

σ GTTM III の構築

Developing σGTTM III

浜中 雅俊^{*1}

Masatoshi Hamanaka

^{*1} 京都大学

Kyoto University

We propose an automatic analyzer for acquiring a time-span tree based on the generative theory of tonal music (GTTM). Although analyzer based on GTTM was previously proposed, it requires manually tweaking the 46 adjustable parameters on a computer screen in order to analyze them properly. We re-formalized the time-span reduction in GTTM based on a statistical model called probabilistic context-free grammar, which enables us to acquire the most probabilistic time-span tree. We applied leave-one-out cross validation using three hundred sets training data, which revealed that our analyzer outperformed the previous one.

1. はじめに

本稿では、音楽理論 GTTM (A Generative Theory of Tonal Music) [Lerdahl 1983] に基づき、楽曲およびその楽曲のグルーピング構造、拍節構造からタイムスパン木を獲得する手法について述べる。我々の手法の特長は、音楽家による楽曲分析結果を確率文脈自由文法 (Probabilistic context-free grammar, PCFG) [Charniak 1996] に基づき統計的に学習することでタイムスパン木を自動生成するためのモデルを獲得する点である。

一般に、楽曲の解釈は一意ではなく複数の解釈が有り得る。そして、この多義性・曖昧性が楽曲解釈を計算機上で実現することを難しくしていた。そこで本研究では、楽曲分析器に確率文脈自由文法を導入する。確率文脈自由文法は、確率が付与された複数の生成規則を持ち、ある文を生成する確率は、その導出で使われた生成規則群の確率の積で表れる。そして、ある文を生成する生成規則群を見つけることを「導出」、導出結果を示す木構造を「導出木」と呼ぶ。本研究では、楽曲分析器に確率文脈自由文法を導入することで、生成確率の異なる複数の解釈が導出されるシステムを構築することを可能とする。

音楽理論 GTTM に基づく分析で獲得されるタイムスパン木は、木の先端で時系列順に並んだ音符に接続され、先端から根に至る途中の分岐では構造的に重要な音符を幹、装飾的な音符が葉となるように構成された二分木である。したがって、構造的に重要な音符を残し、装飾的な音符を簡約することで、簡約したメロディを抽出することができる。

本研究では、タイムスパン木を確率文脈自由文法におけるメロディの導出木とみなす。すると、タイムスパン木を用いて簡約されたメロディを求める問題を順問題、メロディを生成する問題をその順問題と捉えることができる。そこで我々はまず、762 個の生成規則を作成し、音楽家が手作業で分析した 300 曲の分析データから各生成規則の生成確率を求める。次に、タイムスパン木を求めたい楽曲およびその楽曲のグルーピング構造、拍節構造から、可能なすべてのタイムスパン木を生成し、その導出確率を求める。そして、導出確率が最大となるタイムスパン木を適切なタイムスパン木として選択することで逆問題を解く。

実験の結果、精度は平均 0.76 となり従来の GTTM 分析器を上回る性能であることを確認した。

2. 関連研究

これまでタイムスパン木を用いたメロディの要約手法[Hirata 2003], モーフィング手法[Hamanaka 2008], 編曲手法[Hirata 2004]が提案されてきた。しかしそこで用いられたタイムスパン木は、音楽家が手作業で分析したものであった。

我々は、これまで 4 つの GTTM 分析器を構築してきた。ATTA (Automatic time-span tree analyzer) は、音楽理論 GTTM を外在化・パラメータ化することにより計算機実装用に拡張した exGTTM (machine executable extension GTTM) を計算機上に実装したシステムである[Hamanaka 2006]。外在化には、大局的な構造に関するルールと局所的な構造に関するルールの両方を適切に組み合わせ、階層的な構造を獲得するアルゴリズムの導入も含まれる。パラメータ化は、ルールの優先順序を決め、ルール間の競合を防ぐだけでなく、階層的なタイムスパン木の構造を変化させることを可能としていた。しかし、ATTA では画面に表示される分析結果を見ながら手作業でパラメータを調節する困難な作業を必要としていた。ATTA のタイムスパン簡約における問題点については、3.3 で再度議論する。

