加することで確率モデルに発展させたモデルである。PCFG G の形式的表現は以下で表される。ここで本来、生成規則の右側の個数に制限は無いが、ここでは等価な変換のできるチョムスキー標準型で表記している。

- $G = \{V_T, V_N, P, S\}$
- V_N : a finite set of nonterminal symbols
- V_T : a finite set of terminal symbols
- P : a finite set of production rules $\langle A \rightarrow \alpha, p \rangle$
($A \in V_N, \alpha \in (V_N \cup V_T)^*, p$: probability)
- S : start symbol ($S \in V_N$)

開始記号 S から始め、生成規則を繰り返し適用して終端記号の列を得る過程を導出と呼び、得られた終端記号列 *string* を文字列と呼ぶ。本手法で我々は、この文字列を音符列、導出を拍節構造と見なす。導出は導出木表現で表され、その確率は以下で与えられる。

$$P(\mathbf{S}, \mathbf{T}) = p(t_1) \cdots p(t_N) = \prod_{i=1}^N p(t_i)$$

ここで、 t_i は i 番目に使用された生成規則である。文法 G から、 $\mathbf{S}$ を生成する導出が複数個ある場合、その文法は曖昧であると言われる。PCFG では、導出に確率が計算でき、確率の最も高いものを考えることで、最尤導出を尤もらしい導出として区別することができる。

3.2 Metrical PCFG Model

表1に示すような文法を持つ PCFG を構築し、拍節 PCFG モデルと呼ぶ。生成規則の確率パラメータ数は148個である。この文法は、拍節構造を構築しながら単旋律メロディを生成する。また、アウフタクトや最後の小節が不完全な長さの音符列も*1、*2マークの生成規則により許容する。また、拍節構造は、言語処理の構文木のように導出木そのものに対応する。

音符列の生成は次の手順で行われる。

1. 開始記号 ST から “ $Nx \text{ Beat}x$ ” を生成する。 x は例えば $3/4$ などの拍子記号に対応する変数である。
2. ルール “ $\text{Beat}x \rightarrow Nx \text{ Beat}x$ ” を繰り返し適用し、“ Nx

表 1 Grammar of Metrical PCFG

Nonterminal Symbols	Terminal Symbols
ST, Beat x , N y ($x \in A, y \in B$)	note: $x[p]$, rest: x ($x \in B, p \in P$)
Production Rules	
ST $\rightarrow$ N x Beat x ($x \in A$)	
ST $\rightarrow$ N x Beat y ($x \in B, y \in A, x < y$)	*1
ST $\rightarrow$ N x N x ($x \in A$)	
Beat x $\rightarrow$ N x Beat x ($x \in A$)	
Beat x $\rightarrow$ N x N x ($x \in A$)	
Beat x $\rightarrow$ N x N y ($x \in A, y \in B, y < x$)	*2
N x $\rightarrow$ N s N t ($x, s, t \in B, s + t = x$)	
N x $\rightarrow$ note: $x[p]$ ($x \in B, p \in P$)	
N x $\rightarrow$ rest: x ($x \in B$)	
A = {1/1, 3/4}, P = {1...7}	
B = {1/1, 1/2, 1/4, 1/8, 1/16, 1/32, 3/4, 3/8, 3/16, 3/32}	

...N x ”を連続して生成する。もしくは“Beat x $\rightarrow$ N x N x ”ルールで小節の生成をストップする。

3. “N x ”をより細かい音価の組み合わせ“N s ”“N t ”に再帰的に分割する。もしくは終端記号“note: $x[p]$ ”へと変化させる。(x と p はそれぞれ長さと同程を表す)
4. もし全ての記号が終端記号“note: $x[p]$ ”へ変化したら終了する。

