# ガウス過程を利用した
# ピアノ演奏の自動採譜に関する研究

今村 武史

博士（学術）

総合研究大学院大学

複合科学研究科

統計科学専攻

平成２９（２０１７）年度

目　次

# 第 1 章　背景

## 1.1　研究の目的

インターネット等を通じて大量の音楽データを扱うことが可能な時代となり、これらの音楽データから様々な特徴を引き出して楽曲の検索や推薦を行う技術が求められている。このようなニーズと、機械学習の諸技術の発展を背景とし、音楽データを科学的に扱う音楽情報処理の研究が広範に行われている。楽曲の音響信号のみを与え、これを演奏内容の表記へと変換する自動採譜は音楽情報処理の最も基本的な要素技術の一つである。演奏内容の表記の形態は必ずしも五線譜上に表された音符に限らず､MIDI 信号のように鍵盤を打鍵したタイミングや強さを数値で表したものも含まれる（亀岡・嵯峨山（2009))。楽曲の音響信号を音響データ、鍵盤の打鍵タイミングや打鍵の強さ、押鍵した時間などを表した情報を演奏データと呼ぶことにすると、自動採譜は音響データから演奏データを推定する処理であると考えることができる。

自動採譜の研究の歴史は長く、これまで様々な手法が提案されてきたが（Benetos et al.（2013))、現在に至るまで確定的な手法は見出されていない。また近年の研究における推定性能の頭打ちも指摘されており、新たな手法の開発による推定性能の向上が求められている。本研究は既存の自動採譜手法に残された下記の諸課題に取り組み、それらの改善を図るとともに、実用レベルの自動採譜アルゴリズムの実現を目指すものである。

既存研究では、対象の鍵盤が押鍵されているのか、いないのかといった 2 値の結果を推定するものも多く見られた。しかし、演奏の表情についての情報を抽出するためには、打鍵の強さを連続値として推定する必要がある。本研究では楽曲の振幅スペクトル（音響データ）から、打鍵の強さを表す MIDI ベロシティ値（演奏データ）を回帰によって求める手法を検討する。

また自動採譜のように、モノラル録音された混合音から個々の音を聞き分ける困難さについてはこれまでも指摘されてきた（亀岡・嵯峨山（2009)、吉井・糸山

（2015））。混合音のある一瞬の状況を切り取ったフレームにおいて同時に多数の周波数成分が存在し、どの周波数成分がどの音高に由来しているのかを区別することは解が一意に定まらない不良設定問題となる。一つの対処方法としては、混合音の情報をより低次元な変数へと縮約して単純な問題へと置き換え、困難さの緩和を図る方法が考えられる。統計的な手法ではしばしば潜在変数の導入が行われるが、本研究でもこの方向での自動採譜アルゴリズム開発を検討する。

亀岡ら（亀岡・嵯峨山（2009））は自動採譜を、何らかの演奏プロトコル（五線譜上の音符や MIDI 信号など）に従って生成された音響信号から、そのプロトコルを解読する逆プロセスであると位置付けているが、このように多くの既存研究では演奏データを「原因」、音響データを「結果」と捉え、「結果」から「原因」を推定する逆問題として自動採譜の問題を扱ってきた。しかし音響データから演奏データを直接推定しようとする場合、その対応関係は複雑なものとなり、上記の問題が発生してしまう。

一方、 MIDI インターフェイスなどの機器を介して、演奏者による演奏内容が演奏データとしてリアルタイムに得られる場合を想定すると、楽器の発音機構を介して得られる音響データと同様に、演奏データも「結果」、即ち観測変数とみなすことができる。音響データと演奏データはそれぞれ異なる形態をとるが、 両者はともに演奏者による演奏により生じたデータであり、例えば演奏者の意図のような、直接観測できない共通の情報源を「原因」として生成されたものと考えることができる。未知楽曲については音響データのみが観測され、演奏データが欠損している状況に相当する。この場合、自動採譜は観測された音響データに基づき、欠損している演奏データを復元する問題と考えることができる。

音響データ、演奏データをともに観測変数と捉え、未知楽曲の音響データに対する演奏データの推定を、欠損した観測変数の推定問題として扱う手法については、これまで調査されていない。本研究では音響データと演奏データを直接関連付けるのではなく、共通の情報源から音響データおよび演奏データが生成される構造をモデル化し、この構造の中で自動採譜を行う方法を新たに検討する。この共通の情報源を低次元の潜在変数として表すことで、先に述べた問題の困難さを

緩和し、推定性能の向上が期待できる。

混合音の音響データからの演奏データの推定については、各フレーム内の情報のみで音響データと演奏データの対応を考えるのではなく、調波性やスペクトル形状といった周波数方向の先験的知識、あるいは各周波数成分の共起性や動特性といった時間方向の先験的知識の利用や（亀岡・嵯峨山（2009））、音楽知識の利用（亀岡・嵯峨山（2009）、吉井（2016））も提唱されている。このような流れの中で、本研究では時間方向の情報の利用に着目する。

音楽や音声といった音響データは時間方向の連続性を持つ情報であり、自動採譜においてもこの点を考慮した手法が提案されてきた（Poliner and Ellis（2007）、Kameoka（2007））。1 フレームずつ推定を行った場合、各フレームの推定結果はそれぞれ独立しているため、非常に短い継続時間の音の出現や、継続しているはずの音の途切れといった状況が発生するが、これらは時間的な連続性を取り入れた推定により改善することができると考えられる。先行研究では離散的なラベルの時間的連続性に着目したものも多く見られたが（Poliner and Ellis（2007）、Cheng et. al.（2015））、先に述べたように打鍵の強さを回帰によって求めようとする場合には、信号自体の動特性をモデル化する必要があると考える。本研究では音響データ、演奏データを時系列データととらえ、動的なモデルを構築して採譜を行う手法について検討する。

多様な音楽データを学習するためには多数の学習データを必要とする。コンピュータの性能が向上した現代においても、大量の学習データを効率的に学習する手法は必要とされており、さらに楽音のオンセットやオフセットといった発生頻度の低い現象、あるいは低音域や高音域の出現頻度の低い音高への対処も求められる。この観点から、学習用楽曲からあらかじめサンプリングしたデータセットをバッチ式に学習するのではなく、学習用楽曲の全てのフレームを逐次与え、オンラインで学習する手法を開発する。

自動採譜のモデルは、特定の形式のものがあらかじめ想定されているのではなく、これまで様々なモデルが考案されてきた。本論文では上記の課題に対し、高い推定性能を持ったモデルを所与のデータの学習によって構築することを目的に、

優れた表現力を持ったノンパラメトリックな回帰手法であるガウス過程およびその応用手法を用いて新たな自動採譜手法の提案を行い、採譜精度の向上を図る。

以上の内容から、本研究の目的は以下のようにまとめられる。

・押鍵の有無（2 値出力）だけではなく、演奏の表情についての情報も抽出できるよう、打鍵の強さ（ベロシティ値）を連続値として推定する手法を確立する。

・音響データのみならず演奏データも観測変数と捉え、両者が共通の情報源より生成される構造をモデル化し、この構造の中で自動採譜の問題を扱うことで採譜精度の向上を図る。

・音響データ、演奏データを時系列データととらえ、自動採譜の問題を時系列における推定問題として扱い、時間的な情報を利用した推定を行うことで採譜精度の向上を図る。

・オンラインでモデルを学習する手法を確立して多様なデータを大量に学習可能とし、さらに発生頻度の低い現象や出現頻度の低い音高についても効率的に学習可能とする。

本論文は以下の 7 章より成る。

第 1 章は研究の背景として、本研究の目的と自動採譜の研究の歴史について述べている。

第 2 章では、本論文で扱う音楽の音響データおよび演奏データについての説明を行う。

第 3 章はガウス過程による自動採譜に先立ち、識別的な手法である Support Vector Machine（Vapnik（1995））による採譜手法について記している。各鍵盤に対して識別器を割り当てて one vs all 識別機を構成し、それぞれの鍵盤の押鍵の有

無を推定している。Support Vector Machine のような識別的な手法では、対象の鍵盤が押鍵されているのか、いないのかといった 2 値の結果のみが得られ、打鍵の強さは推定できない。この点は、音響データから押鍵の有無だけではなく、打鍵の強さといった演奏の表情についての情報も抽出しようとする場合には大きな制約となる。問題の解決のために打鍵の強さを連続値として推定する必要性を提起し、次章以降の導入としている。

第 4 章ではガウス過程回帰による自動採譜手法について記している。前章で提起された課題に対処するために、ガウス過程回帰によって楽曲の音響データから各鍵盤の打鍵の強さ(MIDI 信号のベロシティ値)を推定している。推定結果には、非線形フィルタの一種である rank order filter による後処理を施して最終的な採譜結果を得ている。採譜実験においては、学習データ数や後処理手法に対する採譜性能を評価している。

第 5 章では Shared Gaussian Process Latent Variable Model による自動採譜手法について記している（今村・松井（2017))。従来の自動採譜の手法は、結果である演奏音から、原因である演奏内容を推定する逆問題という構図で考えられることが多かった。第 4 章ではこの考え方に基づき、「結果」である音響データから「原因」である演奏データを推定する自動採譜アルゴリズムをガウス過程回帰により実現した。しかし電子楽器の演奏時にように、演奏音と同時に MIDI 信号などの形で演奏データが得られる場合には、演奏データも演奏音同様に観測変数と捉えることができる。本章では演奏音の音響データと、MIDI 信号より得られる演奏データが、共通の情報源より発生した異なる形式を持つ観測変数であると捉え、これらの観測変数が共通の潜在変数を共有するモデルを Shared Gaussian Process Latent Variable Model により構築している。未知楽曲の音響データに対する演奏データの推定は欠損した観測データの推定として行われる。本章では、Shared Gaussian Process Latent Variable Model による演奏データの推定と、rank order filter による後処理の組み合わせにより、自動採譜アルゴリズムを構築している。第 4 章と同様に学習データ数、後処理手法に加え、潜在変数の次数が採譜性能の与える影響についても検証している。

第 6 章では Online Gaussian Process Dynamical Model による自動採譜について記している。音は時間と共に変化する情報であり、自動採譜においても演奏音の時間的な連続性を考慮する必要があるとの指摘はこれまでも行われてきた。これらの指摘に対し、本章では第 5 章で導入した音響データ、演奏データに対する潜在変数に時間的な依存性を持たせることを考える。まず時間的な依存性を持たせた潜在変数を状態変数とし、音響データ、演奏データ、状態変数の関係を非線形状態空間モデルで表す。非線形状態空間モデルを構成するシステムモデル、観測モデルは Sparse Online Gaussian Process を多次元化した Multi-output Sparse Online Gaussian Process により実現する。この際、平均および分散をそれぞれ別の回帰モデルによって推定する異分散モデルを構成する。この非線形状態空間モデルに対して、非線形な状態推定手法である Cubature Kalman Filter を適用し、学習用楽曲の音響データ、演奏データを与えて状態推定を行いながら、Multi-output Sparse Online Gaussian Process によってシステムモデル、観測モデルをオンライン学習する。また未知楽曲の音響データのみが与えられた際に、それに対応する演奏データを推定する方法についても併せて提案する。

第 7 章は結論として全体のまとめと考察および今後の課題について述べる。

## 1.2 研究の歴史

自動採譜の研究の歴史は長く、1970 年代の単音フレーズ推定に源流を持つ（Moorer（1975）、Piszczalski and Galler（1979））。その後、単旋律から多重音へと対象の複雑さを増してきたが、手法についても多様化してきた（後藤・平田（2004））。系統的な解説は困難であるが、音響信号から何等かの特徴量を抽出し、演奏されている楽音の候補音を選出し、更にそれらを枝刈りして演奏音を絞り込むという手法は比較的よく用いられてきた（Klapuri（2003）、Pertusa and Iñesta（2008）, Yeh（2008））。

また 1990 年代からは統計的な手法が用いられるようになり、演奏音の事後確率の推定が行われるようになった。我が国でも柏野らの OPTIMA（Kashino（1994）、 Kashino et al.（1995））、後藤の PreFEst（Goto（2004））、亀岡らの harmonic temporal structured clustering（HTC、亀岡 他（2005）、Kameoka et al.（2007））などが提案された。

2000 年代以降によく用いられるようになった手法として、非負値行列分解（Non-negative Matrix Factorization、以下 NMF と表記、Lee and Seung（2001）、Smaragdis and Brown（2003））がある。これは演奏音のスペクトルを時間に沿って並べた正定値行列 $\mathbf{X} \in \Re^{K \times N}$ を、$R$ 個のピッチ成分より成る周波数基底 $\mathbf{W} \in \Re^{K \times R}$ と、時間方向の活性化行列 $\mathbf{H} \in \Re^{R \times N}$ によって以下のように分解し、

$$\mathbf{X} \approx \mathbf{WH} \tag{1.2.1}$$

**W** と **H** から演奏されている音高、タイミングを求めるという手法である。この手法は更に様々な発展形が提案され（Bertin et al.（2010）、O'Hanlon and Plumbley（2014）、Vincent et al.（2010）等）、現在でも研究事例が多い。

機械学習の手法も 2000 年代以降用いられるようになっており、Support Vector Machine（Poliner and Ellis（2007））や、深層学習（Sigtia et al.（2015）、Wang et. al.（2017））といった、その時代の先端の技術が導入されてきた。深層学習については一つの報告の中で Deep Neural Network、Convolutional Neural Network、Recurrent Neural Network といった複数の方式が比較されているものが多く（Sigtia et al.（2015）、Wang et. al.（2017））、どのようなアーキテクチャーが自動採譜に適しているのか模索が続けられている。

どのような手法を用いるにせよ、従来のような一つのフレーム内の情報のみで音響データと演奏データの対応を考えるのではなく、様々な先験的情報を利用して推定精度の向上を図ろうとする試みが現在の自動採譜研究の動向であるように思われる。スペクトルのスパース性に着目した NMF、調波性等の周波数方向の先験情報や共起性、動特性といった時間方向の先験的知識の利用（亀岡・嵯峨山（2009））の提唱等は、このような流れの中に位置づけられるものであると考えられる。また、自動採譜と音声認識のタスクの類似性に着目し、音声認識における言語モデルに相当する音楽モデルを構築して、音楽知識を積極的に自動採譜に利

用しようという提案も行われている（亀岡・嵯峨山（2009)、吉井（2016)）。このような自動採譜における様々な先験的情報の利用については今後も継続され、発展していくものと考えられる。自動採譜問題への潜在変数の導入や、時間情報の利用に取り組んだ本研究は、この流れの一つとして位置づけられるものである。

以下では、自動採譜以外の音楽情報処理のタスクについても簡単に触れる（亀岡・中村・高宗（2015)）。自動採譜に近いタスクとしては、和音推定や楽譜追跡が挙げられる。和音推定はクラッシック音楽の和音やポピュラー音楽のコードネームを推定する処理であり、厳密な演奏内容ではなく、演奏されている和音がどのような音名で構成されているかを推定する（Wakefield（1999)、Fujishima（1999)）。このタスクではクロマグラムと呼ばれる和音の特徴を表した特徴量がよく用いられる。また一時的な調性や和声からの逸脱を吸収するために隠れマルコフモデルなどによる平滑化が行われる場合もある（Sheh and Ellis（2003)）。

楽譜追跡は与えられた演奏音を実時間で処理しながら、楽譜上の位置を推定するタスクであり、自動伴奏や自動譜めくり等の実現を目的としている（Dannenberg（1984)、Vercoe（1984)）。同じ楽譜に基づいた演奏であっても、人間の演奏にはテンポや強弱、演奏ミスなどの不確定要素が存在し、毎回異った演奏音の音響信号が生成される。これらの不確定性を扱うために統計的なモデルがしばしば用いられる。

また音の高さに関する情報だけではなく、リズムについての推定も行われている。代表的なものとしてテンポ推定（武田・西本・嵯峨山（2004)、Kameoka et. al.（2012)、高宗・亀岡・嵯峨山（2014)）、拍の推定（ビートトラッキング、Grosche et. al.（2010)、Goto（2001)）といったタスクがある。

楽曲自体の構造解析（Paulus et. al.（2010)）や、より深い音楽理解を目指した研究も行われている。音声認識における言語モデルに相当する音楽文法モデルを作成し、楽譜の背後にある階層構造を推定しようという試みも行われている（Yoshii and Goto（2011)、Kameoka et. al.（2012)、Nakamura et. al.（2016)）。

これらのタスクは独立したものであるが、それぞれのタスクで得られた知見を相互に取り入れることで、個々のタスクの性能向上に繋がるものと考えられ、今

後の音楽情報処理の応用範囲の拡大が期待できる。

# 第 2 章　自動採譜に用いる音楽データ

音楽のデータ化には、演奏の結果発せられる演奏音のデータ化と、演奏内容自体のデータ化があり、本研究ではそれぞれのデータを音響データ、演奏データと呼ぶ。本章では音響データ、演奏データについて説明する。更に本研究で使用した MIDI Aligned Piano Sounds データベース（Emiya et al（2010），以下 MAPS と表記）についても説明する。

## 2.1　音響データ

### 2.1.1　オーディオデータ

音声や楽器の演奏音は空気の圧力変化の波動であり、時間、変位ともに連続したアナログ信号である。これを計算機上で扱うためには、一定の時間間隔（標本化周期）で信号を標本化し、更に変位を量子化してディジタル信号に変換する必要がある（図 2.1）。この時、ディジタル信号として記録できる最高周波数と標本化周期との間には標本化定理で示される関係があり、また量子化の分解能も音質に影響を与えることが知られている（斎藤・中田（1981））。市販されている一般的な音楽 CD では、標本化周波数（標本化周期の逆数）44.1kHz、量子化 16 ビット（$2^{16}$=65536 段階に離散化）という仕様が採用されている。Windows 内ではオーディオデータは wav、mp3 といった形式で扱われるが、本研究では波形圧縮を行わない wav 形式のデータを用いる。

### 2.1.2　周波数分析

オーディオデータから音声の発話内容や楽器の演奏内容を読み取ろうとする場合、信号内にどのような周波数成分が含まれているのか分析することがよく行われる。しかしこれらの情報は、オーディオデータから直接得られるものではないので、オーディオデータに対して何らかの周波数分析を施す必要がある。周波数分析の方法としてはフーリエ変換やウェーブレット変換、フィルタバンクによる