一方、FATTA (full-automatic time-span tree analyzer) は、タイムスパン木からグルーピング構造、拍節構造へのフィードバックを導入し、ATTA のパラメータの値を推定することを可能としていた[Hamanaka 2007]。しかし、FATTA の出力する構造は誤りが多く、特にタイムスパン木では、ATTA に比べて大きく性能が低下していた。

σGTTM[Miura 2009]と σGTTM[Kanamori 2014]は、音楽家の分析結果を用いて決定木に基づきルールの優先度を学習することで、局所的グルーピング境界検出することを可能としていた。σGTTM およびσGTTMII は、局所的グルーピング境界の検出について ATTA や FATTA に比べて高い性能を示していたが、拍節構造やタイムスパン木を獲得することはできなかった。

音楽分析に確率文脈自由文法を利用した研究事例としては、ジャズのコード進行の分析[Granroth 2012], 拍節構造の分析[Tanji 2008], 自動採譜[Kameoka 2012]などがある。これらの研究は、音楽分析における確率文脈自由文法の有用性を示した点が優れていた。GTTM でも、タイトルに表れているように Generation(生成)を強く意識しているが、タイムスパン木を確率文脈自由文法の導出木と捉え、最尤なタイムスパン木を求めようとする試みはなかった。

3. タイムスパン簡約の実装上の問題

GTTM では、グルーピング構造分析、拍節構造分析、タイムスパン簡約の順で分析が進められる。すなわち、タイムスパン木を獲得するためには、グルーピング構造と拍節構造が必要である。グルーピング構造分析は、連続したメロディをフレーズやモチーフなどに階層的に分割するものである。拍節構造分析は、4分音符/2分音符/1小節/2小節/4小節など各拍節レベルにおける強拍と弱拍を同定するものである。タイムスパン木は、構造的に重要な音符を幹、装飾的な音符が葉となるように構成された二分木である（図1）。

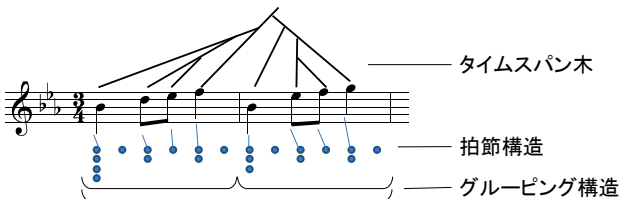


図1: グルーピング構造, 拍節構造, タイムスパン木

図2aの左図は、簡単なメロディとそのタイムスパン木を表したもので、そのタイムスパン(<---->で示した範囲)は、右図のようにヘッドと呼ばれる1音(ここではC4)で代表させることができる。このとき、枝の分岐において、幹となっているほうをプライマリ、葉となっているほうをセカンダリと呼ぶことにする(図2b)。

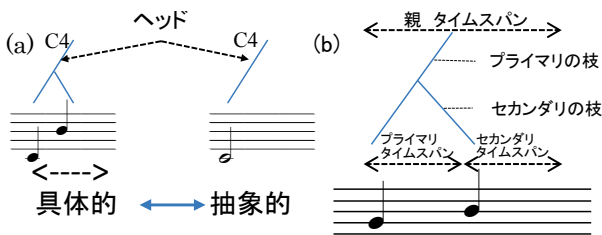


図2: ヘッド, プライマリ, セカンダリ

3.1 タイムスパン分割

タイムスパン簡約の前処理として、楽曲全体を階層的なタイムスパンに分割する(図3)。タイムスパン分割は以下のように行われる。

1. すべてのグルーピング構造をタイムスパンとする。
2. タイムスパンが二つ以上の音符を含む場合、タイムスパン内で最も拍点の多い(強い拍の)位置でタイムスパンを2つに分割する
3. 2を繰り返す