例として、図 3 に導出木の例を示す。この場合 N1/1 は小節の拍子を表す。

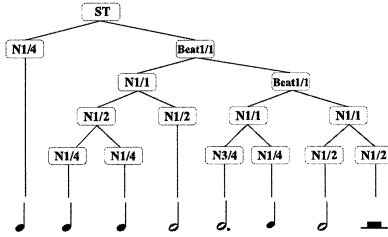


図 3 Sample tree of Metrical PCFG

拍節 PCFG 文法からは、一つの音符列に対して複数の導出木が存在するため、曖昧な文法である。例えば、図 4 に 2 つの導出木を示す。この音列を聞いたとき、人間の聴者は 2 番目の拍を知覚すると考えられる。この違いを確率的に解釈すると次のようになる。生成規則 “N1/2 $\rightarrow$ N1/4 N1/4” は一般的なリズムをつくる規則であり、一方、“N1/2 $\rightarrow$ N1/8 N3/8” はあり得なくは無いが、あまり一般的なリズムの分割では無い。このような生成規則の確率の違いから、我々は 2 番目の拍節構造が確率が高く、尤もらしいと結論付けることが

できる。

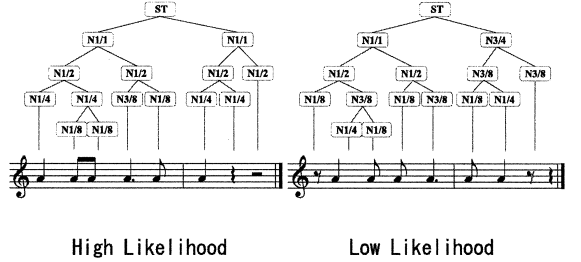


図 4 High and Low likelihood Structures

3.3 最尤導出とパラメータ推定手法

PCFG において、最尤導出 $\arg\max_T P(S, T)$ は、CKY アルゴリズムを使って、Viterbi アルゴリズムと同じように求めることが出来る [9]。また、*Inside-Outside* アルゴリズムと呼ばれる効率的なパラメータ推定手法が提案されている。Inside-Outside アルゴリズムは EM アルゴリズムの PCFG 版と捉えることができ、学習データの尤度を山登り法で増加させる [5][3]。Inside-Outside アルゴリズムに従うと、学習データ W の導出中に出現した記号 A の回数 $count(A)$ は $count(A) = \sum_{RHS} count(A \rightarrow RHS; W)$ で推定される。ここで $count(A \rightarrow RHS)$ は推定された生成規則 “ $A \rightarrow RHS$ ” の回数である。EM アルゴリズムのパラメータ更新式は以下で表される。

$$\hat{P}(A \rightarrow \alpha) = \frac{count(A \rightarrow \alpha)}{\sum_{\beta} count(A \rightarrow \beta)}$$

また、括弧付きのデータから、その括弧を導出の一部として効率的に Inside-Outside アルゴリズムを高速に実行する改良アルゴリズムが提案されており [7]、括弧情報が導出の曖昧性を減少させるため、高速化と共に精度が向上する。

本稿では上記のアルゴリズムを使って学習、最尤導出の計算を行う。

4 文法の拡大手法

前節の拍節 PCFG モデルは、小節の連続的な生成と音価の分割ルールのみを用いると言う意味で、非常にシンプルなモデルである。本節では PCFG 文法を拡大し、拡大した記号及び生成規則によって学習データ中の音列パターンに特化した文法を推定する手法について述べる。CFG の帰納学習は昔から多くの研究がなされており、確率を付加した PCFG の帰納学習や文法の探索もいくつか研究されている [4][1]。

本稿では、文法の拡大を、非終端記号を 2 つにコピー

し、関連する生成規則も同様にコピーする操作とする。その後、2つの記号の対称性を崩すために、ランダムなノイズを生成規則の確率に加え、EM アルゴリズムによりパラメータを推定する。以上を繰り返すことで、生成規則の確率が学習データの一部に特化した形に分化し、特定の記号が学習データのパターンを表すことが期待される。なお、文法の記号・生成規則の数が増え、パラメータ数が多くなると、過学習してしまうという危険性があるため、本手法では、学習データ・検証データ・テストデータの3つのデータを使い、学習データの下で推定したパラメータを検証データでテストし、拡大された文法が正しく検証データを予測できればその拡大を採択するという手法をとる。以下に、文法拡大の擬似コードを示す。