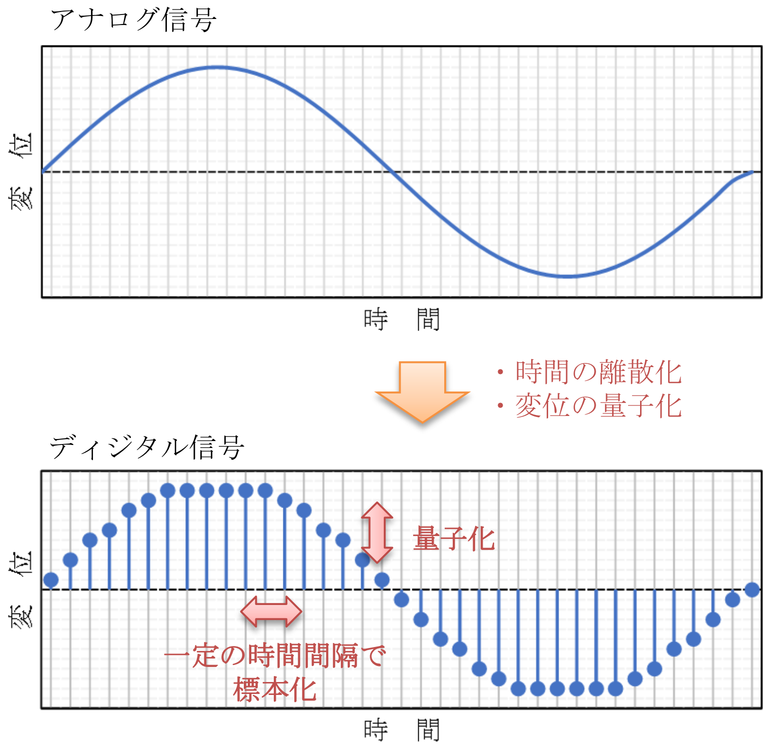



図 2.1 アナログ信号からディジタル信号への変換

分析といった手法もあるが（河原（1991））、本研究では最も基本的な手法で古くから広く用いられている離散フーリエ変換（Discrete Fourier Transform，以下 DFT と表記）による周波数分析を採用する。

$$X[k] = \sum_{n=0}^{N-1} x[n]w[n]\exp\left(-j\frac{2\pi kn}{N}\right) \quad (k=0,\ldots,N\text{-}1,\ j\ \text{は虚数単位}) \tag{2.1.1}$$

ここで $x[n]$はオーディオデータ、$X[k]$は第 $k$ 番目の周波数成分である。また $w[n]$は切り出し区間の端の影響を軽減するための窓関数であり、本研究では以下に示す hanning 窓を採用した。

$$w[n] = 0.5 - 0.5\cos\left(\frac{2\pi n}{N-1}\right) \tag{2.1.2}$$

式（2.1.1）より明らかなように $X[k]$は複素数であり、各周波数成分の位相情報

も含んでいるが、実際の分析にあたっては成分ごとの振幅またはパワーが得られれば十分であることも多い。本研究では式（2.1.3）より求めた振幅スペクトルを音響データとして用いる。

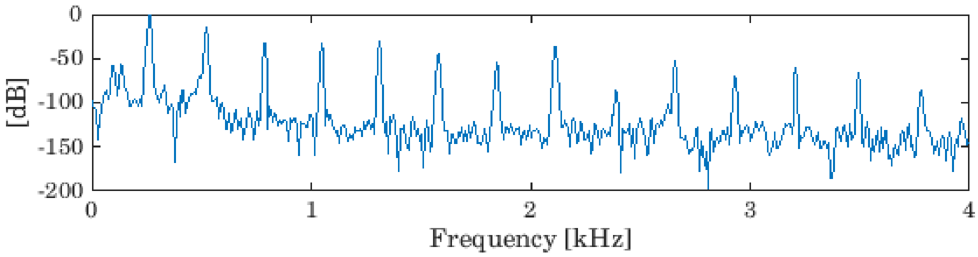
$$\overline{X}[k] = \sqrt{Re(X[k])^2 + Im(X[k])^2} \tag{2.1.3}$$

なお、実際の DFT の計算においては、DFT の高速計算アルゴリズムである高速フーリエ変換（Fast Fourier Transform，以下 FFT と表記）を使用する。この場合、$N$ は 2 のベキ乗となる。

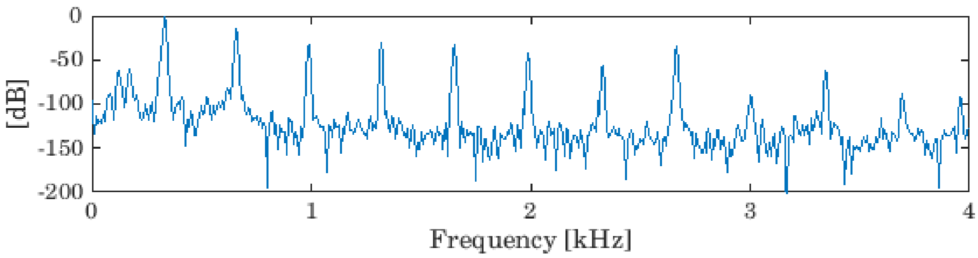
図 2.2 にピアノの C4 音（基本周波数 261.6Hz）、E4 音（基本周波数 329.6Hz）、G4 音（基本周波数 392.0Hz）および C4 音，E4 音，G4 音の和音（ドミソの和音）の振幅スペクトルをそれぞれ示す。各単音の振幅スペクトルでは、基本周波数成分のみならず、基本周波数成分の整数倍の周波数を持つ成分（高調波成分）が現れていることが分かる。また和音の振幅スペクトルでは各単音の高調波成分が重なっていることが分かる。和音における各周波数成分の振幅は、単音の各周波数成分の振幅の単純な和にはならず、重畳するタイミング（位相）によって変化するため、その形状は複雑に変化する。このような未知和音の振幅スペクトルから、その構成音を推定することが自動採譜（音高推定）の目的である。

### 2.1.3　時間-周波数表現

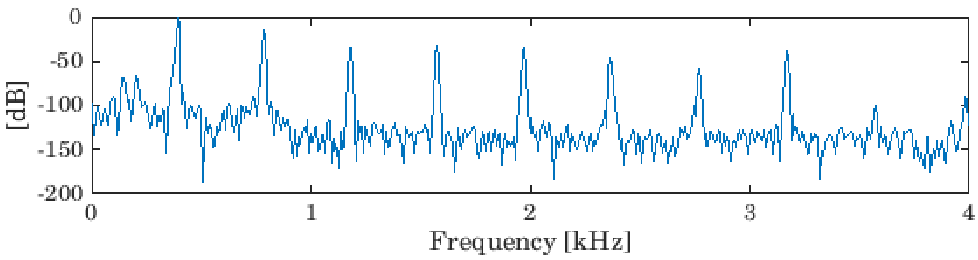
離散フーリエ変換では、切り出した区間内では各周波数成分は定常であると仮定しているため、時間的に局在する周波数成分や、各周波数成分の振幅が時間的に変化する場合の分析は行えない。このような状況を回避するために、短い区間を少しずつシフトしながら離散フーリエ変換を行う短時間フーリエ変換が広く用いられてきた（図 2.3）。この結果、式（2.1.3）より得られる振幅スペクトルは周波数方向への広がりとともに、時間方向への広がりを持った変数 $\overline{X}[k,i]$ へと拡張される（$i$ は離散化された時間の番号）。図 2.4 に振幅スペクトルの時間-周波数表現（スペクトログラム）を示す。色が明るい程、振幅が大きいことを示している。時間とともに各周波数成分の振幅が変化する様子が分かる。



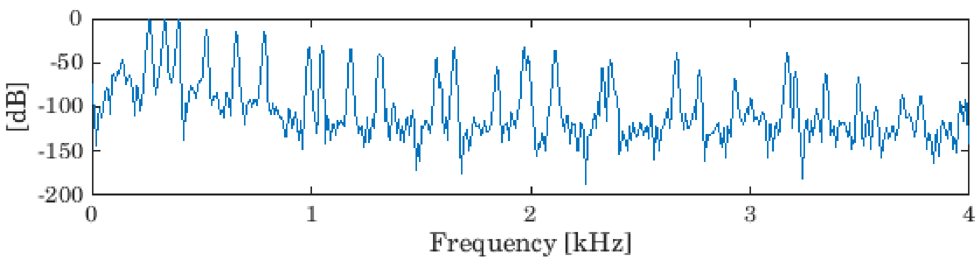

（a） C4 音（基本周波数 261.6Hz）の振幅スペクトル




（b） E4 音（基本周波数 329.6Hz）の振幅スペクトル




（c） G4 音（基本周波数 392Hz）の振幅スペクトル




（d） C4E4G4 の和音（ドミソの和音）の振幅スペクトル

図 2.2　ピアノ単音と和音の振幅スペクトル

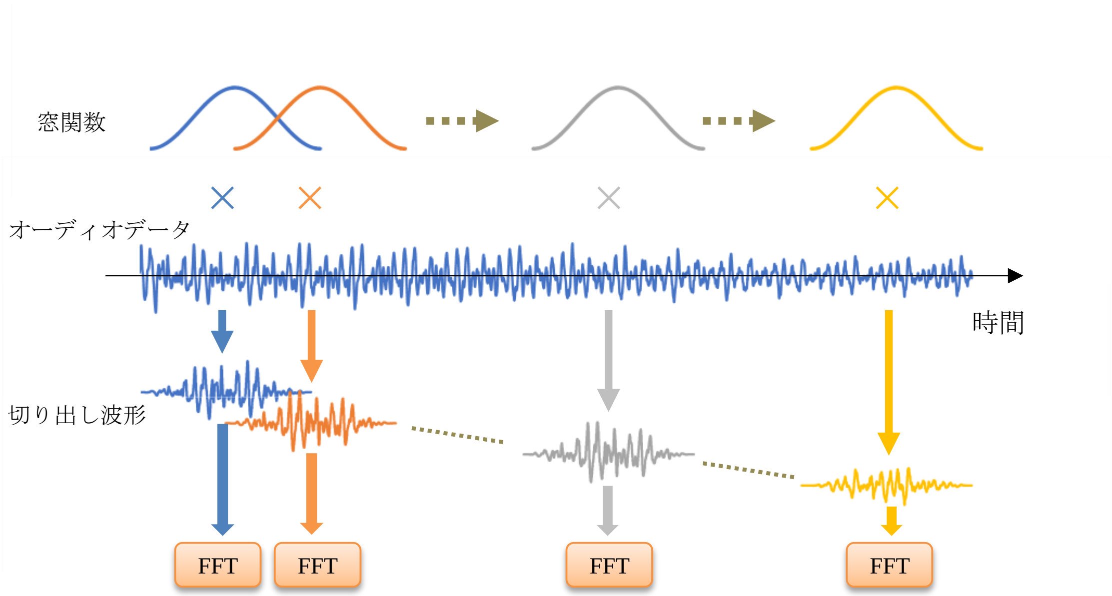



図 2.3　短時間フーリエ変換

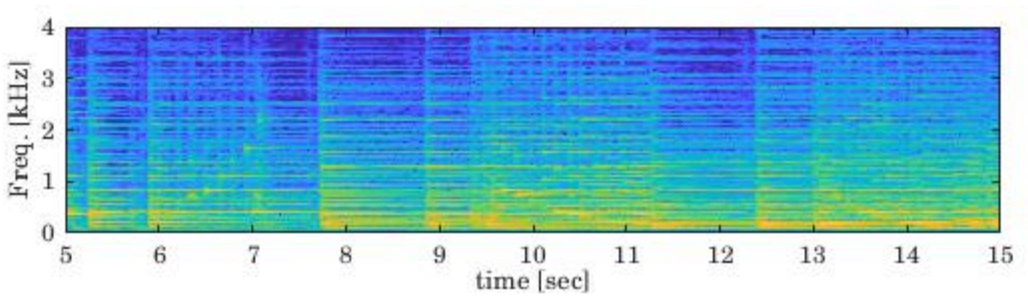



図 2.4　振幅スペクトルの時間-周波数表現（スペクトログラム）の例

## 2.2　演奏データ

多くの電子楽器では、楽器に対して行った演奏行為を情報として出力することができる。また専用のインターフェイスを装着することにより、アコースティック楽器からも演奏情報を得ることができる場合がある。以下では演奏情報の信号規格であるMIDIと、演奏情報の表現形式であるピアノロールについて説明する。

### 2.2.1　MIDI データ

MIDI（Musical Instrument Digital Interface，一般社団法人音楽電子事業協会

（2016)）は電子楽器やコンピュータ間で演奏情報や音色情報、コントロール情報を転送するための規格である。多数の項目から構成されるが、ピアノ演奏の場合、どの鍵盤（MIDI ノートナンバー）を、どの時点で打鍵（ノートオン）し、どの時点で離鍵（ノートオフ）したか、また打鍵時の強さ（ベロシティ）はどのくらいであったか、といった鍵盤に対する操作情報が記録されれば十分である。88 鍵のピアノの場合、MIDI ノートナンバーは 21～108 に対応している。またベロシティ値の範囲は 0～127 である。各 MIDI ノートナンバーのベロシティ値は対応する鍵盤を演奏者が押鍵している間維持され、離鍵すると 0 となる。

表 2.1　ピアノ（88 鍵）の音名、MIDI ノートナンバー、基本周波数の対応

| 音名 | MIDI ノートNo. | 周波数 (Hz) | 音名 | MIDI ノートNo. | 周波数 (Hz) | 音名 | MIDI ノートNo. | 周波数 (Hz) | 音名 | MIDI ノートNo. | 周波数 (Hz) | 音名 | MIDI ノートNo. | 周波数 (Hz) |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | C2 | 36 | 65.4 | C4 | 60 | 261.6 | C6 | 84 | 1046.5 | C8 | 108 | 4186 |
| | | | C#2 | 37 | 69.3 | C#4 | 61 | 277.2 | C#6 | 85 | 1108.7 | | | |
| | | | D2 | 38 | 73.4 | D4 | 62 | 293.7 | D6 | 86 | 1174.7 | | | |
| | | | D#2 | 39 | 77.8 | D#4 | 63 | 311.1 | D#6 | 87 | 1244.5 | | | |
| | | | E2 | 40 | 82.4 | E4 | 64 | 329.6 | E6 | 88 | 1318.5 | | | |
| | | | F2 | 41 | 87.3 | F4 | 65 | 349.2 | F6 | 89 | 1396.9 | | | |
| | | | F#2 | 42 | 92.5 | F#4 | 66 | 370 | F#6 | 90 | 1480 | | | |
| | | | G2 | 43 | 98 | G4 | 67 | 392 | G6 | 91 | 1568 | | | |
| | | | G#2 | 44 | 103.8 | G#4 | 68 | 415.3 | G#6 | 92 | 1661.2 | | | |
| A0 | 21 | 27.5 | A2 | 45 | 110 | A4 | 69 | 440 | A6 | 93 | 1760 | | | |
| A#0 | 22 | 29.1 | A#2 | 46 | 116.5 | A#4 | 70 | 466.2 | A#6 | 94 | 1864.7 | | | |
| B0 | 23 | 30.9 | B2 | 47 | 123.5 | B4 | 71 | 493.9 | B6 | 95 | 1975.5 | | | |
| C1 | 24 | 32.7 | C3 | 48 | 130.8 | C5 | 72 | 523.3 | C7 | 96 | 2093 | | | |
| C#1 | 25 | 34.6 | C#3 | 49 | 138.6 | C#5 | 73 | 554.4 | C#7 | 97 | 2217.5 | | | |
| D1 | 26 | 36.7 | D3 | 50 | 146.8 | D5 | 74 | 587.3 | D7 | 98 | 2349.3 | | | |
| D#1 | 27 | 38.9 | D#3 | 51 | 155.6 | D#5 | 75 | 622.3 | D#7 | 99 | 2489 | | | |
| E1 | 28 | 41.2 | E3 | 52 | 164.8 | E5 | 76 | 659.3 | E7 | 100 | 2637 | | | |
| F1 | 29 | 43.7 | F3 | 53 | 174.6 | F5 | 77 | 698.5 | F7 | 101 | 2793.8 | | | |
| F#1 | 30 | 46.2 | F#3 | 54 | 185 | F#5 | 78 | 740 | F#7 | 102 | 2960 | | | |
| G1 | 31 | 49 | G3 | 55 | 196 | G5 | 79 | 784 | G7 | 103 | 3136 | | | |
| G#1 | 32 | 51.9 | G#3 | 56 | 207.7 | G#5 | 80 | 830.6 | G#7 | 104 | 3322.4 | | | |
| A1 | 33 | 55 | A3 | 57 | 220 | A5 | 81 | 880 | A7 | 105 | 3520 | | | |
| A#1 | 34 | 58.3 | A#3 | 58 | 233.1 | A#5 | 82 | 932.3 | A#7 | 106 | 3729.3 | | | |
| B1 | 35 | 61.7 | B3 | 59 | 246.9 | B5 | 83 | 987.8 | B7 | 107 | 3951.1 | | | |

厳密には鍵盤以外にペダル操作があるが、ダンパーペダルを踏み込んで音を伸ばしている期間は離鍵してもノートオフ情報を記録せず、ペダルを踏み離した時点で離鍵したものとみなしノートオフ情報を記録するといった対応をとる。他の

ペダル（ソステヌートペダル、シフトペダル）の操作については本研究では扱わない。

なお、本論文では楽音としての音の高さを A0（88 鍵のピアノの最低音）,...,C4（真中のド）,...,C8（88 鍵のピアノの最高音）といった音名で表す。音名と鍵盤（従って MIDI ノートナンバーも）との対応についてはいくつかの規格があるが、本論文では国際式（ISO 16（1975））を用いる。表 2.1 に 88 鍵のピアノの音域に対応した音名、MIDI ノートナンバー、基本周波数の対応を示す。

### 2.2.2　ピアノロール表現

コンピュータ上での音楽制作の場面では、縦方向に MIDI ノートナンバー（または音高）を、横方向に時間をとった表形式のインターフェイス上で MIDI データを編集することがしばしば行われる（図 2.5）。このような MIDI データの表現はピアノロール表現と呼ばれるが、ピアノロール表現は MIDI ノートナンバー軸（または音高軸）と時間軸上に MIDI データを配置し、行列形式で表現したものとも考えることができる。本論文での演奏データは、各鍵盤（MIDI ノートナンバー）のベロシティ値を、音響データと同じ時間分解能で展開したピアノロール表現したものを用いる。

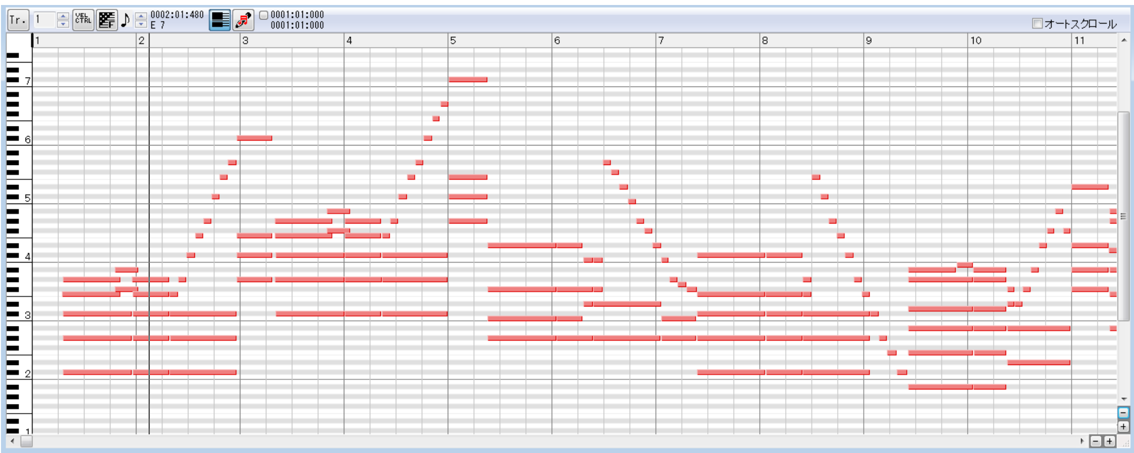



図 2.5　ピアノロールの例

## 2.3　本研究で扱う音楽データベースについて

本稿での自動採譜の学習およびテストには、近年の自動採譜の研究（Benetos and Weyde（2013）、Berg-Kirkpatrick et al.（2014）、Cheng et al.（2015）、Emiya et al.（2010）、O'Hanlon and Plumbley（2014）、Sigtia et al.（2015）、Vincent et al.（2010））で使用されている MAPS database（Emiya et al.（2010））を使用する。