タイムスパンセグメンションでは、2つのルールが定義されており、それぞれが上記の1と2に対応している。

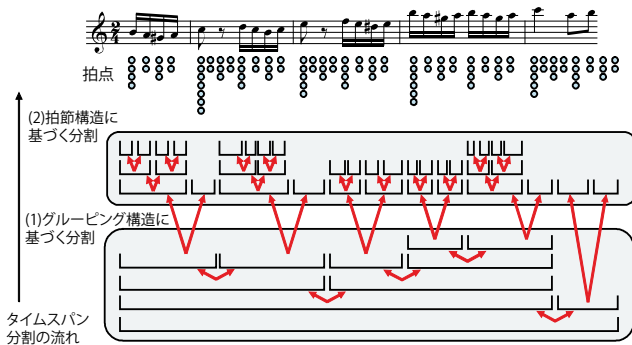


図3: タイムスパン分割

3.2 タイムスパン簡約の概要

タイムスパン簡約では、構造的に重要な枝が根に近い枝に接続されるような二分木を構成する。ヘッドの選択は、木の先端から根に向かって順に行われていく。その際、どのような木構造が望ましいかを示す9個のタイムスパン簡約選好ルールが定義されている。9個のルールの中には、局所的な部分に関するルールもあれば、大局的な構造に関するルールもある。

たとえば、TSRPR1とTSRPR5は両方とも拍節構造に関するルールである。TSRPR1 (Metrical Position)は局所的なルールで、ヘッドを選ぶ際、強い拍の音符であるほどヘッドになりやすいと定義されている。一方、TSRPR5 (Metrical Stability)は、大局的なルールで、ヘッドを選ぶ際、拍節構造が安定するように選ぶと定義されている。タイムスパン簡約を計算機上に実装する上での大きな問題の一つは、このような局所的なルールと大局的なルールをどのように組み合わせるのかについての説明が不十分なことである。

3.3 ATTAにおけるタイムスパン簡約の問題

3.2節で述べた問題を解決するため、我々はATTAにおいて各ルールの強弱をコントロールできるように調節可能なパラメータを導入した。そして、現在のレベルのヘッドの強さの計算と、次のレベルのヘッドの選択を繰り返すことにより階層的なタイムスパン木の獲得を可能とした(図4)。

評価実験の結果、ATTAのタイムスパン簡約では、極端に性能の低い曲が存在するために、平均の性能を大きく押し下げていることが分かった。それらの曲について調べたところ、多くは、木の枝の先端付近では弱拍がヘッドになり、根本付近では強拍がヘッドになっていたことが確認された(図5)。このように、木の先端付近と根本付近で重要なルールが異なる曲の場合では、ATTAを用いて正しく分析することは困難である。

このような曲についても正しく分析可能な分析器を実現するための方法としては、2つが考えられる。一つは、調節可能なパラメータをさらに追加し、ルールの強弱を木の先端と根本で独立に調整可能にすることである。しかし、ATTAには既に多くのパラメータが導入されており、その調整が難しくなっていることから、さらなるパラメータの追加は現実的ではない。二つ目の方法については、次の節で詳細に述べる。