Algorithm 4.1: GRAMMAR EXPANSION(G, C)

```

 $G := \text{grammar}$ 
 $\theta := \text{parameters}$ 
 $T := \text{TrainingData}$ 
 $V := \text{ValidationData}$ 

# 初期化
 $G \leftarrow \text{basic grammar}$ 
 $\theta \leftarrow \text{initial value}$ 
for  $i \leftarrow 1$  to  $M$  # 初期パラメータ推定
  do  $\theta \leftarrow \text{EMStep}(G, \theta, T)$ 
for  $i \leftarrow 1$  to  $N$  # 文法の拡大を最大  $N$  回繰り返す
  {
     $X \leftarrow \text{selectSymbol}(G, \theta)$  # 文法の拡大
     $(G', \theta) \leftarrow \text{expand}(G, \theta, X)$ 
    for  $i \leftarrow 1$  to  $M$  # パラメータ推定 (EM アルゴリズム)
      do  $\theta \leftarrow \text{EMStep}(G, \theta, T)$ 
    # 改悪になっていなければ文法を更新交換
    if  $\text{score}(G, V) < \text{score}_i(G', V)$ 
      then  $G \leftarrow G'$ 
  }
return  $(G, \theta)$ 

```

4.1 Expansion Operator

文法拡大処理を $\text{Expand}(G, \theta, X)$ と表記する。これは文法 G 、パラメータ θ の下で、適当な非終端記号 X を取り、新しい非終端記号 X' を作り、また、 X に関連するルールをコピーする操作になる。以下に X を拡大する操作の例を示す。

$$\left[\begin{array}{lll} A \rightarrow B X & A \rightarrow B X & A \rightarrow X X \\ A \rightarrow X X & \Rightarrow A \rightarrow B X' & A \rightarrow X X' \\ X \rightarrow A B & X \rightarrow A B & A \rightarrow X' X \\ & X' \rightarrow A B & A \rightarrow X' X' \end{array} \right]$$

また、コピーされたルールの確率パラメータは以下のようを与える。

$$p_{\text{new}}(A \rightarrow B X) = p_{\text{new}}(A \rightarrow B X') = \frac{p_{\text{old}}(A \rightarrow B X)}{C(A \rightarrow B X)}$$

$$p_{\text{new}}(X' \rightarrow A B) = p_{\text{old}}(X \rightarrow A B) + \epsilon$$

ここで、 $C(A \rightarrow B X)$ は、複製された生成規則の数を表す。また、 ϵ はコピーされた非終端記号の対称性を崩すためのノイズであり、本稿では 0.1 とした。

文法のどの非終端記号を選ぶかという選択基準には、いくつかの種類が存在する。本研究では以下にあげる選択基準を使用する。

ランダム選択 全ての非終端記号を同確率で選ぶ。この際、複製の正のフィードバックを防ぐために、複製された記号に関してはオリジナルの記号と合わせて一つと考え、選ばれた記号が複製されていた場合、その中からまたランダムで選択する。

出現率比例選択 学習データ中に現れたと推定された回数に比例した確率で選択する。 $P_{\text{select}}(X) = \sum_{\alpha} \text{count}(X \rightarrow \alpha)$

出現率最大選択 学習データ中に現れたと推定された回数が最大の非終端記号を選択する。

5 実験

5.1 単一曲に対する適用

音楽データに対して文法の拡大がどの程度有効に働くかを検証するため、計算機実験を行った。初めに、単一の曲に対して PCFG のパラメータを学習させ、文法の拡大を行うことで、楽曲のパターンを適切に抽出できるかを確認する。

実験に使用した楽曲は、J.S.Bach の Menuet BWV1009 からの抜粋である。この曲は明確なパターンの繰り返しで成り立っているため、本手法が有効に働くと考えられる。

異なった非終端記号の選び方を用いて、文法の拡大による尤度の上昇を比較した。非終端記号の選び方は、[拡大無し], [N1/4], [N1/4, N1/2], [N1/4, N1/2, N3/4], [N1/4, N1/2, N3/4, Beat3/4], [N1/4, N1/2, N3/4, N3/4, Beat3/4], [N1/4, N1/2, N3/4, N3/4, Beat3/4, Beat3/4] の7種類である。まず、シンプルな文法の下での EM アルゴリズムを 10 回繰り返し、初期パラメータを学習し、その後上の文法の拡大を行い、また EM アルゴリズムでパラメータを学習させた。図 5 に、結果の対数尤度の上昇カーブを示す。図より、拡大する非終端記号が多いほど対数尤度が上昇していることが判る。また、対数尤度が最大になったのは拡大した非終端記号