MAPS database で提供されているデータは、MIDI インターフェイスを装着したアコースティックピアノによる演奏音と MIDI データを同時に記録したもので、単音や和音、楽曲演奏のオーディオデータ（wav 形式）と MIDI データ（mid 形式および txt 形式）で構成されている。既出論文と同様に、本稿の実験も ENSTDkAm フォルダに置かれた楽曲を学習用データとして、ENSTDkCl フォルダに置かれた楽曲をテスト用データとして使用する。各フォルダ内には 30 曲分のデータが置かれており、既出論文と同条件となるよう、各楽曲の冒頭 30 秒間を使用する。以下に音響データ、演奏データの作成方法について述べる。

### 2.3.1　音響データ

MAPS database のオーディオデータはサンプリング周波数 44.1kHz、量子化 16 ビットのステレオデータである。これをモノラル化するために、左右の波形の平均波形を求める。続いて窓長 4096 点、シフト数 512 点の短時間フーリエ変換を行ってスペクトログラムに変換した。シフト数は既出論文（Berg-Kirkpatrick et al.（2014）、Sigtia et al.（2015））と同じ値とした。

楽曲 $m$ の第 $l$ フレーム、第 $j$ 成分の音響データ $y_{m,l}^{(j)}$ は、短時間フーリエ変換で得られたスペクトログラムの振幅 $s_{m,l}^{(j)}$ の dB 値 $S_{m,l}^{(j)}$ より下記の式によって求める。

$$y_{m,l}^{(j)} = \begin{cases} (S_{m,l}^{(j)} + 110)/110 & (-110 \le S_{m,l}^{(j)} \le 0) \\ 0 & (S_{m,l}^{(j)} < -110) \end{cases} \qquad (2.3.1)$$

但し　$S_{m,l}^{(j)} = 20\log_{10}(s_{m,l}^{(j)} / s_{\max})\ \ [dB]$

$s_{\max}$ は学習用楽曲のスペクトログラムにおける最大振幅である。学習用楽曲内に

おける $s_{m,l}^{(j)}$ のレベルの範囲は、$s_{\max}$ に対しておおよそ-110dB〜0dB であったため、$y_{m,l}^{(j)}$ の値が 0〜1 の範囲に収まるよう、110dB で正規化を行っている。

### 2.3.2 演奏データ

各楽曲の MIDI データから、音響データの時間分解能（512/44.1kHz≒11.6msec）ごとに押鍵されている鍵盤の MIDI ノートナンバーを調べ、押鍵されている MIDI ノートナンバーには打鍵時のベロシティ値を 127（ベロシティ値の最大値）で割った値を、押鍵されていないものには 0 をそれぞれ与え、ピアノロールに対応した行列形式で演奏データを作成した。図 2.6 に演奏データの例を示す。演奏内容の変化とともに押鍵されている MIDI ノートナンバーが変化し、打鍵の強さによって記録される値が変化している様子が分かる。

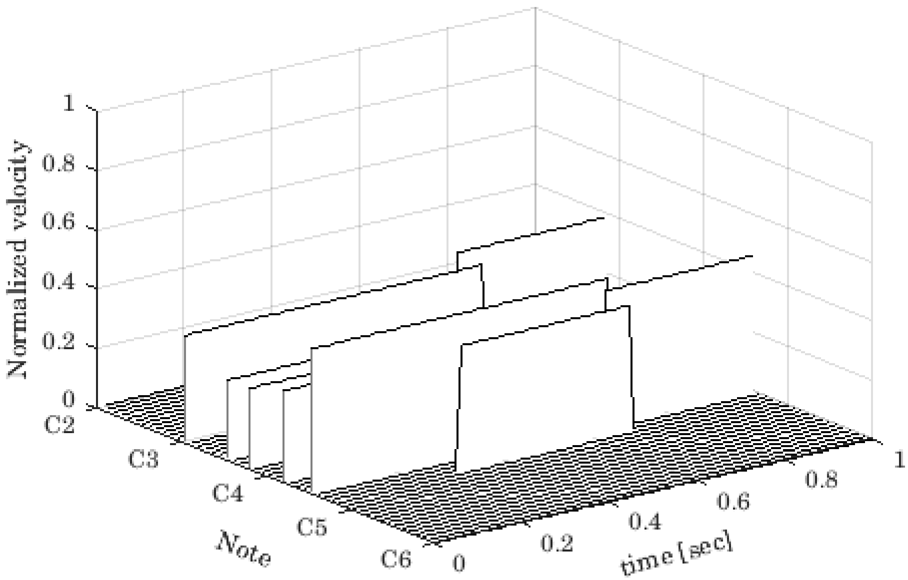



図 2.6　演奏データの例

# 第 3 章　Support Vector Machine による自動採譜

## 3.1　はじめに

ガウス過程での自動採譜に先立ち、識別的な手法である Support Vector Machine（Vapnik（1995）、Cristianini and Shawe-Taylor（2005）、以下 SVM と表記）による自動採譜について説明する。実験結果とともにこの手法の問題点を明らかにし、次章以降の導入とする。

## 3.2　　Support Vector Machine（SVM）

SVM は、元の空間では線形分離困難なデータ群を非線形変換によって高次元の特徴空間へ写像し、特徴空間において線形識別器を構成する手法である。特徴量間の内積をカーネル関数で置き換えることで、高次元の特徴空間でのモデル化を実現している。写像先の特徴空間において各データ群との距離（マージン）が最大となる識別面（識別超平面）を求めて線形分離を行うことで非常に高精度な識別性能を得ることができるが（図 3.1）、この最適化もカーネル関数の導入により容易に行われる。また識別面はサポートベクトルと呼ばれる少数のベクトルによって規定されるため、スパースなモデルを作ることができる。

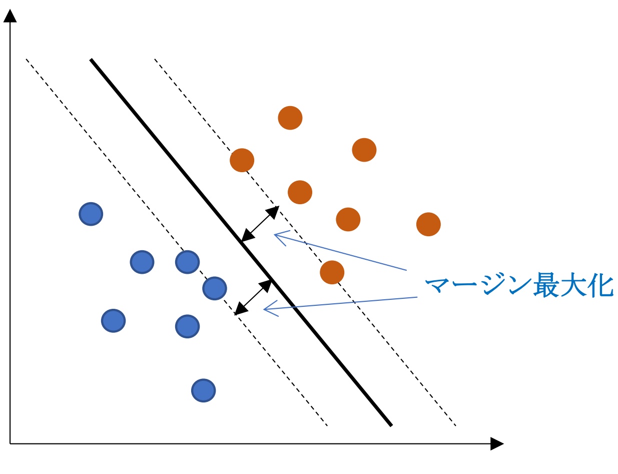



図 3.1　SVM の概念

本来 SVM は 2 値識別器だが、多クラスの識別問題への適用には、2 値識別 SVM を複数組み合わせた One vs All 識別器（Schölkopf and Smola (2001)）などがよく用いられる。

### 3.3　SVM による自動採譜アルゴリズム

Poliner らは、ピアノ鍵盤分の 88 個の学習データセットと SVM を用意し、音高ごとにモデルを学習して未知楽曲の自動採譜を行った（Poliner and Ellis (2007)）。学習データは推定対象の音高が含まれているフレームを正例、含まれていないフレームを負例として収集した。これを SVM で学習し、入力として与えた未知のフレームに推定対象の音高が含まれているか否かを推定する識別機を音高ごとに作成した。更にこれらを組み合わせて One vs All 識別器を構成し、テスト用楽曲の音高推定を行っている。本章では、Poliner らと同様の自動採譜アルゴリズムを構築し、採譜実験を行う。識別器構成を図 3.2 に示す。

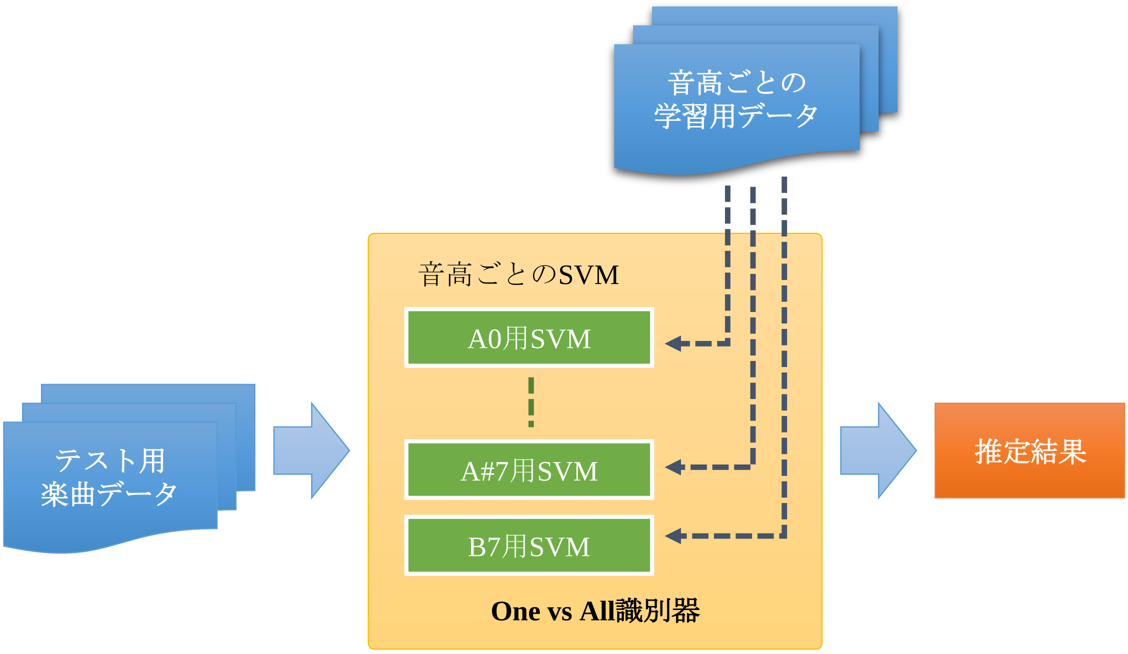



図 3.2　SVM による One vs All 識別器の構成

### 3.3.1　特徴量

実験に用いたデータは、Classical Piano Midi Page（Krueger）より入手した。各楽曲の MIDI データの冒頭 60 秒間を、パーソナルコンピュータ上の Apple 社 iTunes を使用して wav 形式のモノラルオーディオデータに変換した。オーディオデータの仕様は量子化 16bit、サンプリング周波数 8kHz である。これらの楽曲データをそれぞれ学習用、テスト用、バリデーション用のデータセットに分割した。

次に、オーディオデータから窓長 1024 点（Hanning 窓）、シフト数 80 点の短時間フーリエ変換によって振幅スペクトルを求め、更に周波数方向に 71 点の滑走窓による正規化（平均を引き、標準偏差で割る）を行って特徴量を作成した。この処理は Poliner and Ellis (2007)で行われているものと同様である。学習データは、短時間フーリエ変換による学習用楽曲のスペクトル全体より、正例 2250 フレーム、負例 2250 フレームを無作為に選択して作成した。収集可能な正例のフレーム数が上記の数に満たない低音域、高音域の音高については全てのフレームを収集した。この場合も正例、負例は同数とした。またピアノの最高音である C8 の基本周波数は 4186.0Hz であり、ナイキスト周波数を超えてしまうので、半音低い B7(基本周波数は 3951.1Hz)まで学習データを作成した。なお学習及び音高推定に際しては、上記の方法によって求めた特徴量から音高に応じて下記の周波数帯域を切り出して使用した。音高推定を行う際には、テスト用楽曲の特徴量から同じ帯域を切り出して SVM に与えた。

A0～B5　:　0～2kHz

C6～B6　:　1～3kHz

C7～B7　:　2～4kHz

### 3.3.2　モデルの学習

前項で説明した学習用の音響データセットを音高分用意し、音高ごとに 87 個の SVM をそれぞれ学習して One vs All 識別器を構成した。学習に際しては(3.3.1)式で示される RBF カーネルを用い、バリデーションデータセットに対するグリッドサーチによってカーネル関数のパラメータ $\gamma$ および学習時の正則化係数を決定し

た。なお SVM の実装には libsvm（Chang）を使用した。

$$k(\mathbf{x},\mathbf{y}) = \exp\left(-\gamma\|\mathbf{x}-\mathbf{y}\|^2\right) \tag{3.3.1}$$

### 3.3.3　音高ごとの押鍵の有無の推定

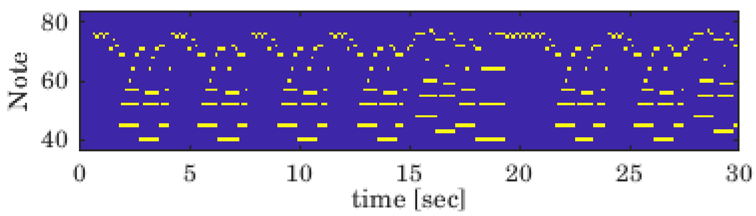
学習したモデルに対して未知楽曲の音響データを与え、音高ごとに押鍵の有無を推定した。推定精度は Poliner and Ellis (2007)と同様に、(3.3.2)式で与えられる Acc で計算した。

$$\mathrm{Acc} = \frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FP}+\mathrm{FN}} \tag{3.3.2}$$

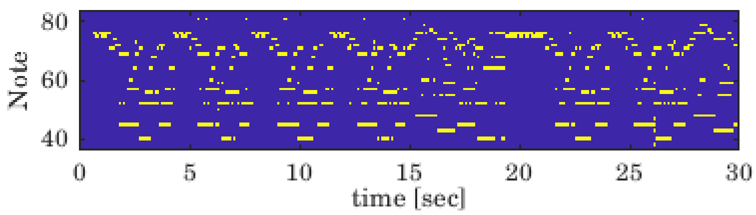
ここで TP，FP，FN は、正常に対象音高が検出されたフレーム数、実際には演奏されていないが検出されたフレーム数、実際に演奏されているが検出されなかったフレーム数をそれぞれ表す。

## 3.4　　実験結果

図 3.3 に推定例を示す。テスト用楽曲全体に対する Acc は 0.72 であった。




(a)正解データ




(b)演奏データ推定値

図 3.3　SVM による自動採譜例

### 3.5　まとめと考察

本章では識別的手法である SVM によって押鍵の有無を推定する SVM を音高ごとに学習し、これらを組み合わせて One vs All 識別器を構成して未知楽曲の自動採譜を行った。その結果、テスト用楽曲全体に対して 0.72 の Acc で推定を行うことができた。

しかし図 3.3 を見ると、楽曲の全体的な内容は推定できているものの、細部を見ると正解データには存在しなかった音高が検出されていたり、逆に正解データには存在しているが検出されなかった音高があることが分かる。ではこれらの誤検出を極力削減し、識別器としての性能を向上させれば自動採譜アルゴリズムとして十分であろうか？

図 3.4 はあるピアノ楽曲の実際の楽譜である。中心的な情報として、どの音高を、どのタイミングで、どのくらいの期間押さえるかといった内容が記述されているが、その他にもメゾピアノ、メゾフォルテ等の強弱記号やクレッシェンド、デクレッシェンドといった、楽曲の表情についての指定も記述されている。また直接楽譜に書き込まれていない場合でも、演奏者の解釈によって打鍵時の強さを


図 3.4　ピアノ楽曲の楽譜例

変化させることも行われる。

SVM は 2 値識別器であり、対象の音高が押鍵されているのか、押鍵されていないのかという推定しか行えず、音響データから演奏の表情を読み取ることはできない。演奏の表情についての情報を抽出するには、打鍵の強さを連続値として推定する必要がある。この点を踏まえ、以降の章では自動採譜を識別問題ではなく、回帰問題として扱う手法について説明する。

# 第 4 章　ガウス過程による自動採譜

## 4.1　はじめに

前章で示した SVM による自動採譜の例では、音高ごとに識別器を置き、各鍵盤の押鍵の有無を推定した。即ち自動採譜問題を 2 値識別問題として扱った。しかし前章の考察で指摘したように、SVM によって推定できる情報は各鍵盤の打鍵および離鍵のタイミングのみであり、演奏の表情、即ち打鍵の強さについての情報は得られない。与えられたピアノ演奏の音響データから演奏の表情についての情報を抽出しようとする場合、打鍵の強さについても推定する必要がある。この点に対処するために、本章では回帰問題として自動採譜を扱う手法を提案する。

多くの既存研究では演奏データを「原因」、音響データを「結果」と捉え、「結果」から「原因」を推定する逆問題として自動採譜の問題を扱ってきた（亀岡・嵯峨山（2009））。ある鍵盤をある強さで打鍵した結果、打鍵の強さに対応した振幅スペクトルが発生したと考えれば、打鍵の強さ（演奏データ）を「原因」、発生した振幅スペクトル（音響データ）を「結果」と捉えることができる。本章では、ガウス過程によって「結果」である音響データから、「原因」である演奏データを、回帰により推定する。即ち音響データである振幅スペクトルを与え、対応する鍵盤のベロシティ値を推定する回帰モデルを鍵盤ごとに作成することで自動採譜アルゴリズムを構築する。回帰モデルにはガウス過程回帰（Rasmussen and Williams（2006）、Bishop（2007））を用いる。

## 4.2　ガウス過程回帰

本節ではガウス過程について簡潔に説明する。ガウス過程は関数 $y(\mathbf{x})$上の確率分布として定義され、任意の点集合 $\mathbf{x}=\{\mathbf{x}_1,\ldots,\mathbf{x}_N\}$に対する関数の出力 $y(\mathbf{x})$の値の同時分布がガウス分布に従うとしたものである（Bishop（2007））。ガウス過程回帰を考えるには、まず以下のようなノイズの重畳した回帰モデルを考える。

$$y = \mathbf{w}^T \varphi(\mathbf{x}) + \varepsilon \quad , \quad \varepsilon \sim N(\varepsilon \mid 0, \beta^{-1}) \tag{4.2.1}$$

$f(\mathbf{x})=\mathbf{w}^T\varphi(\mathbf{x})$とし、$\mathbf{w}$ の事前分布を $\mathrm{N}(\mathbf{w}|0,\alpha^{-1}\mathbf{I})$とすると、$f(\mathbf{x})$の平均，分散はそれぞれ、

$$\mathrm{E}[f(\mathbf{x})] = \mathrm{E}[\mathbf{w}^T \varphi(\mathbf{x})] = \mathrm{E}[\mathbf{w}^T]\varphi(\mathbf{x}) = 0 \tag{4.2.2}$$

$$\begin{aligned} \mathrm{E}[f(\mathbf{x}) f(\mathbf{x}')] &= \mathrm{E}[(\mathbf{w}^T \varphi(\mathbf{x}))^T (\mathbf{w}^T \varphi(\mathbf{x}'))] \\ &= \varphi(\mathbf{x})^T\, \mathrm{E}[\mathbf{w}\mathbf{w}^T]\varphi(\mathbf{x}') = \alpha^{-1}\varphi(\mathbf{x})^T \varphi(\mathbf{x}') = k(\mathbf{x}, \mathbf{x}') \end{aligned} \tag{4.2.3}$$