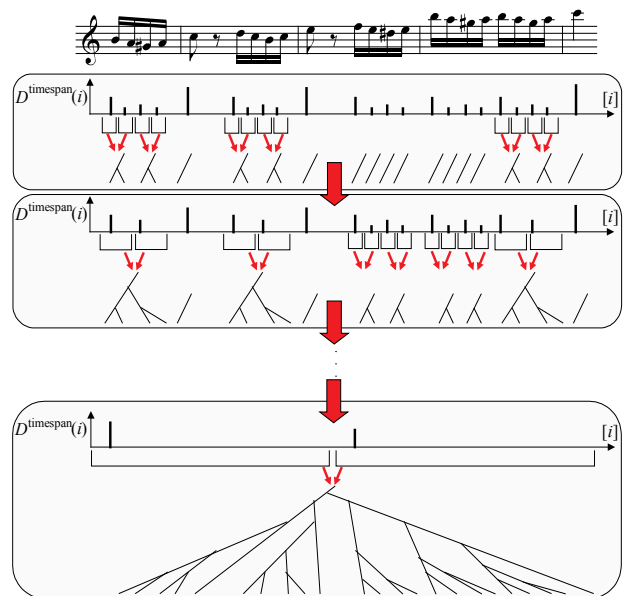


図4: ヘッドの強さの計算と次のレベルのヘッドの選択

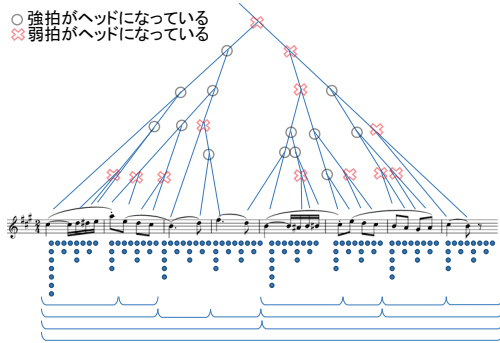


図 5：性能が低かった曲の例

4. σGTTMIII: 学習に基づく音楽分析器

タイムスパン木は、木の先端の装飾的な音符を簡約することで、より抽象化したメロディを抽出することができる。図 6a のメロディ A 上にある木構造は、メロディ A を分析して求めたタイムスパン木である。レベル B より下で分岐している枝に接続している音符を簡約するとメロディ B が抽出される。さらに、レベル C より下で分岐している枝に接続している音符を簡約するとメロディ C が抽出される。

この簡約プロセスを逆方向に眺めると、以下のような生成のプロセスと捉えることができる(図 7)。

1. 楽曲全体の長さのタイムスパンを持つ一つの音符を考える。
2. タイムスパンをプライマリとセカンダリに分割する。
3. 枝の末端になるまで 2 を繰り返す。

上記の生成プロセスを確率モデルで表現することで、最も尤らしいタイムスパン木の獲得が可能となる。



図 6：メロディの簡約と音符列の生成モデル

4.1 学習データ

確率モデルの学習には、我々が構築した GTTM データベースを用いる。GTTM データベースは、GTTM を良く理解している 3 人の音楽家がクラシック曲から切り出した 8 小節の長さの 300 個のメロディの楽譜データと、それを GTTM に基づき手作業で分析したグルーピング構造解析データ、拍節構造解析データ、タイムスパン解析データ、プロロンゲーション解析データ、および、和声を解析した和声解析データからなる。GTTM データベースおよび解析に使用した解析ツールは、以下のアドレスからダウンロードできる。

<http://www.gttm.jp/>

4.2 音符列の生成モデル

音楽の生成プロセスをモデル化するため確率文脈自由文法を導入する。確率文脈自由文法は、複数の生成規則とその生成確率からなり、各生成規則は、タイムスパン木におけるプライマリとセカンダリの分岐を表す。確率文脈自由文法 $G = \{T, M, S, R, P\}$ は 5 つの要素で定義される。

T: 終端記号の集合
音符が終端記号である。

M: 非終端記号の集合
タイムスパンは非終端記号である。

S: 開始記号
開始記号は、タイムスパン木の根にあたる。すなわち、休符を含まない楽曲全体の長さのタイムスパンである。

R: 生成規則の集合
生成規則には、タイムスパン分割ルールと、音符生成ルールの 2 種類がある。タイムスパン分割ルールは、タイムスパンを 2 つのタイムスパンに分割する。分割された 2 つのタイムスパンの長さの合計は、もとのタイムスパンの長さに等しい。タイムスパンの分割のされ方、すなわち、2 つのタイムスパンの音価の比は様々なものが考えられる。音符生成ルールは、タイムスパンから同じ長さの音符を生成する(図 7a)。

P: 生成規則の確率
各生成規則は確率が付与される。たとえば、32 分音符の音価のタイムスパンが 32 分音符を生成する確率は 1 に非常に近い値となる。なぜなら、32 分音符よりも音価の小さい音符は、データベース中でほとんど存在せず、32 分音符の長さのタイムスパンがさらに分割される確率は極めて低いためである。一方、たとえば倍全音符の音価のタイムスパンが、倍全音符の音符を生成する確率は極めて低い(図 7b)。

4.3 楽曲の生成モデル

4.2 では、タイムスパンを分割し音符列(音価列)を生成する方法について述べた。本節では、その他の属性について検討する。

プライマリとセカンダリ:タイムスパン分割ルールでタイムスパンが 2 つに分割される際、片方がプライマリ、もう片方がセカンダリとなる。

音高:プライマリのタイムスパンは、分割前のタイムスパンの音高を引き継ぎ、セカンダリの音高は新たに生成される。

タイムスパンの順序:プライマリのタイムスパンが前、セカンダリのタイムスパンが後となる場合と、プライマリが後、セカンダリが前となる場合の 2 通りが考えられる。

拍点の数:3.2 で述べたように、拍節構造はタイムスパン木の構成に強い影響を及ぼす。したがって、プライマリとセカンダリのそれぞれの拍点の数はモデルに含めることが望ましい。