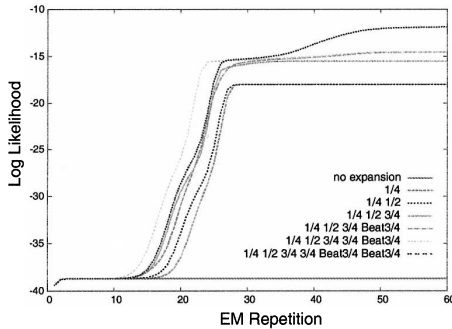


図5 The log likelihood graph in 6 types of expansions

が [N1/4, N1/2, N3/4, N3/4, Beat3/4, Beat3/4] のときであった。図6に、対数尤度が最大になった文法から得られた最尤導出の一部を示す。A,B,Cそれぞれ、同一のリズムパターンに対応しているのが見れる。このように複製された記号 N3/4, N3/4*, N3/4**がそれぞれのパターンに特化した生成規則を持つことで、尤度が上昇することが確認できた。

以上のことから、適切な非終端記号を選ぶことで、学習データに対して適合した PCFG の文法が得られることが期待できる。

5.2 楽曲コーパスに対する検証実験

文法拡大によるモデルの精度の評価尺度の一つは、未知の楽曲に対する予測精度がある。定量的にこれらを測定するために、楽曲コーパスを用いて本手法を適用し、非終端記号の選択基準を比較した。

5.2.1 Corpus

学習データには, Essen Folksong Collection [2] を使用した。主にヨーロッパの民謡を集めたもので、単旋律であり、小節とフレーズの情報メタデータが付加されている。我々が構築した拍節 PCFG モデルでは、 $\frac{4}{4}$ 、及び $\frac{3}{4}$ の長さの小節のみを扱うため、拍子が $\frac{3}{2}$, $\frac{4}{2}$ の楽曲などは除いた。

全部で約 3400 曲の学習データから、その中で学習データ 50%・検証データ 1%・テストデータ 10% と分割して実験に使用した。

5.2.2 Grammar Improving.

拍節 PCFG モデルの文法は、前節の文法の拡大手法に従って拡大された。拡大の最大の回数は $N=20$ とし、各拡大に対する EM アルゴリズムの繰り返し回数は 10 回とした。また、検証データでのパフォーマンスが向上した時のみ文法を更新し、テストデータでのパフォーマンスを測定した。パフォーマンスの測定には以下で与えられる F-Score を用い、テストデータの

節位置の予測精度を測定した。

$$\text{Precision} = \frac{\# \text{correctly predicted boundaries}}{\# \text{predicted boundaries}}$$

$$\text{Recall} = \frac{\# \text{correctly predicted boundaries}}{\# \text{original boundaries}}$$

$$\text{F-score} = \frac{2 * \text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}}$$

5.2.3 結果

文法拡大による、未知楽曲に対する予測精度の変化を図7に示す。グラフの各線は、各選択基準で文法の拡大を行った際のテストデータに対する F-Score を表す。ほぼ拡大の回数の対して F-Score が上がる傾向が見られた。なお、本手法では、検証データに対する F-Score が向上した時のみテストデータを評価する方法を取っているため、まばらにしかデータ点は得られておらず、直線的なデータとなっている。

もし、学習データ中にしか現れないパターンが学習されていたとすると、テストデータには影響しないと考えられるため、この F-Score が向上したという結果からは、学習データ中になんらかの一般的な音列パターンが存在しており、文法の拡大によってそれらが非終端記号で表現されていることを示唆される。F-Score の上昇は出現率比例選択で約 8~9% 程度であり、ランダム選択、出現率最大選択では約 5~6% 程度であった。