となる。ただし $k(\mathbf{x},\mathbf{x}')$ はカーネル関数である。ここで $\mathbf{X}_N=[\mathbf{x}_1,\ldots,\ \mathbf{x}_N]^T$，$\mathbf{f}_N=[\mathbf{w}^T\varphi(\mathbf{x}_1),\ldots,\mathbf{w}^T\varphi(\mathbf{x}_N)]^T$，$\mathbf{x}_i\in \mathrm{N}\mathfrak{R}^D$ 、$\mathbf{Y}_N=[y_1,\ldots,y_N]^T$，$\mathbf{y}_i\in\mathfrak{R}^Q$ とし、$\mathbf{K}$ を$(i,j)$要素が $k(\mathbf{x}_i,\mathbf{x}_j)$のカーネル行列とすると、

$$p(\mathbf{Y}_N \mid \mathbf{f}_N) = N(\mathbf{Y}_N \mid \mathbf{f}_N, \beta^{-1}\mathbf{I}) \tag{4.2.4}$$

$$p(\mathbf{f}_N \mid \mathbf{X}_N) = N(\mathbf{f}_N \mid \mathbf{0}, \mathbf{K}) \tag{4.2.5}$$

ノイズの独立性を考慮して周辺分布 $p(\mathbf{Y}_N|\mathbf{X}_N)$は、

$$p(\mathbf{Y}_N \mid \mathbf{X}_N) = \int p(\mathbf{Y}_N \mid \mathbf{f}_N) p(\mathbf{f}_N \mid \mathbf{X}_N)\, \mathrm{d}\mathbf{f}_N = N(\mathbf{Y}_N \mid \mathbf{0}, \mathbf{C}_N) \tag{4.2.6}$$

と求まる。ただし、$\mathbf{C}_N$ は$(i,j)$要素を $k(\mathbf{x}_i,\mathbf{x}_j)+\beta^{-1}\delta(i,j)$とする分散共分散行列であり、$\delta(i,j)$はクロネッカーのデルタである。

ここで新たな入力 $\mathbf{x}_{N+1}$ が得られたとし、それに対応する出力を $y_{N+1}$ とすると、同時分布 $p(\mathbf{Y}_{N+1}|\mathbf{X}_{N+1})$は、

$$\begin{aligned} p(\mathbf{Y}_{N+1} \mid \mathbf{X}_{N+1}) &= N(\mathbf{Y}_{N+1} \mid \mathbf{0}, \mathbf{C}_{N+1}) \\ &= \frac{1}{\sqrt{(2\pi)^D |\mathbf{C}_{N+1}|}} \exp\left( -\frac{1}{2} (\mathbf{Y}_N, y_{N+1})^T \begin{pmatrix} \mathbf{C}_N & \mathbf{k}_N \\ \mathbf{k}_N^T & c_{N+1} \end{pmatrix}^{-1} (\mathbf{Y}_N, y_{N+1}) \right) \end{aligned} \tag{4.2.7}$$

となる。ただし、$\mathbf{k}_N$ は $i$ 要素が $k(\mathbf{x}_i,\mathbf{x}_{N+1})$ （$i=1,\ldots,N$） のベクトル、$c_{N+1}$ は $k(\mathbf{x}_{N+1},\mathbf{x}_{N+1})+\beta^{-1}$ である。従って、$y_{N+1}$ の予測値の平均 $\mu_{N+1}$ および分散 $\sigma_{N+1}^2$ は、条件付き確率 $p(y_{N+1}|\ \mathbf{X}_N,\mathbf{x}_{N+1},\ \mathbf{Y}_N)$の平均、分散より以下のように求まる。

$$\mu_{N+1} = \mathbf{k}_N^T \mathbf{C}_N^T \mathbf{Y}_N \tag{4.2.8}$$

$$\sigma_{N+1}^2 = c_{N+1} - \mathbf{k}_N^T \mathbf{C}_N^T \mathbf{k}_N \tag{4.2.9}$$

ガウス過程回帰の出力はカーネル関数 $k(\mathbf{x}_\mathrm{i},\mathbf{x}_j)$のパラメータや $\alpha$、$\beta$ といった超パラメータに依存している。得られたデータ $\{\mathbf{x}_i, y_i\}_1^N$ よりこれらの超パラメータを決定することがガウス過程回帰における学習の目的となる。超パラメータをまとめたベクトルを $\mathbf{\Phi}$ とすると、対数尤度は、

$$\ln p(\mathbf{Y}_N \mid \mathbf{X}_N, \mathbf{\Phi}) = -\frac{1}{2}\ln\left|\mathbf{C}_N\right| - \frac{1}{2}\mathbf{Y}_N^T \mathbf{C}_N^{-1} \mathbf{Y}_N - \frac{N}{2}\ln(2\pi) \tag{4.2.10}$$

超パラメータ $\mathbf{\Phi}$ は以下の最適化により求めることができる。

$$\mathbf{\Phi} = \arg\max_{\mathbf{\Phi}} p(\mathbf{Y}_N \mid \mathbf{X}_N, \mathbf{\Phi}) \tag{4.2.11}$$

## 4.3　ガウス過程回帰による自動採譜アルゴリズム

ピアノの和音演奏時においては各構成音の周波数成分が同時に発生するため、音響データには複数の演奏音の周波数成分が混在している。このような状況の音響データから、各音高のベロシティ値を推定する最も単純な方法は、音高ごとに音響データからベロシティ値を推定する回帰モデルを学習し、未知楽曲の音響データに対するベロシティ値を推定することである。すなわち、鍵盤数分の回帰モデルを学習し、MIDI ノートナンバーごとにベロシティ値を推定する。

各音高の回帰モデル出力においては押鍵の有無の区別は明確ではなく、また細かい変動も含んでいる。これらに対処するために、各音高の回帰モデル出力に対して平滑化処理、枝刈り処理といった後処理を行い、最終的な推定結果を得る。図 4.1 にガウス過程回帰による自動採譜アルゴリズムの概要を示す。

### 4.3.1　モデルの学習

図 4.2 はピアノ演奏の音響データ（(a)）と演奏データ（(b)）である。同図(c)は演奏データより C4 音（MIDI ノートナンバー60）にあたる成分を抜き出したも

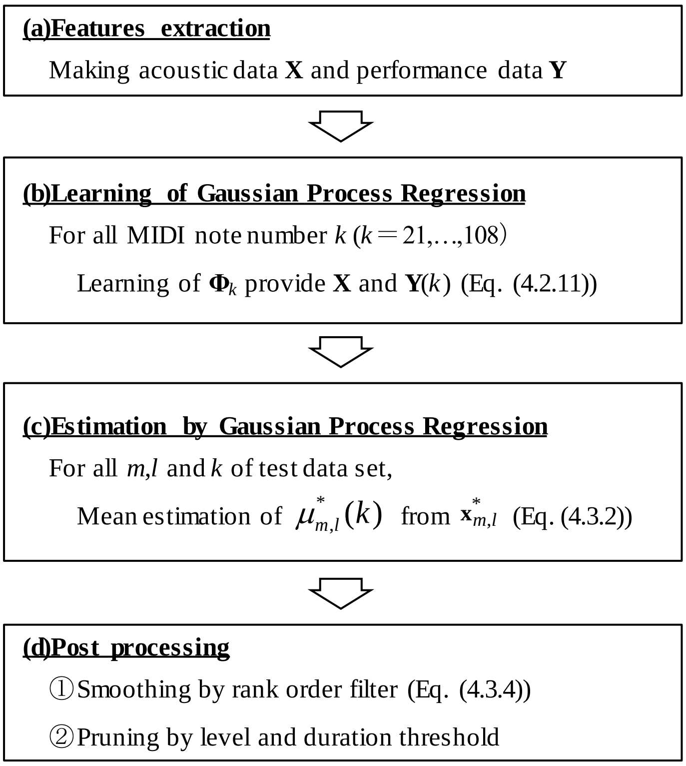
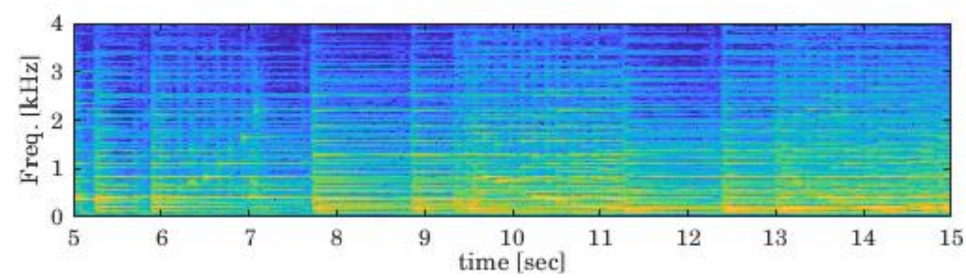


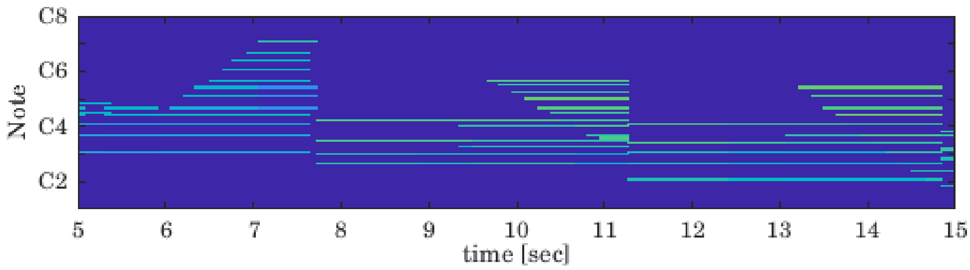

図 4.1　ガウス過程回帰による自動採譜アルゴリズム

のであり、C4 音のベロシティ値の変化が示されている。音響データには複数の演奏音の周波数成分が混在しているが、このような状況の音響データとともに対応する演奏データを与え、C4 音のベロシティ値を推定する回帰モデルを学習すれば、未知楽曲の音響データに含まれる C4 音のベロシティ値を推定することができる。ここで、学習用楽曲から無作為に抽出した音響データフレームを $\mathbf{X}=[\mathbf{x}_1,\ldots,\mathbf{x}_N]^T$, $\mathbf{x}_i\in\Re^D$ 、対応する箇所の演奏データフレームを $\mathbf{Y}=[\mathbf{y}_1,\ldots,\mathbf{y}_N]^T$, $\mathbf{y}_i\in\Re^Q$ とし、$\mathbf{Y}$ の MIDI ノートナンバー $k$（$k$=21,…108、ピアノの音域に対応）に対応する成分を $\mathbf{Y}(k)=[y_1(k),\ldots,y_N(k)]^T$, $y_i(k)\in\Re$ と表すと、学習データセット$\{\mathbf{X},\mathbf{Y}(k)\}$によって MIDI ノートナンバー$k$ に対応する回帰モデルを学習することができる。なお、本章ではカーネル関数として以下のものを使用する。

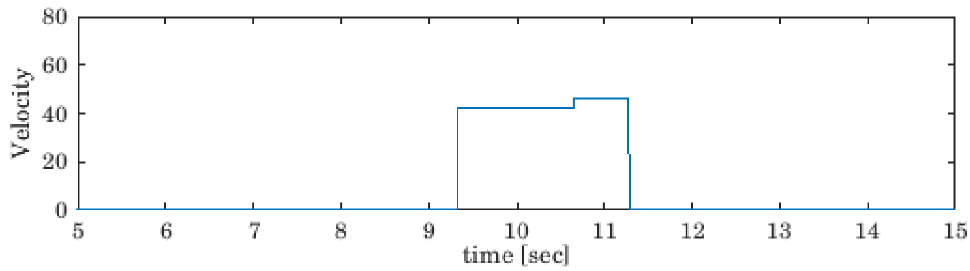
$$k(\mathbf{x}_i,\mathbf{x}_j)=\theta_1\exp\left(-\frac{\|\mathbf{x}_i-\mathbf{x}_j\|^2}{2\theta_2^2}\right) \tag{4.3.1}$$




(a) 音響データ




(b) 演奏データ




(c)C4 音のベロシティ値

図 4.2　音響データと演奏データの対応

### 4.3.2　各音高ベロシティ値の推定

学習済みのモデルを用い、未知のテスト用楽曲の演奏データを推定する。テスト用楽曲 $m$（$m$=1,…,$M$）における第 $l$ フレーム（$l$=1,…,$L$）の音響データを $\mathbf{x}^*_{m,l} \in \Re^D$、対応するフレームにおける MIDI ノートナンバー$k$ のベロシティ値を $y^*_{m,l}(k)$とすると、$y^*_{m,l}(k)$の平均値 $\mu^*_{m,l}(k)$、分散 $\sigma^{*2}_{m,l}(k)$は(4.2.8)式、(4.2.9)式によりそれぞれ以下のように求めることができる。

$$\mu^*_{m,l}(k) = \mathbf{k}_N^{*T}\mathbf{C}_N^T\mathbf{Y}_N \tag{4.3.2}$$

$$\sigma^{*2}_{m,l}(k) = c^{*}_{m.l} - \mathbf{k}^{*T}_{N}\mathbf{C}^{T}_{N}\mathbf{k}^{*}_{N} \tag{4.3.3}$$

ただし $\mathbf{k}^{*T}_{m,l}$ は $i$ 要素が $k(\mathbf{x}_i, \mathbf{x}^{*}_{m,l})$ （$i$=1,…,$N$,$m$=1,…,$M$, $l$=1,…,$L$）のベクトル、$c^{*}_{m,l}$ は $k(\mathbf{x}^{*}_{m,l}, \mathbf{x}^{*}_{m,l}) + \beta^{-1}$ である。(4.3.2)式、(4.3.3)式を該当するフレーム、および MIDI ノートナンバーに適用してベロシティ値の推定を行い、演奏データの推定を行う。

図 4.3 に未知楽曲に対する C4 音ベロシティ値の推定例を示す。複数の演奏音が混在する音響データから、C4 音のみのベロシティ値が推定できていることが分かる。

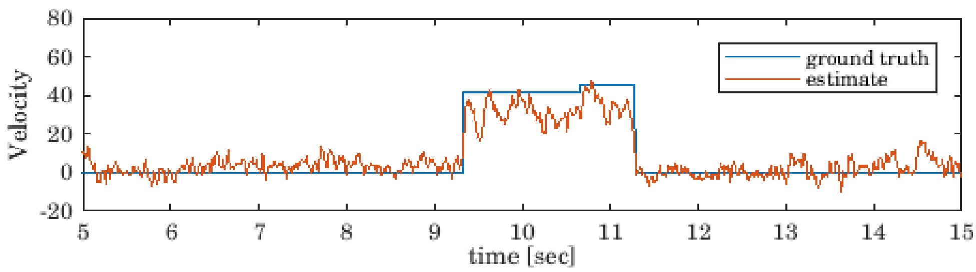



図 4.3　未知楽曲に対する C4 音ベロシティ値の推定例

### 4.3.3　推定結果の後処理

図 4.3 に示すように、演奏データ推定値は押鍵の有無の区別は明確ではなく、また細かい変動も含んでいる。自動採譜には、推定対象の楽曲における演奏音を音符イベントとして推定するノートトラッキングと、各フレームにおいて演奏されている音高を推定するフレームレベルの推定という考え方があるが（Benetos et al.（2013））、いずれの場合においても、押鍵の有無についての閾値の決定や、細かい変動の除去といった後処理が必要となる。最も簡単な方法は、演奏データ推定値のレベルや継続時間について閾値を設けることであるが（Benetos et al.（2013））、メジアンフィルタのような非線形フィルタが使用されるケースもある（Cheng et al.（2015））。また、平滑化の結果を離散値として扱う場合には隠れマルコフモデルが用いられることもある（Benetos et al.（2013）, Benetos and Weyde（2013）, Cheng et al.（2015）, Poliner and Ellis（2007）, Ryynänen and Klapuri（2005））。

本研究では推定結果を連続値として平滑化することを考え、テスト用楽曲における各音高の推定値に非線形フィルタの一種である rank order filter（棟安・田口（1999））を適用する。rank order filter は実装が容易でありながら、信号成分の保存とインパルス性雑音の除去について優れた性質を持っている。

テスト用楽曲 $m$（$m$=1,...,$M$）における第 $l$ フレーム（$l$=1,･･･,$L$）の演奏データ推定値を $\mathbf{y}^{*}_{m,l} \in \Re^{Q}$、$\mathbf{y}^{*}_{m,l}$ の第 $k$ 成分を $y^{*(k)}_{m,l}$ ($k$=1,･･･,$R$)とすると、$y^{*(k)}_{m,l}$ に対する平滑値 $u^{(k)}_{m,l}$ は以下の rank order filter の出力として得られる。

$$u^{(k)}_{m,l} = s\text{-}th\ \ largest\ \ value\ \ of\ \ \{y^{*(k)}_{m,l-L_1},\ldots,y^{*(k)}_{m,l+L_2}\} \tag{4.3.4}$$

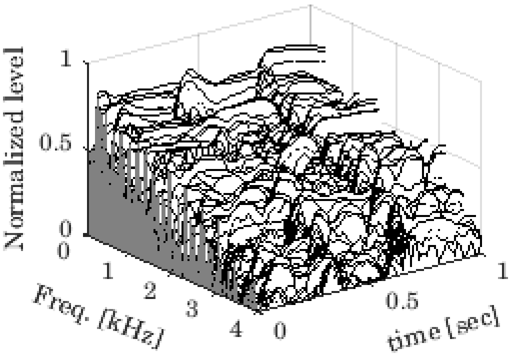
即ち、$\{y^{*(k)}_{m,l-L_1},\ldots,y^{*(k)}_{m,l+L_2}\}$ の中で $s$ 番目に大きな値が $u^{(k)}_{m,l}$ となる。$L_1$=$L_2$ かつ $s$=$L_1$+1=$L_2$+1 のとき、メジアンフィルタの出力と一致する。

更に $u^{(k)}_{m,l}$ のレベルが閾値 $h$ を下回る値を 0 で置き換え、また $h$ を連続して上回るフレーム数が閾値 $\tau$ より短い箇所についても 0 で置き換えて枝刈り処理を行う。これらの後処理の結果、レベルが $h$ 以上となった部分が押鍵している部分に対応する。図 4.4 に各段階の推定値の例を示す。後処理により、滑らかな推定結果が得られていることが分かる。

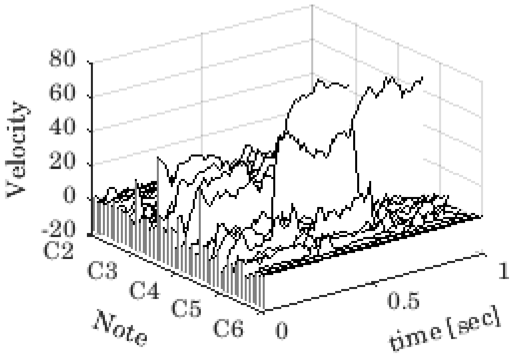
## 4.4　実験結果

学習用楽曲群から学習データを無作為に選出し、各フレームに対応する音響データ，演奏データを抽出して学習用の音響データセット **X**，演奏データセット **Y** をそれぞれ作成した。ガウス過程の学習および推定には MATLAB を使用した。