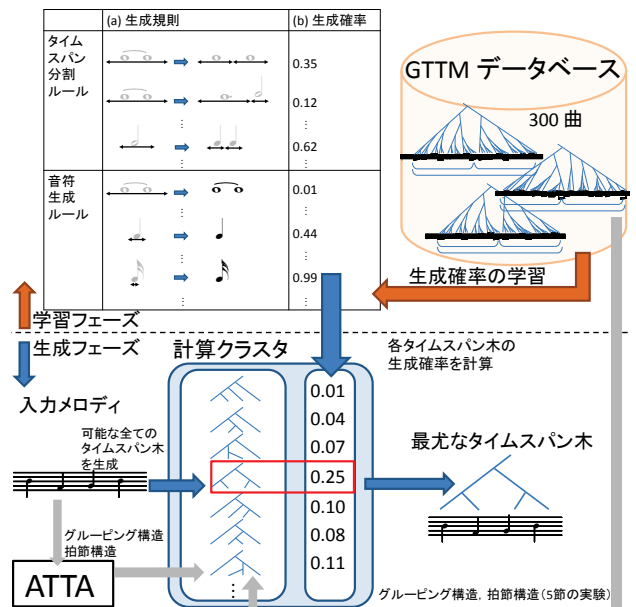


図 7：σGTTMIII の全体像

上記のすべての場合を区別して生成規則を作成すると、膨大な数の規則が作成され、限られた学習データから確率を求めると、多くの規則で確率が0になってしまう。そこで、本研究では学習データのスパース性の問題を解決するため、以下のように抽象化したルールを生成する。

- 3 種類の音高変化: 2つのタイムスパンの音高変化は、上がる、下がる、等しい、の3種類に分類する。
 - 7種類の音価比: 2つのタイムスパンの音価の比は、4倍、3倍、2倍、1倍、1/2倍、1/3倍、1/4倍の7種類のうち最も近いものに分類する。
 - 2種類の順序: 2つのタイムスパンの順序は、プライマリが前、セカンダリが後となる場合と、プライマリが後、セカンダリが前となる場合の2通りに分類する。
 - 3種類の拍点数: 2つのタイムスパンの拍点の数は、プライマリの拍点が多い、セカンダリの拍点が多い、プライマリとセカンダリの拍点の数が等しい、の3種類に分類する。
 - 6種類のタイムスパンの長さ: 分割前のタイムスパンの無さを、16分音符、8分音符、4分音符、2分音符、全音符、倍全音符の中から一番近いものに分類する。
- 上記より、756個(=3x7x2x3x6)のタイムスパン分割ルールと、6個の音符生成ルール、合計762個のルールを作成した。

4.4 タイムスパン木の生成

生成確率は、GTTM データベースにある300曲のタイムスパン木の19,296個の分岐から数え上げて求めた。ある楽曲の最尤なタイムスパン木は、可能なすべてのタイムスパン木を生成し生成確率が最大となったものを選択することで獲得した。タイムスパン分割に用いたグルーピング構造および拍節構造は、音楽家が分析したGTTM データベースのデータを用いた。

全てのタイムスパン木の生成およびその生成確率の計算には膨大な計算時間がかかるため、計算時間の短縮のため12並列で処理を行った。100曲の最尤なタイムスパン木を求めるのに計算クラスター(16台×Intel Xeon E5-2430@2.00 GHz 12core)を使って6週間の時間を要した。計算時間が最も短い曲では2分、長い曲では4週間であった(図7)。

5. 実験結果

σ GTTMIII の性能を leave-one-out cross validation で評価し、ATTA の結果と比較した。GTTM データベースにある300曲のうち、ATTA の評価でも用いた1番から100番までの曲を用いた。評価は以下の式で行う。

$$Accuracy = \frac{\text{正解データと同じ分岐となっている数}}{\text{タイムスパン木の分岐の数}}$$

表1は、結果を示したもので、ATTA の結果はパラメータ調整前の精度と、1曲につき10分程度でパラメータ調整をした後の精度の両方を示している。 σ GTTMIII は、すべての曲でパラメータ調整前の ATTA の性能を上回る性能を示していた。いくつかの曲では、ATTA のパラメータ調整後のほうが σ GTTMIII より性能が高い場合もあったが、平均では σ GTTMIII のほうが高い性能を示していた。