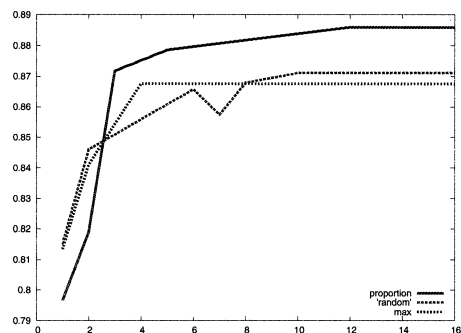


図7 テストデータの対する F-Score の遷移

6 Conclusion

本稿で、我々は PCFG をベースにした音楽の拍節構造モデルとその文法を拡大する手法を提案した。文法の拡大によって、複製した非終端記号を学習データのパターンに特化させることで尤度が向上し、単一の楽曲に適用した実験から音列パターンに複製された非終端記号が対応することが示された。文法の拡大と EM アルゴリズムによって学習データの尤度はほぼ単調に

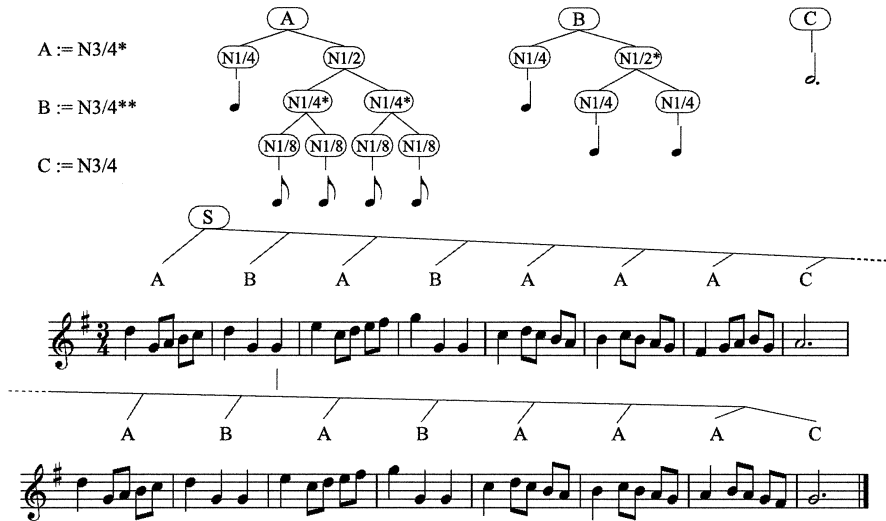


図 6 最尤導出で割り当てられた小節毎の記号

増加し、また未知楽曲の予測精度も向上することが確認できた。本研究では、リズムの分割ルールと小節の生成ルールのみで音楽的な事前知識をほとんど使わずに、音楽知識を表現するモデルの構築を目指しており、文法の拡大手法は学習データに自動的に適合する点で望ましい特性と備えている。

今後の課題として、文法の Merge やスムージングを組み合わせた、逆方向の文法の探索手法を試していきたい。

参考文献

- [1] Joseph Bockhorst and Mark Craven. Refining the structure of a stochastic context-free grammar. In *IJCAI*, pages 1315–1322, 2001.
- [2] H.Schaffrath. *The Essen Folksong Collection in the Humdrum Kern Format*. Menlo Park, CA: Center for Computer Assisted Research in the Humanities, 1995.
- [3] Kenji Kita. *Probabilistic Language Model (Japanese)*. University of Tokyo Press, 1999.
- [4] Kenichi Kurihara, Yoshitaka Kameya, and Taisuke Sato. Efficient grammar induction algorithm with parse forests from real corpora. *Transactions of the Japanese Society for Artificial Intelligence*, Vol. 19:360–367, 2004.
- [5] K. Lari and S.J. Young. The estimation of stochastic context-free grammars using the inside-outside algorithm. *Computer speech & language*, 4:237–

257, 1990.

- [6] Fred Lerdahl and Ray Jackendoff. *A Generative Theory of Tonal Music*. The MIT Press, 1983.
- [7] F. Pereira and Y. Schebes. Inside-outside reestimation from partially bracketed corpora. In *the 30th Annual Meeting of the Association for Computational Linguistics*, pages 128–135, 1992.
- [8] David Temperley. An evaluation system for metrical models. *Computer Music Journal*, 28:3:28–44, 2004.
- [9] 北 研二. 確率的言語モデル. 言語と計算 4. 東京大学出版会, 1999.
- [10] 丹治 信, 安藤 大地, and 伊庭 斉志. 確率文脈自由文法による旋律の拍節モデル推定. In 情報処理学会研究報告 *MUS-73*, 2007.