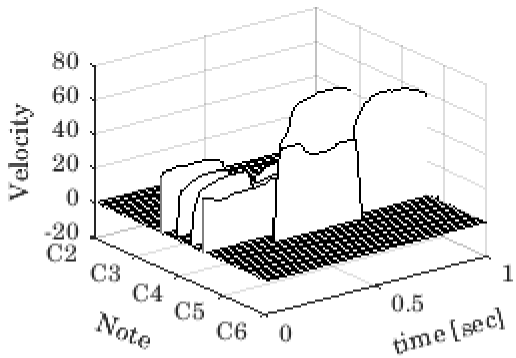
図 4.5 は後処理前のガウス過程回帰出力における打鍵時のベロシティ値正解値と推定値の対応を示したものである。学習データは、学習用楽曲より 5000 フレームを無作為に抽出して作成した。推定値は各音高における打鍵タイミング（オンセット）前後 5 フレーム中の最大値を取り出した。また各点における正解値を $y_n$、推定値を $\mu^{*}_{n}$（$n$=1,...,$N$，ただしここでは $N$ はオンセットの個数とする）とし、正解値と推定値の平均誤差平方根（Root mean squared error，以下 RMSE と表記）を以下の式により求めた。




(a)音響データ




(b)演奏データ推定値




(c)後処理後の演奏データ推定値

図 4.4　音響データと各段階の推定値

$$\mathrm{RMSE} = \sqrt{\frac{1}{N}\sum_{n=1}^{N}(\mu_n^* - y_n)^2} \tag{4.4.1}$$

図 4.5 より、ばらつきはあるが、強く打鍵した場合にはベロシティ推定値が大きくなる傾向が表れていることが分かる。従って、回帰の結果は、演奏時の打鍵の強さを反映したものであると考えられる。また、推定値が 0 前後となっているものがあるが、これらは後述するように低音域および高音域における推定精度の低下によるものと考えられる。

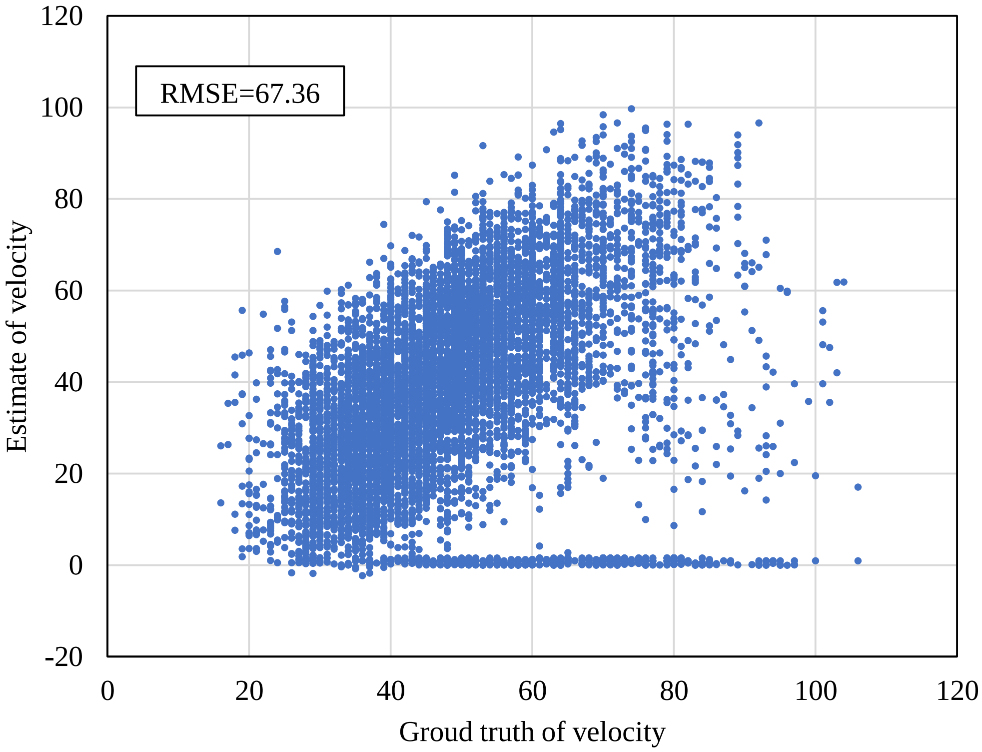



図 4.5　ガウス過程回帰における正解データと推定値

次に、回帰の結果得られた推定値を識別値として評価する。後処理の結果、値が 0 より大きくなったものを検出フレーム、0 となったものを未検出フレームとし、適合率（precision）を $P$、再現率（recall）を $R$、F 値（F-measure）を $F$ として、以下の式により評価を行った。

$$
\begin{aligned}
P &= c/e \\
R &= c/r \\
F &= 2PR/(P+R)
\end{aligned}
\tag{4.4.2}
$$

$c$ は正解フレーム数、$e$ は検出フレーム数、$r$ は正例フレーム数である。演奏データの推定はテスト用楽曲を用いて行い、後処理のパラメータは、それぞれのケースにおいて学習用楽曲に対する F 値が最も高くなるものをグリッドサーチにより決定した。

図 4.6 は学習データを 1000～5000 フレームの範囲で 1000 フレームずつ増加さ

せたときの推定結果の推移である。2000 フレームまでは F 値が大きく改善されているが，それ以降では飽和している。この結果から、本手法で必要とされる学習データは 2000 フレーム程度であり、それ以上学習データを増やしても大幅な推定精度の改善は見られないことが分かった。学習データ数の決定は、推定結果の改善幅と学習時間のトレードオフによりフレーム数を決定する必要がある。

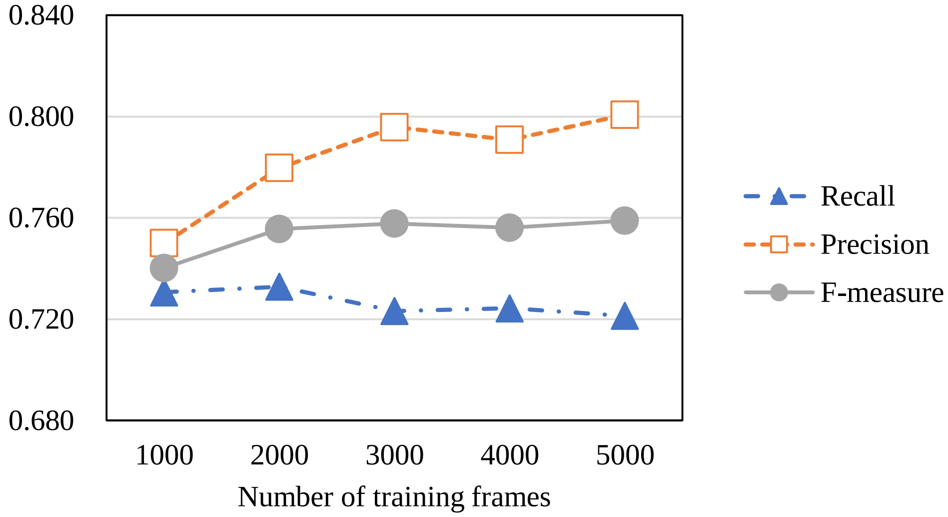



図 4.6　ガウス過程回帰自動採譜アルゴリズムの学習データ数に対する推定結果

図 4.7 は 5000 フレームの学習データを用いたときの、後処理の手法に対する推定結果を示している。比較した手法は、

① レベル閾値による枝刈り処理

② レベル閾値および継続フレーム数閾値による枝刈り処理

③ メジアンフィルタによる平滑化処理およびレベル閾値，継続フレーム数閾値による枝刈り処理

④ rank order filter による平滑化処理およびレベル閾値，継続フレーム数閾値による枝刈り処理

である。③のメジアンフィルタを用いた場合に最も高い推定精度が得られており、同手法の有効性が示された。

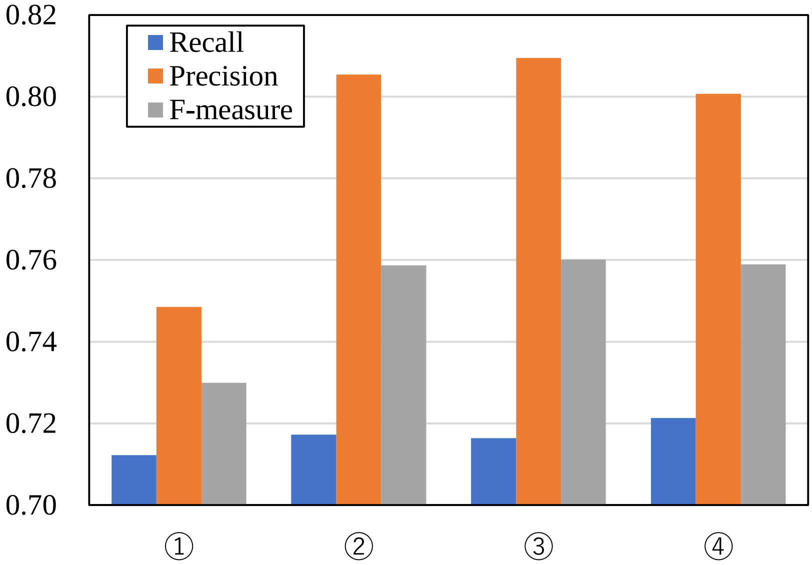


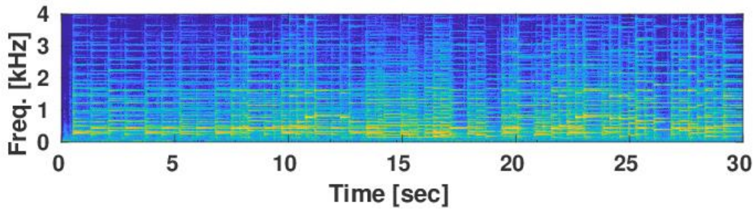

図 4.7　後処理手法とガウス過程回帰自動採譜アルゴリズムの推定結果

図 4.8 にテストデータの楽曲に対する推定例を示す。ガウス過程回帰により音響データ（振幅スペクトル，図 4.8(a)）から演奏データ（各音高のベロシティ値）を推定したものが図 4.8(b)、演奏データに後処理を施したものが図 4.8(c)である。演奏データ推定値に残っていた微小な変動が除去され、押鍵の有無がより明確になっている。

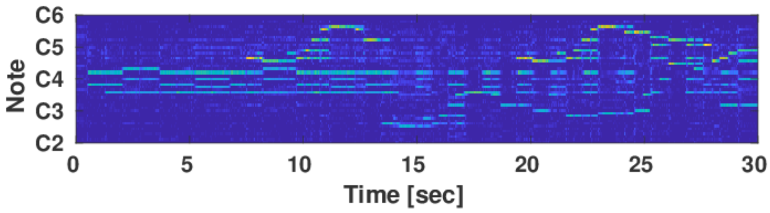
図 4.9 に音高別の推定精度を示す。全音域を通じての推定結果は、再現率 0.721、精度 0.801，F 値 0.760 であった。同図より、中音域では比較的高い推定精度が得られているが、低音域（C2 以下）および高音域（A#5 以上）での推定精度が低下していることが分かる。表 4.1 に本研究と同じテスト用楽曲を使用した先行研究における採譜結果とともに、本章で提案した手法による採譜結果を示す。

## 4.5　まとめと考察

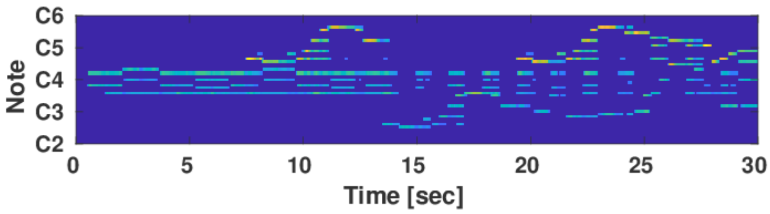
本章では、音響データから演奏データのベロシティ値を推定するモデルをガウス過程回帰により実現した。1 出力の回帰モデルを音高ごとに置くことで、各音高のベロシティ値を反映した推定値が得られることを確認した。さらに rank order filter による平滑化処理およびレベル閾値, 継続フレーム数閾値による枝刈り処理を行って後処理を施すことで、自動採譜アルゴリズムを構築した。その結果、全音域を通じて F 値 0.760 の推定結果を得ることができた。



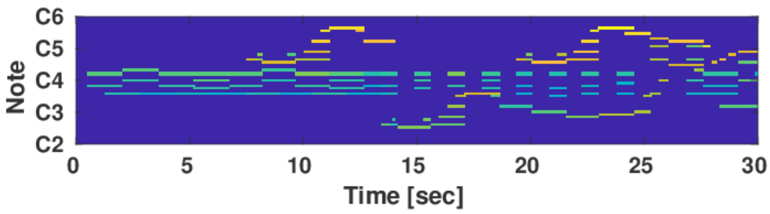

(a)音響データ




(b)演奏データ推定値

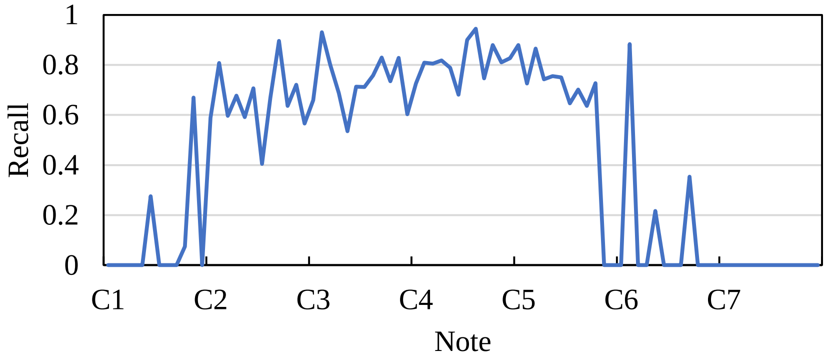



(c)後処理後の演奏データ推定値

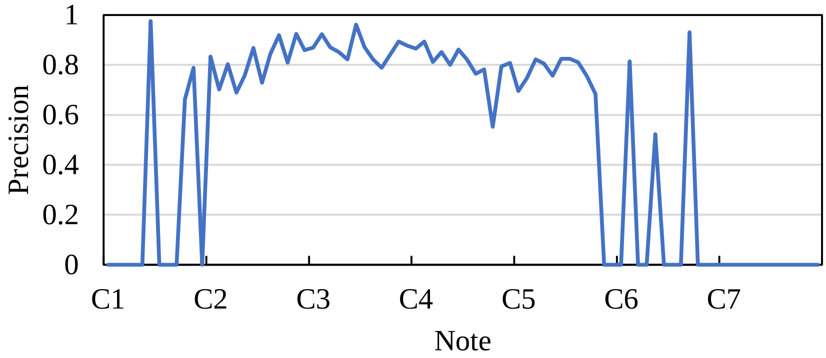



(d)正解データ

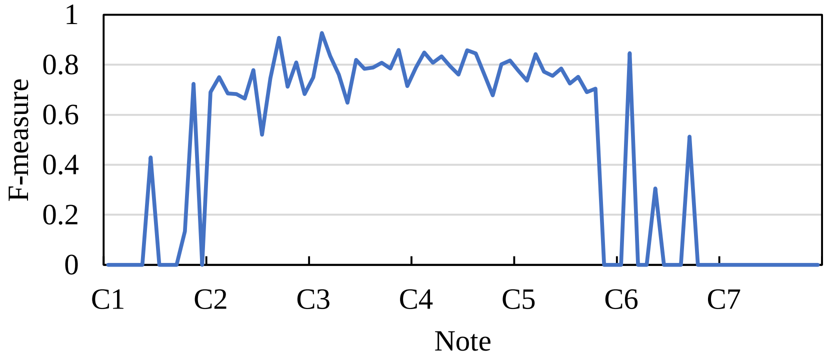
図 4.8　ガウス過程回帰自動採譜アルゴリズムのテスト用楽曲に対する推定例




(a)再現率




(b)適合率




(c)F 値

図 4.9　ガウス過程回帰自動採譜アルゴリズムの音高別推定精度

表 4.1　先行研究および本章での採譜結果（フレームレベル）

| System | Precision | Recall | F-measure |
|---|---|---|---|
| Benetos and Weyde (2013) | - | - | 0.680 |
| Vincent et al. (2010)* | 0.796 | 0.636 | 0.707 |
| O'Hanlon and Plumbley (2014) | 0.755 | 0.705 | 0.729 |
| Berg-Kirkpatrick et al. (2014) | 0.691 | 0.807 | 0.744 |
| Cheng et al. (2015) | 0.854 | 0.729 | 0.777 |
| Cheng et al. (2016) | - | - | **0.790** |
| Gaussian Process Regression | 0.716 | 0.810 | 0.760 |

* 原文ではテスト用楽曲の構成が本研究とは異なるため、Berg-Kirkpatrick et al. (2014)で評価された結果を記載。

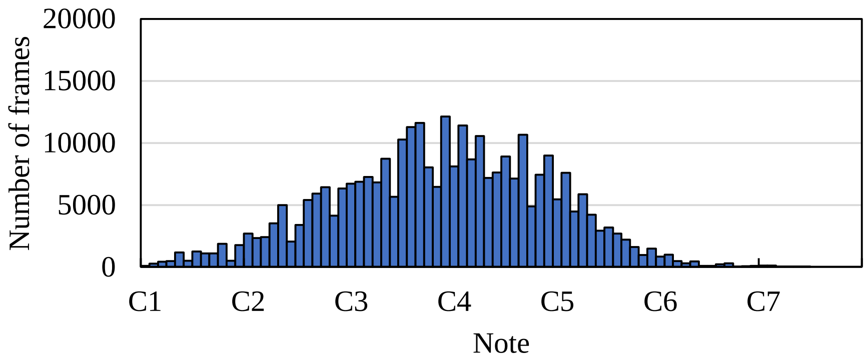
先行研究との比較では、今回比較した事例の中で最も高い推定精度を示した Cheng et al.（2016）および Cheng et al.（2015）には届かなかったものの、比較した事例の中では比較的高い推定精度を得ることができた。

本章冒頭で述べたように、本章の目的は打鍵，離鍵のタイミングだけでなく、打鍵の強さも推定することで、演奏の表情についての情報をも抽出しようとするものである。回帰モデルに基づいた自動採譜アルゴリズムを構築することでこの課題への対応を図ったが、図 4.5 に示した結果から打鍵の強さを反映した推定結果を得られることを示すことができた。ただし現状では推定結果のばらつきが大きく、このばらつきの縮小が今後の課題と考えられる。ばらつきが発生する原因として、演奏データであるベロシティ値は打鍵時から離鍵時まで一定値をとるのに対し､音響データである振幅スペクトルは打鍵の直後から減衰する信号であり、一定のベロシティ値に対して減衰途中の様々な大きさの振幅スペクトルが対応した形で学習が行われることが考えられる。この点はピアノのような減衰音では大きな問題となり、今後の課題である。

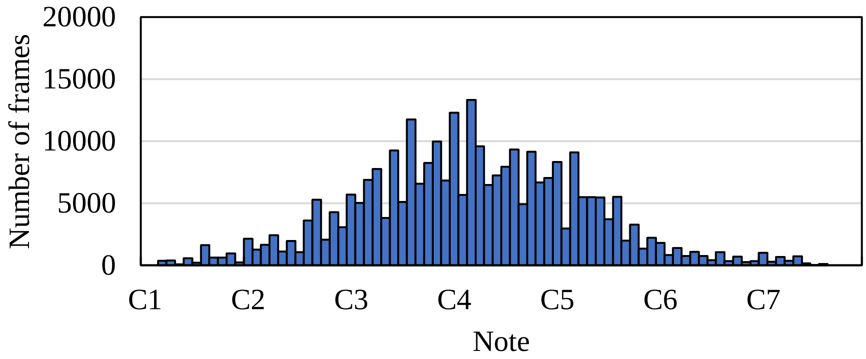
後処理についてはメジアンフィルタによる平滑化処理とレベル閾値，継続フレーム数閾値による枝刈り処理の組み合わせにより、大きな効果を得ることができ