表1: σ GTTMIII と ATTA の比較

メロディ	ATTA 未調整	ATTA 調整済	σ GTTMIII
1. Moments Musicaux	0.71	0.84	0.88
2. Wiegenlied	0.54	0.69	0.78
3. Traumerei	0.50	0.63	0.84
4. An die Freude	0.22	0.48	0.68
5. The Nutcracker	0.42	0.91	0.72
	:	:	:
平均(100曲)	0.44	0.60	0.76

6. まとめ

本稿では、確率文脈自由文法に基づく楽曲分析器 σ GTTMIII を提案した。具体的には、762個のルールを作成し、音楽家が作成したデータからその確率を教師付き学習した。そして、得られた確率を用いてタイムスパン木を最尤推定した結果、100曲の平均で σ GTTMIII が従来のタイムスパン木分析器より高い性能であることを確認した。

これまで、タイムスパン木獲得の精度が低かったために、タイムスパン木を用いたアプリケーションを構築することが困難であったが、今後は様々なアプリケーションの構築を進めていきたい。また、本稿ではルールをあらかじめ設定し確率を教師付き学習したが、今後、教師なし学習する手法について検討していく。

参考文献

- [Lerdahl 1983] Lerdahl, F., and Jackendoff, R.: A Generative Theory of Tonal Music. MIT Press, Cambridge, 1983.
- [Charniak 1996] Charniak, E.: Tree-bank grammars, Proceedings of Association for the Advancement of Artificial Intelligence (AAAI-96), pp. 1032-1036, 1996.
- [Hirata 2003] Hirata, K., and Matsuda, S.: Interactive Music Summarization based on Generative Theory of Tonal Music. Journal of New Music Research, 32:2, 165-177, 2003.
- [Hamanaka 2008] Hamanaka, M., Hirata, K., and Tojo, S.: Melody morphing method based on GTTM, Proceedings of the 2008 International Computer Music Conference (ICMC2008), pp. 155-158, 2008.
- [Hirata 2004] Hirata, K., and Matsuda, S.: Annotated Music for Retrieval, Reproduction, and Sharing, Proceedings of International Computer Music Conference, pp. 584-587, 2004.
- [Hamanaka 2006] Hamanaka, M., Hirata, K., and Tojo, S.: Implementing 'A Generative Theory of Tonal Music', Journal of New Music Research, 35:4, 249-277, 2006.
- [Hamanaka 2007] Hamanaka, M., Hirata, K., and Tojo, S.: FATTA: Full Automatic Time-span Tree Analyzer, Proceedings of the 2007 International Computer Music Conference (ICMC2007), pp. 153-156, 2007.
- [Miura 2009] Miura, Y., Hamanaka, M., Hirata, K., and Tojo, S.: Use of Decision Tree to Detect GTTM Group Boundaries, Proceedings of the 2009 International Computer Music Conference (ICMC2009), pp. 125-128, 2009.
- [Kanamori 2014] Kanamori, K., and Hamanaka, M.: Method to Detect GTTM Local Grouping Boundaries based on Clustering and Statistical Learning, Proceedings of the 2014 International Computer Music Conference (ICMC2014), pp. 125-128, 2014.
- [Granroth 2012] Granroth, W.M., and Steedman, M.: Statistical Parsing for Harmonic Analysis of Jazz Chord Sequences, Proceedings of the 2012 International Computer Music Conference (ICMC2012), pp. 478-485, 2012.
- [Tanji 2008] Tanji, M., Ando, D., and Iba, H.: Improving Metrical Grammar with Grammar Expansion, in Australasian Conference on Artificial Intelligence, AI08, LNAI 5360, Springer, pp. 180-191, 2008.
- [Kameoka 2012] Kameoka, H., Ochiai, K., Nakano, M., Tsuchiya M., Sagayama, S.: Context-free 2d Structure Model of Musical Notes for Bayesian Modeling of Polyphonic Spectrograms, Proceedings of the 2012 International Society for Music Information Retrieval Conference (ISMIR2012), pp. 307-312, 2012.