た。メジアンフィルタは rank order filter の特別なケースと捉えることができるので、実装においては rank order filter による平滑化処理とレベル閾値，継続フレーム数閾値による枝刈り処理により実現することができると考えられる。

後処理後の全音域を通じた F 値は 0.760 であったが、図 4.9 に示した音高別の推定精度を見ると、C2 以下の低音域および A#5 以上の高音域で推定精度が低下している。図 4.10 に MAPS database の音高別フレーム数を示すが、推定精度が低下している音域では各音高の出現頻度が低いことが分かる。本章で提案した手法では、学習用楽曲中より無作為に学習フレームを抽出したため、学習データセットに含まれる低音域および高音域の音高が少なくなってしまい、推定精度の低下




(a)学習用楽曲




(b)テスト用楽曲

図 4.10　MAPS database における音高別フレーム数

を招いたものと考えられる。この点においては、学習フレームの抽出方法の見直しと、少ないデータからの安定した推定方法の確立の両面からの改善が求められる。

# 第 5 章　Shared Gaussian Process Latent Variable Model による自動採譜

## 5.1　はじめに

亀岡ら（亀岡・嵯峨山（2009））は自動採譜を、何らかの演奏プロトコル（五線譜上の音符や MIDI 信号など）に従って生成された音響信号から、そのプロトコルを解読する逆プロセスであると位置付けているが、多くの既存研究では演奏データを「原因」、音響データを「結果」と捉え、「結果」から「原因」を推定する逆問題として自動採譜の問題を扱ってきた。前章ではこの考え方に基づき、「結果」である音響データ（振幅スペクトル）を与え、「原因」である演奏データ（ベロシティ値）を推定する自動採譜アルゴリズムをガウス過程回帰により実現した。

一方、 MIDI インターフェイスなどの機器を介して、演奏者による演奏内容が演奏データとして得られる場合を想定すると、楽器の発音機構を介して得られる音響データと同様に、演奏データも「結果」、即ち観測変数とみなすことができる。音響データと演奏データはそれぞれ異なる形態をとるが、両者はともに演奏者による演奏により生じたデータであり、例えば演奏者の意図のような、直接観測できない共通の情報源を「原因」として生成されたものと考えることができる。図 5.1 にその構造を示す。未知楽曲については音響データのみが観測され、演奏データが欠損している状況に相当する。この場合、自動採譜は観測された音響データに基づき、欠損している演奏データを復元する問題と考えることができる。

先に述べた共通の情報源を、両データが共有する潜在変数として表現し、図 5.1 の構造をモデル化することができれば、与えられた未知楽曲の音響データに対応する情報源（潜在変数）の推定をまず行い、続いてその情報源に対応する演奏データを推定するといった手順で自動採譜を行うことが可能であると考えられる。音響データと演奏データが生成される構造の中で両データに共通する情報を見出し、この情報に基づいて音響データと演奏データの対応を考えることで、推定精度の向上が期待できる。このとき、潜在変数は必ずしも音楽的な表現である必要

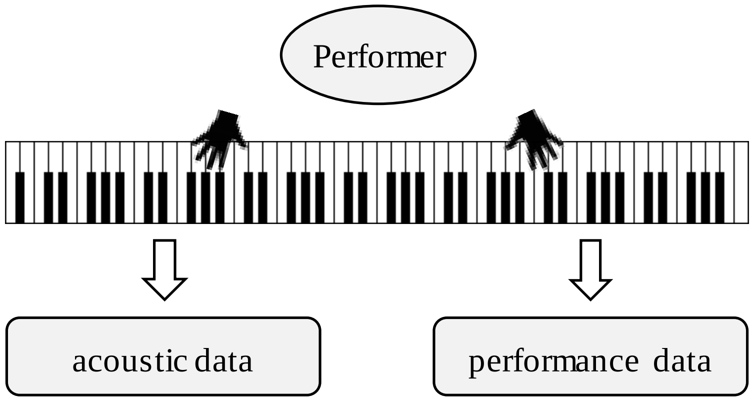



図 5.1. ピアノ演奏による音響データと演奏データの生成

はなく、音響データ、演奏データの双方を効率よく説明できる形式であればよい。統計的なモデル化では、しばしば本質的な情報を集約した低次元の潜在変数によって高次元の複雑な変数の挙動を説明することが行われるが、統計的な解析やパターン認識の諸問題でそれらのモデルの有効性が報告されている（Tipping and Bishop（1999））。

音響データ、演奏データをともに観測変数と捉え、未知楽曲の音響データに対する演奏データの推定を、欠損した観測変数の推定問題として扱う手法については、これまで調査されていない。本章では音響データと演奏データを直接関連付けるのではなく、共通の情報源から音響データおよび演奏データが生成される構造をモデル化し、この構造の中で自動採譜を行う方法を検討・提案する。更に、実験を通じてその有効性を検証する。

前述したように、音響データおよび演奏データを生成する共通の情報源は直接観測することはできず、またその表現形式は明らかではない。従ってモデルの構築に際しては、音響データおよび演奏データの挙動をよく説明できる表現形式、並びに両データとの関連を学習によって獲得することが求められる。このような目的のため、本章ではモデルの構築に Shared Gaussian Process Latent Variable Model（Shon et al.（2005），以下 SGPLVM と表記）を用いる。SGPLVM では形式の異なる複数の観測変数が低次元の潜在変数を共有する。音響データおよび演奏

データを観測変数として、両データに共通の情報源を潜在変数としてそれぞれ扱うことで、これらの変数を SGPLVM によって関連付けることができる。

次節以降では、SGPLVM の基礎となる Gaussian Process Latent Variable Model（Lawrence（2005），以下 GPLVM と表記）について説明した後に SGPLVM について説明し、更に SGPLVM を用いた自動採譜の方法（今村・松井 (2017)）について説明する。

### 5.2　Gaussian Process Latent Variable Model（GPLVM）

Lawrence らは高次元データの非線形な確率的次元削減手法である GPLVM を提案した（Lawrence（2005），図 5.2(a)）。GPLVM では観測変数 $\mathbf{Y}=[\mathbf{y}_1,\ldots,\mathbf{y}_N]^{\mathrm{T}}$，$\mathbf{y}_i\in\Re^Q$ が、低次元の潜在変数 $\mathbf{X}=[\mathbf{x}_1,\ldots,\mathbf{x}_N]^{\mathrm{T}}$，$\mathbf{x}_i\in\Re^D$（$D < Q$）より、以下のような確率過程によって生成されると考える。

$$\mathbf{y}_i = f(\mathbf{x}_i) + \boldsymbol{\varepsilon} \ \ , \ \ \boldsymbol{\varepsilon} \sim N(\mathbf{0}, \beta^{-1}\mathbf{I}) \tag{5.2.1}$$

$f$ は非線形マッピング関数、$\beta$ はガウスノイズの精度パラメータである。

上式の非線形マッピング $f$ に事前分布を与えて周辺化すると、以下のように周辺尤度が求まる。

$$p(\mathbf{Y} \mid \mathbf{X}, \boldsymbol{\Phi}) = \frac{1}{\sqrt{(2\pi)^{N \cdot Q} |\mathbf{K}|^{Q}}} \exp\left( -\frac{1}{2} tr(\mathbf{K}^{-1}\mathbf{Y}\mathbf{Y}^{T}) \right) \tag{5.2.3}$$

$\boldsymbol{\Phi}$ は超パラメータ、$\mathbf{K}$ はカーネル関数 $k(\mathbf{x}_i,\mathbf{x}_j)$が$(ij)$要素のカーネル行列である。観測変数 $\mathbf{Y}$ に対する未知の潜在変数 $\mathbf{X}$ および超パラメータ $\boldsymbol{\Phi}$ の推定値は周辺尤度を最大化することで求めることができる。

$$\{\hat{\mathbf{X}}, \hat{\boldsymbol{\Phi}}\} = \mathop{\arg\max}_{\mathbf{X}, \boldsymbol{\Phi}} \ p(\mathbf{Y} \mid \mathbf{X}, \boldsymbol{\Phi}) \tag{5.2.3}$$

GPLVM では、潜在変数空間から観測変数空間へのマッピングは滑らかなものとなり、潜在変数空間で距離の近い点は観測変数空間でも近い距離となるが、逆方向のマッピングは滑らかなものとはならず､必ずしも距離関係は保障されない。Lawrence らは観測変数空間から潜在変数空間へのマッピングにおいて距離関係

を保存するマッピングを、パラメトリックな関数 $\mathbf{x}_i=g(\mathbf{y}_i,\mathbf{W})$によって実現する back constraint （Lawrence and Quiñonero-Candela（2006））を提案した。$\mathbf{W}$ は back constraint のパラメータである。この場合、以下の目的関数の最大化によって学習を行う。

$$\{\hat{\mathbf{W}},\hat{\mathbf{\Phi}}\}=\arg\max_{\mathbf{W},\mathbf{\Phi}}\ p(\mathbf{Y}\,|\,\mathbf{W},\mathbf{\Phi}) \tag{5.2.4}$$

なお、GPLVM の学習については、スパース近似（Quinonero-Candela and Rasmussen（2005））に基づく高速な学習アルゴリズムが提案されている（Lawrence（2007））。

### 5.3　Shared Gaussian Process Latent Variable Model（SGPLVM）

GPLVM では、潜在変数に対応する観測変数は一つだけであったが、Shon らはこれを拡張し、複数の観測変数が一つの潜在変数を共有する SGPLVM を提案した（Shon et al.（2005））。図 5.2(b)に SGPLVM のグラフィカルモデルを示す。

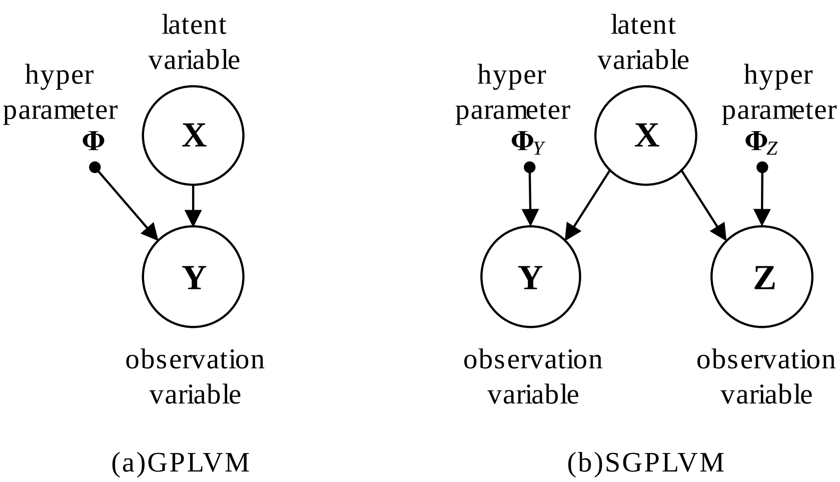



図 5.2　GPLVM，SGPLVM のグラフィカルモデル

SGPLVM では学習データセット $\mathbf{Y}=[\mathbf{y}_1,\ldots,\mathbf{y}_N]^{\mathrm{T}}$, $\mathbf{y}_i\in\Re^Q$ および $\mathbf{Z}=[\mathbf{z}_1,\ldots,\mathbf{z}_N]^{\mathrm{T}}$, $\mathbf{z}_i\in\Re^R$ を与え、これらに対応する潜在変数 $\mathbf{X}=[\mathbf{x}_1,\ldots,\mathbf{x}_N]^{\mathrm{T}}$，$\mathbf{x}_i\in\Re^D$（$D<Q$，$D<R$）および超パラメータ（hyper parameter）$\mathbf{\Phi}_Y$、$\mathbf{\Phi}_Z$ を以下の最適化により求める。

$$\{\hat{\mathbf{X}}, \hat{\mathbf{\Phi}}_Y, \hat{\mathbf{\Phi}}_Z\} = \underset{\mathbf{X}, \mathbf{\Phi}_Y, \mathbf{\Phi}_Z}{\arg\max}\ p(\mathbf{Y}, \mathbf{Z} \mid \mathbf{X}, \mathbf{\Phi}_Y, \mathbf{\Phi}_Z)$$

$$= \underset{\mathbf{X}, \mathbf{\Phi}_Y, \mathbf{\Phi}_Z}{\arg\max}\ p(\mathbf{Y} \mid \mathbf{X}, \mathbf{\Phi}_Y)\, p(\mathbf{Z} \mid \mathbf{X}, \mathbf{\Phi}_Z) \tag{5.3.1}$$

ここで、

$$p(\mathbf{Y} \mid \mathbf{X}, \mathbf{\Phi}_Y) = \frac{1}{\sqrt{(2\pi)^{N \cdot Q} |\mathbf{K}_Y|^Q}} \exp\left(-\frac{1}{2} tr(\mathbf{K}_Y^{-1} \mathbf{Y}\mathbf{Y}^T)\right) \tag{5.3.2}$$

$$p(\mathbf{Z} \mid \mathbf{X}, \mathbf{\Phi}_Z) = \frac{1}{\sqrt{(2\pi)^{N \cdot R} |\mathbf{K}_Z|^R}} \exp\left(-\frac{1}{2} tr(\mathbf{K}_Z^{-1} \mathbf{Z}\mathbf{Z}^T)\right) \tag{5.3.3}$$

であり、$\mathbf{K}_Y$、$\mathbf{K}_Z$ はそれぞれカーネル関数 $k_Y(\mathbf{x}_i,\mathbf{x}_j)$、$k_Z(\mathbf{x}_i,\mathbf{x}_j)$を$(ij)$要素とするカーネル行列である。

推定時には未知の観測変数 $\mathbf{y}^* \in \Re^Q$ のみを与え、これに対応する潜在変数 $\mathbf{x}^* \in \Re^D$ をまず推定し、続いて $\mathbf{x}^*$に対応する $\mathbf{z}^* \in \Re^R$ を推定するという手順をとる。Ek らは、back constraint によって $\mathbf{y}^*$から $\mathbf{x}^*$を点推定し、 続いて $\mathbf{x}^*$に対応する $\mathbf{z}^*$の平均ベクトルを求めている（Ek et al.（2007）, Ek（2009））。

観測変数 $\mathbf{Y}$ から潜在変数 $\mathbf{X}$ へのマッピングに back constraint を用いる際、学習は以下の目的関数の最大化によって行う。

$$\{\hat{\mathbf{W}}, \hat{\mathbf{\Phi}}_Y, \hat{\mathbf{\Phi}}_Z\} = \underset{\mathbf{W}, \mathbf{\Phi}_Y, \mathbf{\Phi}_Z}{\arg\max}\ p(\mathbf{Y} \mid \mathbf{W}, \mathbf{\Phi}_Y)\, p(\mathbf{Z} \mid \mathbf{W}, \mathbf{\Phi}_Z) \tag{5.3.4}$$

$\mathbf{W}$ は観測変数 $\mathbf{Y}$ から潜在変数 $\mathbf{X}$ へのマッピング関数 $\mathbf{x}_i = g(\mathbf{y}_i, \mathbf{W})$のパラメータである。

## 5.4 SGPLVM による自動採譜アルゴリズム

本節では、SGPLVM による自動採譜アルゴリズムを説明する。まず観測変数である音響データと演奏データが同一の潜在変数を共有する構造を、図 5.2(b)に示

した SGPLVM によってモデル化する。未知の音響データが与えられた際には、まず音響データに対応する潜在変数を推定し、更に潜在変数に対応する演奏データを推定するといった手順で自動採譜を行う。これ以降、潜在変数、音響データ、演奏データを図 5.2(b)の **X**、**Y**、**Z** にそれぞれ対応させて議論を進める。

図 5.3 に自動採譜の処理フローを示す。各処理の詳細については次項以降で説明する。

**(a)Features extraction**

Making acoustic data **Y** and performance data **Z**

⇩

**(b)SGPLVM learning**

Learning of **W**, $\mathbf{\Phi}_Y$, $\mathbf{\Phi}_Z$ provide **Y** and **Z** (Eq. (5.3.4))

⇩

**(c)Estimation by SGPLVM**

For all *m* and *l*,

①Point estimation of $\mathbf{x}^*_{m,l}$ from $\mathbf{y}^*_{m,l}$ by back constraint (Eq. (5.4.6))

②Mean estimation of $\mathbf{z}^*_{m,l}$ from $\mathbf{x}^*_{m,l}$ (Eq. (5.4.8))

⇩

**(d)Post processing**

①Smoothing by rank order filter (Eq. (5.4.9))

②Pruning by level and duration threshold

図 5.3 自動採譜の処理フロー

### 5.4.1 モデルの学習

まず同一の学習用楽曲の同一時点から音響データ **Y** と演奏データ **Z** を無作為に *N* 点抽出し、学習用データセットを作成する。このとき、音響データ **Y**、演奏データ **Z** ともに、連続する *r* フレームを連結して特徴量を作成する（図 5.4)。

続いて、抽出した学習データにより SGPLVM の学習を行う。音響データ **Y** から潜在変数 **X** へのマッピングについては Ek らと同様に back constraint を用いる

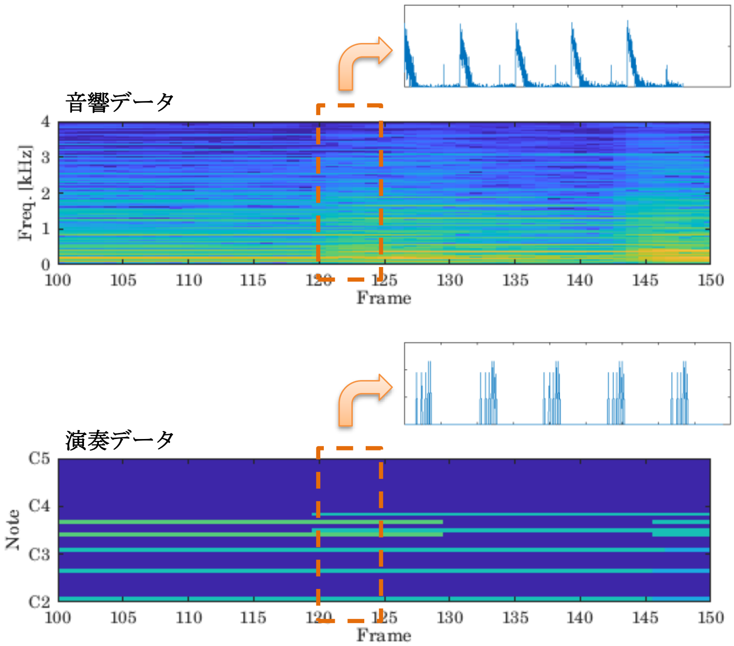



図 5.4　音響データ，演奏データの抽出

ため（Ek et al.（2007），Ek（2009））、学習は(5.3.4)式の最大化によって行う。(5.3.2)式、(5.3.3)式のカーネル関数 $k_Y(\mathbf{x}_i,\mathbf{x}_j)$、$k_Z(\mathbf{x}_i,\mathbf{x}_j)$は以下のものを使用する。

$$k_Y(\mathbf{x}_i,\mathbf{x}_j)=\theta_{Y1}\exp\left(-\frac{\|\mathbf{x}_i-\mathbf{x}_j\|^2}{2\theta_{Y2}^2}\right)+\theta_{Y3}+\beta_Y\delta(i,j) \tag{5.4.1}$$

$$k_Z(\mathbf{x}_i,\mathbf{x}_j)=\theta_{Z1}\exp\left(-\frac{\|\mathbf{x}_i-\mathbf{x}_j\|^2}{2\theta_{Z2}^2}\right)+\theta_{Z3}+\beta_Z\delta(i,j) \tag{5.4.2}$$

$\delta(i,j)$はクロネッカーのデルタであり、$\{\theta_{Y1},\theta_{Y2},\theta_{Y3},\beta_Y\}$および$\{\theta_{Z1},\theta_{Z2},\theta_{Z3},\beta_Z\}$は、それぞれ $k_Y(\mathbf{x}_i,\mathbf{x}_j)$、$k_Z(\mathbf{x}_i,\mathbf{x}_j)$のパラメータである。

また、back constraint による $\mathbf{y}_i$ から $\mathbf{x}_i$ へのマッピング $\mathbf{x}_i=g(\mathbf{y}_i,\mathbf{W})$は以下のカーネル回帰によって行う。

$$x_i^{(d)}=\sum_{j=1}^{N}w_j^{(d)}k_{bc}(\mathbf{y}_i,\mathbf{y}_j) \tag{5.4.3}$$

$x_i^{(d)}$は $\mathbf{x}_i$ の第 $d$ 要素であり、$w_j^{(d)}$は $\mathbf{W}$ の$(d,j)$要素である。カーネル関数 $k_{bc}(\mathbf{y}_i,\mathbf{y}_j)$

は以下の RBF カーネルを用いた。

$$k_{bc}(\mathbf{y}_i, \mathbf{y}_j) = \exp\left( -\frac{\left\| \mathbf{y}_i - \mathbf{y}_j \right\|^2}{2\theta_{bc}^2} \right) \tag{5.4.4}$$

$\theta_{bc}$ は $k_{bc}(\mathbf{y}_i, \mathbf{y}_j)$のパラメータである。

(5.4.1)式～(5.4.4)式より、学習により求めるべきモデルの超パラメータは以下のものとなる。

$$\begin{aligned} \mathbf{W} &= \{w_1^1, \ldots, w_N^D, \theta_{bc}\} \\ \mathbf{\Phi}_Y &= \{\theta_{Y1}, \theta_{Y2}, \theta_{Y3}, \beta_Y\} \\ \mathbf{\Phi}_Z &= \{\theta_{Z1}, \theta_{Z2}, \theta_{Z3}, \beta_Z\} \end{aligned} \tag{5.4.5}$$

### 5.4.2　各音高ベロシティ値の推定

未知のテスト用楽曲の演奏データを推定する。テスト用楽曲 $m$（$m$=1,…,$M$）における第 $l$ フレーム（$l$=1,…,$L$）の音響データを $\mathbf{y}_{m,l}^* \in \Re^Q$ とすると、$\mathbf{y}_{m,l}^*$ に対応する潜在変数 $\mathbf{x}_{m,l}^* \in \Re^D$ は back constraint により以下のように求めることができる。

$$x_{m,l}^{*(d)} = \sum_{i=1}^{N} w_i^{(d)} k_{bc}(\mathbf{y}_{m,l}^*, \mathbf{y}_i) \tag{5.4.6}$$

$x_{m,l}^{*(d)}$ は $\mathbf{x}_{m,l}^*$ の第 $d$ 要素である。$\mathbf{x}_{m,l}^*$ に対する演奏データ $\mathbf{z}_{m,l}^* \in \Re^R$ は以下の最大化により求めることができるが（Ek et al.（2007））、

$$\mathbf{z}_{m,l}^* = \arg\max_{\mathbf{z}} \; p(\mathbf{z} \mid \mathbf{x}_{m,l}^*, \mathbf{X}, \mathbf{\Phi}_Z) \tag{5.4.7}$$

潜在変数 $\mathbf{X}$ から演奏データ $\mathbf{Z}$ へのマッピングはガウス過程（Rasmussen and Williams（2006））であるので、$\mathbf{z}_{m,l}^*$ の平均の推定値は以下の式で与えられる（Ek（2009））。

$$(\mathbf{z}_{m,l}^*)^T = \mathbf{k}_Z^T \mathbf{K}_Z^{-1} \mathbf{Z} \tag{5.4.8}$$

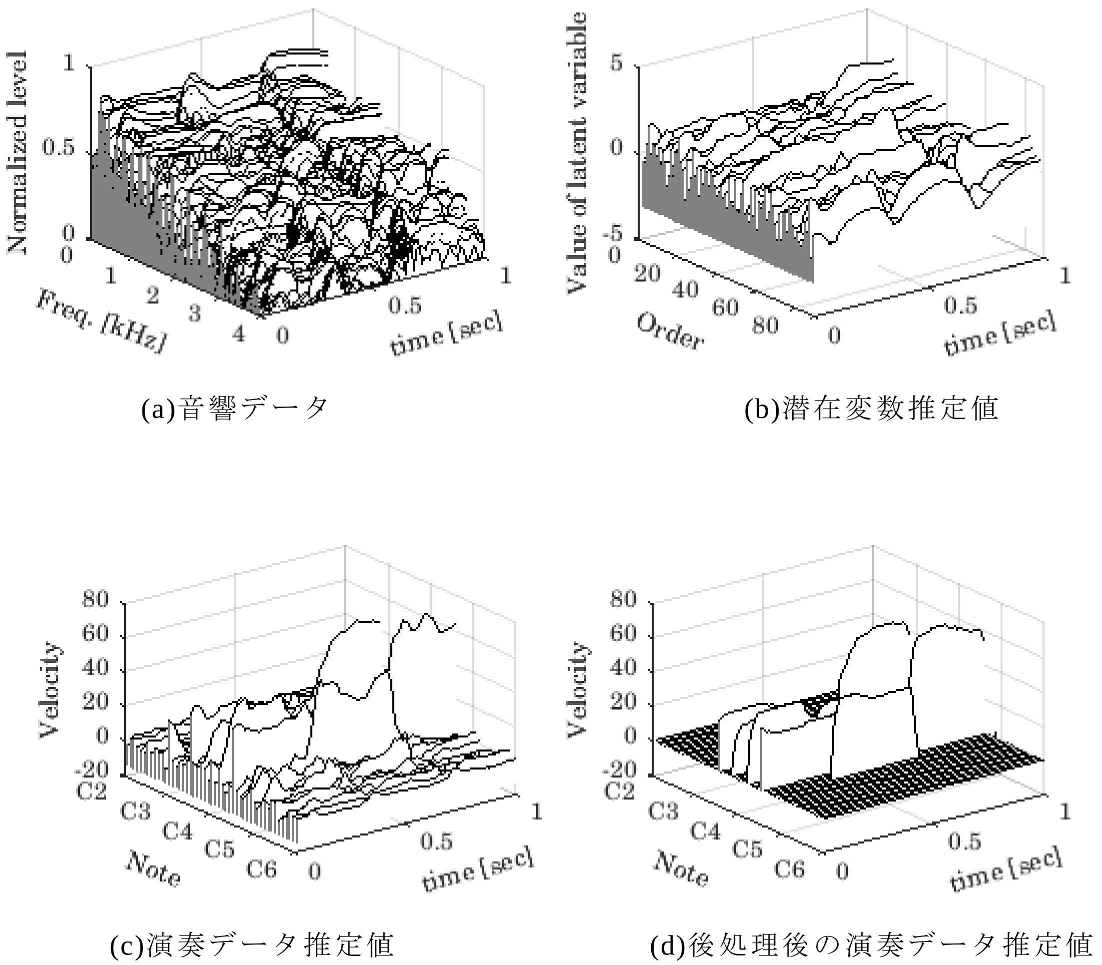
ここで $\mathbf{k}_Z = [k_z(\mathbf{x}_{m,l}^*, \mathbf{x}_1), \ldots, k_z(\mathbf{x}_{m,l}^*, \mathbf{x}_N)]^T$ である。

### 5.4.3　推定結果の後処理

前章のガウス過程回帰によるベロシティ値の推定値と同様、SGPLVM によるベロシティ推定値 $\mathbf{z}_{m,l}^{*}$ も押鍵の有無の区別は明確ではなく、細かい変動も含んでいる。従って $\mathbf{z}_{m,l}^{*}$ に対しても前章と同様の後処理を適用する。$\mathbf{z}_{m,l}^{*}$ の第 $k$ 成分を $z_{m,l}^{*(k)}$ $(k=1,\ldots,R)$ とすると、$z_{m,l}^{*(k)}$ に対する平滑値 $u_{m,l}^{(k)}$ は以下の rank order filter の出力として得られる。

$$u_{m,l}^{(k)} = s\text{-}th\ \ largest\ \ value\ \ of\ \ \{y_{m,l-L_1}^{*(k)},\ldots,y_{m,l+L_2}^{*(k)}\} \tag{5.4.9}$$

図 5.5 に各段階の推定値の例を示す。




(a)音響データ　(b)潜在変数推定値

(c)演奏データ推定値　(d)後処理後の演奏データ推定値

図 5.5 音響データと各段階の推定値

## 5.5　実験結果

学習データは前章で使用した学習データと同一のものを使用した。ただし、音響データ、演奏データともに、該当フレームとその直前 4 フレームを連結して長い特徴ベクトルを作成した。即ち各時刻に対し、4 期前から現在までの音響データ、演奏データそれぞれ 5 フレーム分が対応している。なお、連結させるフレーム数は実験的に求めた。

SGPLVM の学習および推定には Lawrence による MATLAB コード（Lawrence : Shared GP-LVM Software）をサブルーチンとして使用し、実験用プログラムを作成した。Lawrence の MATLAB コードはスパース近似（Quinonero-Candela and Rasmussen（2005））に基づいて学習の高速化を図っている（Lawrence（2007））。表 5.1 に SGPLVM の設定内容について記す。

表 5.1　SGPLVM の設定

| | |
|---|---|
| Number of inducing variables for sparse approximation（Lawrence (2007)） | 200 |
| Parameter $\theta_{bc}$ of back constraint kernel function (Eq.(5.4.4)) | 0.05 |
| Iteration of learning | 1000 |

図 5.6 は後処理前の SGPLVM 出力における打鍵時のベロシティ値正解値と推定値の対応を示したものである。学習データは、学習用楽曲より 5000 フレームを無作為に抽出して作成した。推定値は各音高における打鍵タイミング（オンセット）前後 5 フレーム中の最大値を取り出した。RMSE はガウス過程回帰の場合と同様に(4.4.1)式より求めた。

図 5.6 より、回帰の結果は演奏時の打鍵の強さを反映したものであることが分かる。推定値が 0 前後となっているものについてはガウス過程回帰の場合よりも減少しているが、これらは後述するように低音域および高音域における推定精度

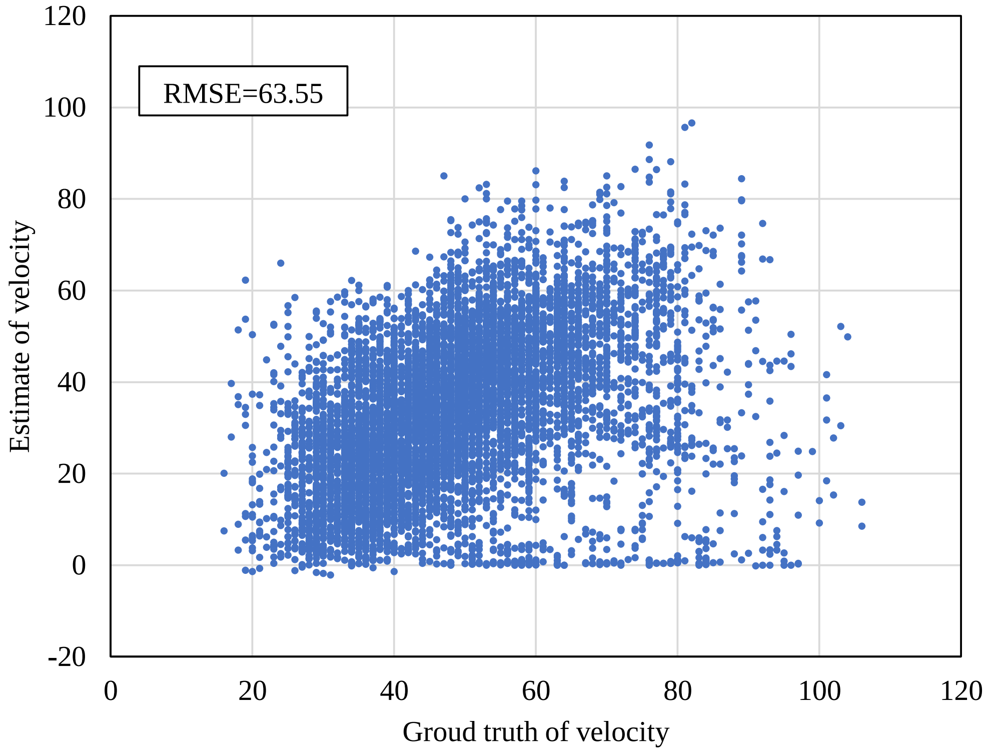



図 5.6　SGPLVM における正解データと推定値

が改善されたためと考えられる。これに伴い、全体の RMSE も 63.55 とガウス過程回帰の場合よりも改善されている。

また識別としての推定結果の評価は前章と同様に、精度（precision）$P$、再現率（recall）$R$、F 値（F-measure）$F$ により行った。まず潜在変数の次数、学習データのフレーム数、後処理手法の決定のための実験を行った。演奏データの推定はテスト用楽曲を用いて行い、後処理のパラメータは、それぞれのケースにおいて学習用楽曲に対する F 値が最も高くなるものをグリッドサーチにより決定した。

図 5.7 は、1000 フレームの学習データを用い、潜在変数の次数を 20 次、50 次、100 次、150 次と変化させて学習したときの推定結果の推移である。また、鍵盤数と同数の 88 次についても実験を行った。図に示すように、88 次としたときに最も高い推定精度が得られた。100 次、150 次ではモデルが収束せず、意味のある推定結果が得られなかった。

図 5.8 は潜在変数の次数を 88 次に固定し、学習データを 1000～5000 フレーム

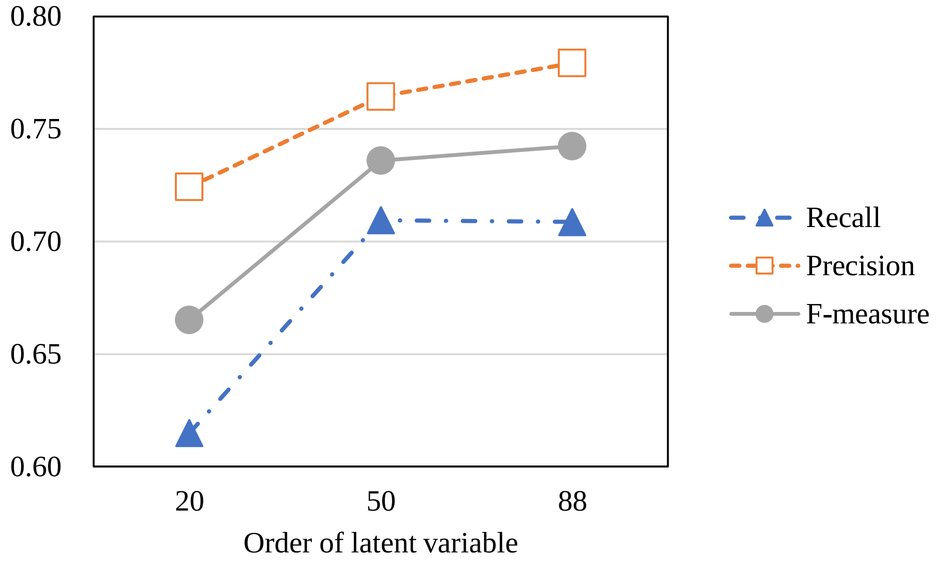



図 5.7　潜在変数次数に対する推定結果

の範囲で 1000 フレームずつ増加させたときの推定結果の推移である。3000 フレームまでは推定結果が大きく改善されているが、それ以降では改善幅は小さくなっている。この結果から、提案法では少なくとも 3000 フレームの学習データが必要であり、それ以上学習データを増やす場合は、推定結果の改善幅と学習時間のトレードオフによりフレーム数を決定する必要があると考えられる。

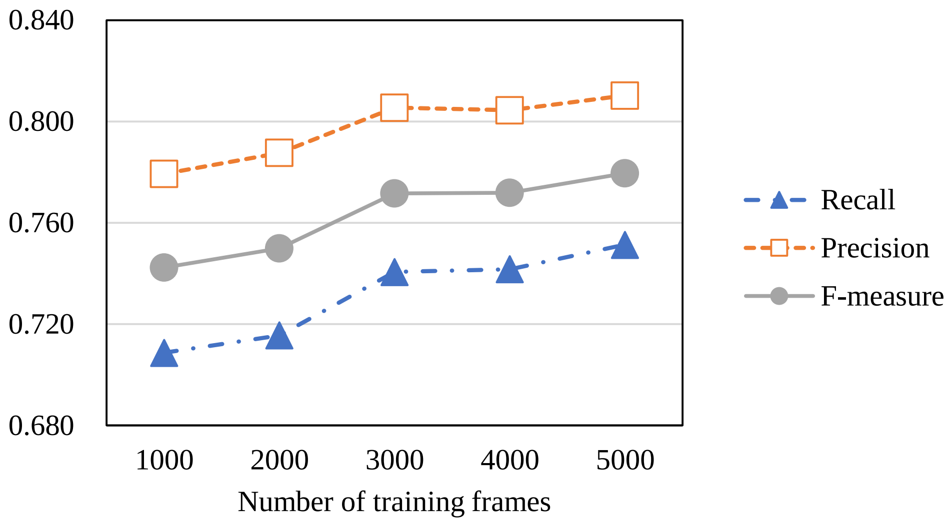



図 5.8　学習フレーム数に対する推定結果

図 5.9 は潜在変数の次数を 88 次とし、5000 フレームの学習データを用いたときの、後処理の手法に対する推定結果を示している。比較した手法は前章と同様、

① レベル閾値による枝刈り処理

② レベル閾値および継続フレーム数閾値による枝刈り処理

③ メジアンフィルタによる平滑化処理およびレベル閾値、継続フレーム数閾値による枝刈り処理

④ rank order filter による平滑化処理およびレベル閾値、継続フレーム数閾値による枝刈り処理

である。④の手法を用いた場合に最も高い推定精度が得られ、rank order filter の有効性が示された。

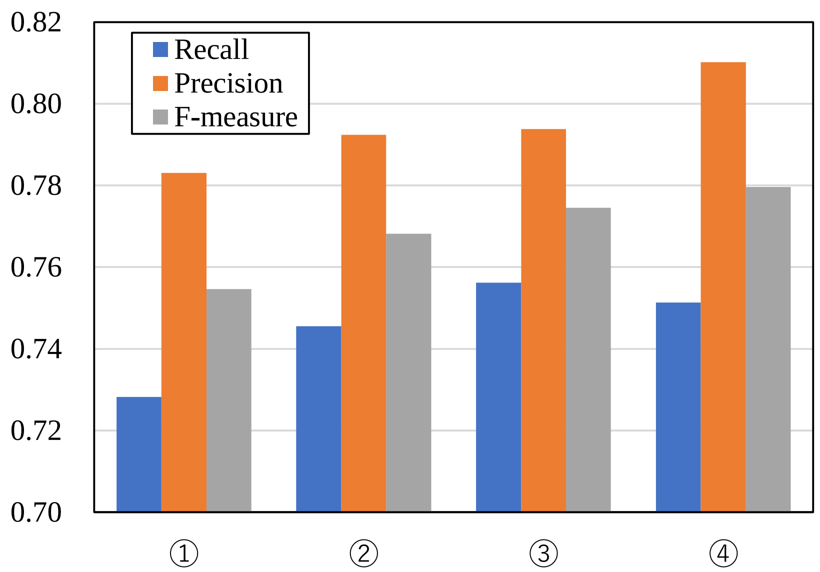



図 5.9 後処理手法と推定結果

以上の結果から、既出論文との比較においては、潜在変数の次数を 88 次とし、5000 フレームの学習データを用いて SGPLVM の学習を行ったモデルを使用した。また、後処理として rank order filter による平滑化処理およびレベル閾値、継続フレーム数閾値による枝刈り処理を行った。このとき、後処理のパラメータは表 5.2 に示すものを採用した。

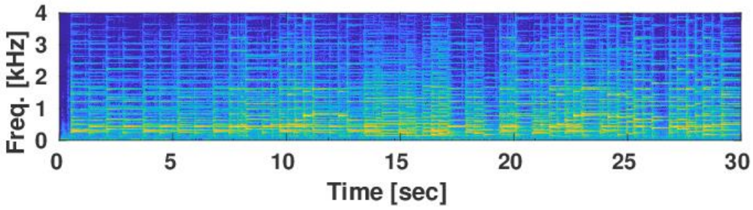
表 5.2 後処理のパラメータ

| | |
|---|---|
| Start point $L_1$ of rank order filter (Eq.(5.4.9)) | 5 |
| End point $L_2$ of rank order filter (Eq.(5.4.9)) | 11 |
| Order $s$ of rank order filter (Eq.(5.4.9)) | 9 |
| Level threshold $h$ of pruning | 16 |
| Duration threshold $\tau$ of pruning | 9 |

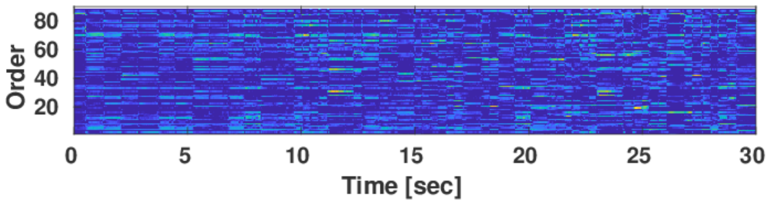
図 5.10 にテストデータの楽曲に対する推定例を示す。潜在変数（図 5.10(b)）は(5.4.6)式の back constraint により音響データ（図 5.10(a)）から求めた。潜在変数と音響データ、演奏データとの対応関係は明らかではないが、潜在変数の挙動がオンセット、オフセット等の挙動と連動していることが分かる。演奏データ推定値（図 5.10(c)）は潜在変数より(5.4.8)式によって求めた。演奏データ正解値（図 5.10(e)）と近いものとなっていることが分かる。このことは、 SGPLVM の学習によって音響データ、演奏データが共有する潜在変数を獲得することができ、更に未知の音響データに対応する潜在変数の推定を介し、演奏データを推定することが可能であることを示している。

演奏データ推定値に後処理を施したものが図 5.10(d)である。演奏データ推定値に残っていた微小な変動が除去され、押鍵の有無がより明確になっている。 図 5.11 に SGPLVM 自動採譜アルゴリズムの音高別の推定精度を示す。また前章との比較のため、ガウス過程回帰による推定結果も併せて示す。SGPLVM 自動採譜アルゴリズムの全音域を通じての推定結果は再現率 0.751、精度 0.810、F 値は 0.780 であり、ガウス過程回帰自動採譜アルゴリズムの推定結果よりも良好な結果が得られた。図 5.11 より、低音域（C2 以下）および高音域（A#5 以上）での推定精度が向上したことが示されており、このことが全音域での推定精度向上に繋がったものと考えられる。

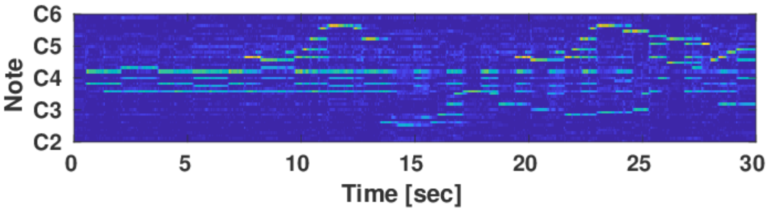
表 5.3 に本研究と同じテスト用楽曲を使用した先行研究における採譜結果とともに、本章で提案した手法による採譜結果を示す。



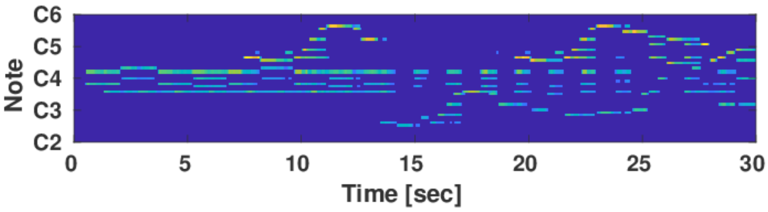

(a)音響データ



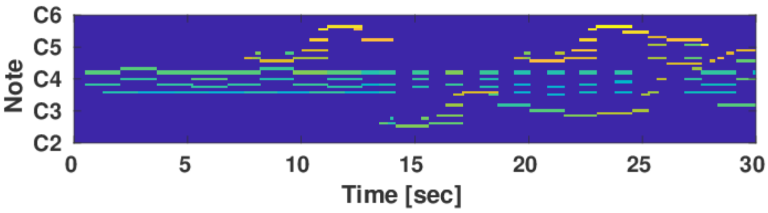

(b)潜在変数推定値




(c)演奏データ推定値

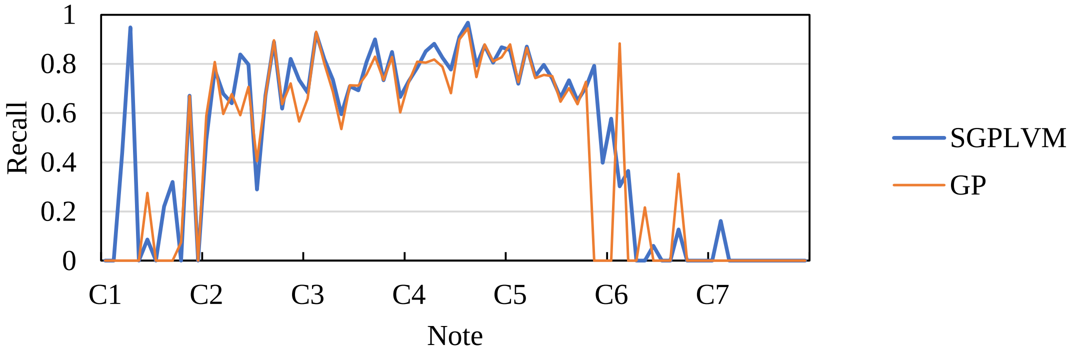



(d)後処理後の演奏データ推定値

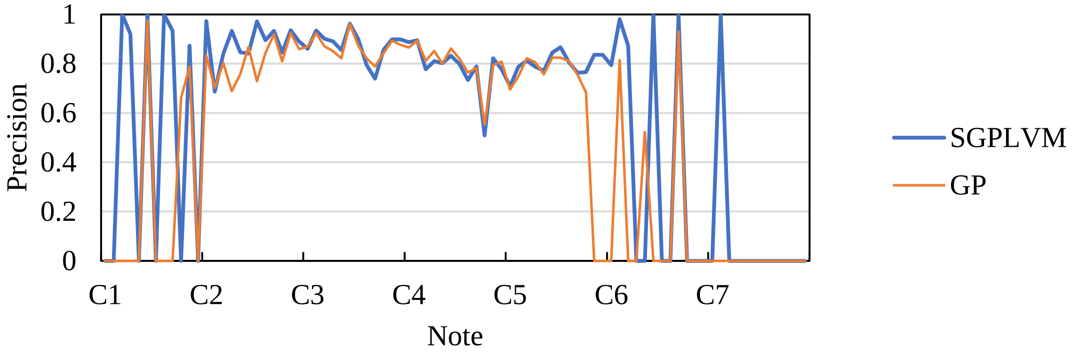



(e)演奏データ正解値

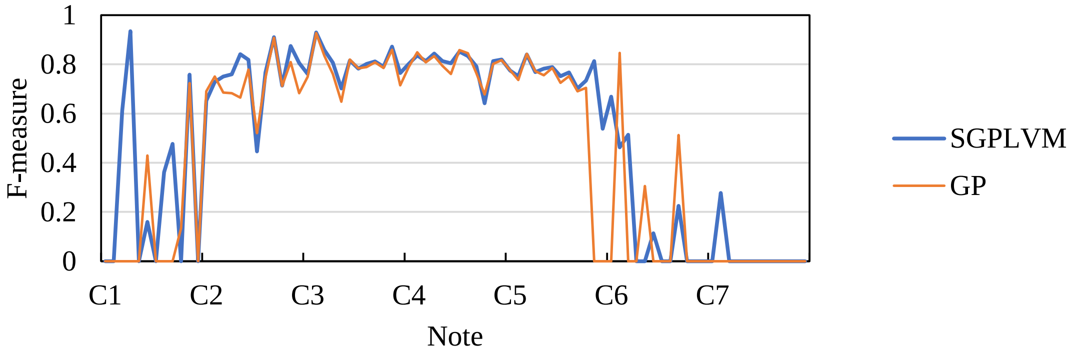
図 5.10 テスト用楽曲に対する推定例




(a)再現率




(b)適合率




(c)F 値

図 5.11　SGPLVM 自動採譜アルゴリズムの音高別推定精度

表 5.3　先行研究および本研究での採譜結果（フレームレベル）

| System | Precision | Recall | F-measure |
|---|---|---|---|
| Benetos and Weyde (2013) | - | - | 0.680 |
| Vincent et al. (2010)* | 0.796 | 0.636 | 0.707 |
| O’Hanlon and Plumbley (2014) | 0.755 | 0.705 | 0.729 |
| Berg-Kirkpatrick et al. (2014) | 0.691 | 0.807 | 0.744 |
| Cheng et al. (2015) | 0.854 | 0.729 | 0.777 |
| Cheng et al. (2016) | - | - | **0.790** |
| Gaussian Process Regression (Chapter 4) | 0.716 | 0.810 | 0.760 |
| SGPLVM (Chapter 5) | 0.751 | 0.810 | 0.780 |

* 原文ではテスト用楽曲の構成が本研究とは異なるため、Berg-Kirkpatrick et al. (2014)で評価された結果を記載。

## 5.6　まとめと考察

本章では、音響データと演奏データが同一の潜在変数を共有する構造をSGPLVMによりモデル化し、ガウス過程回帰の場合と同様、各音高のベロシティ値を反映した推定値が得られることを確認した。さらに rank order filter による平滑化処理およびレベル閾値，継続フレーム数閾値による枝刈り処理を行って後処理を施すことで、自動採譜アルゴリズムを構築した。その結果、全音域を通じて前章を上回る F 値 0.780 の推定結果を得ることができた。

本章冒頭で述べたように、本章では音響データと演奏データが持つ情報を共通のより低次元な潜在変数に縮約し、推定精度の向上を図ることを目的とした。推定は、前章と同様に打鍵，離鍵のタイミングだけでなく、打鍵の強さ（ベロシティ値）を求める回帰問題として扱った。図 5.6 に示した結果から、ガウス過程回帰の場合と同様に、打鍵の強さを反映した推定結果を得られることを示すことができ、また RMSE についても改善することができた。ただしガウス過程回帰の場合と同様に、一定値のベロシティ値に対して減衰途中の様々な大きさの振幅スペ

クトルが対応した形で学習が行われるといった問題は依然として残っており、このことが推定結果のばらつきを発生させているものと考えられる。前章と同様にこの点の改善が今後の課題である。

図 5.7 に示した潜在変数次数の評価結果より、推定精度は潜在変数の次数に依存することが分かった。また潜在変数の次数が、ピアノの鍵盤数と同じ 88 次の場合に最も高い推定精度を示したことから、良好な推定性能を得るためには、潜在変数が鍵盤数と同程度の自由度を持つ必要があると考えられる。

ガウス過程回帰の場合、学習データのフレーム数を 2000 フレームとした時点でほぼ推定結果の改善幅が飽和し、その後学習フレーム数を増やしても推定結果の改善はわずかであったが（図 4.6）、SGPLVM の場合は 3000 フレームの時点でガウス過程回帰の推定結果を上回り、学習フレーム数の増加に伴って更に推定精度が向上している（図 5.8）。本章で提案した手法が少ない学習データから効率的に学習を行うことができ、学習データの増加により更なる推定結果の改善の余地があることを示している。

後処理については、ガウス過程回帰の場合と同様に、rank order filter による平滑化処理とレベル閾値，継続フレーム数閾値による枝刈り処理の組み合わせによって大きな効果を得ることができ、この手法の有効性を改めて示すことができた。

後処理後の全音域を通じた F 値は 0.780 であったが、図 5.11 に示した音高別の推定結果を見ると、C2 以下の低音域および A#5 以上の高音域で推定精度の改善が見られ、このことが全音域の推定結果向上に繋がったものと考えられる。本章で用いた学習データは前章で用いたものと同じものを使用したが、図 5.8 に示したように 3000 フレームの学習データを用いた時点で前章の推定結果を上回った点、および出現頻度の低い低音域および高音域について推定精度の改善が見られた点から、SGPLVM が少ない学習データで効率的に学習を行えることが示され、本章で提案した観測変数の共有、および潜在変数の導入についての有効性が示された。

表 5.3 に記載した先行研究との比較では、最も高い推定精度を示した Cheng et al. (2016)にはわずかに及ばなかったものの、ほぼ同程度の推定性能を実現しており、

本手法の有効性を示すことができた。

# 第 6 章　Online Shared Gaussian Process Dynamical Model による自動採譜

## 6.1　はじめに

第 4 章、第 5 章ではそれぞれガウス過程回帰、SGPLVM によって 1 フレームずつ音響データに対応する演奏データを推定した。この場合、各フレームの推定値は相互に依存性を持たず、フレームごとに独立した結果となる。

一方で音楽や音声といった音響データは時間方向の連続性を持つ情報であり、自動採譜においても時間情報を考慮することで、フレームごとに未知の演奏データを推定することの困難さを緩和する必要性が指摘されてきた（Poliner and Ellis（2007)、Kameoka（2007)、亀岡・嵯峨山（2009))。Poliner らはサポートベクトルマシンによって対象音高の有無を識別し、後処理として隠れマルコフモデルを適用して時間方向の平滑化を行った（Poliner and Ellis（2007))。また kameoka は楽音のスペクトルを時間－周波数平面上に置かれたオブジェクトとして扱い、時間方向の連続性に対処している（Kameoka（2007))。近年自動採譜への適用例の多い非負値行列因子分解（Lee and Seung（2000)）も、スペクトログラムを周波数方向、時間方向へ分解することで、時間方向の連続性を考慮した手法とみなすことができる。

時間とともに変化するデータに対しては、実際には観測されない内部的な状態を表す状態変数と呼ばれる変数を置き、状態変数の動特性を表すシステムモデルと、状態変数から観測変数が生成される様を表す観測モデルを組み合わせた状態空間モデル（片山（2004)、北川（2005)、樋口 他（2011)）を使用して、観測値に対応する状態変数の推定を行うといった手法が、制御工学やデータ同化、マーケティングといった分野で広く用いられている。状態空間モデルの自動採譜への応用例としては深山らの研究が挙げられる（深山・田中（2009)、Fukayama and Tanaka (2013))。ピアノ演奏の音響データに対して鍵盤数と同数の 88 次元の状態変数を置き、部分空間同定法（片山（2004)）と呼ばれる線形状態空間モデルの同

定手法によってモデルを同定して状態変数の推定を行っており、状態変数の推定結果がピアノロールと類似した挙動を示すことを確認している。

本章では、前章で導入した潜在変数の動特性をモデル化することでシステムモデルを構成し、潜在変数（状態変数）から観測値である音響データ、演奏データを生成するモデルを観測モデルとして非線形状態空間モデルを構築する手法を提案する。即ち、自動採譜の問題を非線形状態空間モデルにおける推定問題として扱うことで、フレームごとに演奏データを推定する困難さの緩和を図り、推定精度の向上を目指す。

また、オンライン学習によって学習用楽曲の全フレームを評価することで大量の音響データ、観測データを学習することを可能とし、特に各音高のオンセットやオフセットといった発生頻度の低い現象、あるいは楽曲中での出現頻度の低い低音域、高音域の音高を効率的に学習することで、推定精度の改善を図る。

### 6.2　Gaussian Process Dynamical Model（GPDM）

5.2 節で説明した GPLVM では潜在変数 $\mathbf{x}_{\mathrm{i}}$ はフレームごとに独立しており、時間的な依存性はなかったが、Wang らは各フレームの $\mathbf{x}_i$ が依存性を持ち、ガウス過程により時間発展する Gaussian Process Dynamical Model（Wang et al.（2006），以下 GPDM と表記）を提案した（図 6.1）。

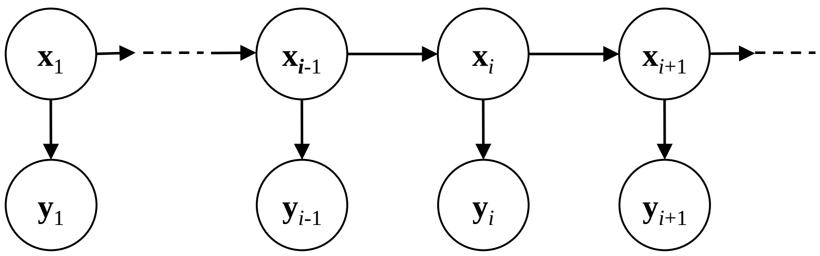



図 6.1　GPDM のグラフィカルモデル

1 期前の潜在変数を $\mathbf{x}_{i\text{-}1}$、現時点の潜在変数を $\mathbf{x}_i$、観測値を $\mathbf{y}_i$ とすると、$\mathbf{x}_{i\text{-}1}$ か

ら $\mathbf{x}_i$、および $\mathbf{x}_i$ から $\mathbf{y}_i$ のマッピングは以下のように表すことができる。

$$\mathbf{x}_i = f(\mathbf{x}_{i-1};\mathbf{A}) + \boldsymbol{\varepsilon}_{x,i} \tag{6.2.1}$$

$$\mathbf{y}_i = g(\mathbf{x}_i;\mathbf{B}) + \boldsymbol{\varepsilon}_{y,i} \tag{6.2.2}$$

$\boldsymbol{\varepsilon}_{x,i}$、$\boldsymbol{\varepsilon}_{y,i}$ はそれぞれ平均 **0** のガウス分布に従うノイズである。また **A**、**B** はそれぞれ $f$、$g$ のマッピングパラメータであり、$f$ および $g$ をそれぞれ、

$$f(\mathbf{x};\mathbf{A}) = \sum_i \mathbf{a}_i \varphi_i(\mathbf{x}) \tag{6.2.3}$$

$$g(\mathbf{x};\mathbf{B}) = \sum_j \mathbf{b}_j \psi_j(\mathbf{x}) \tag{6.2.4}$$

と表すと、$\mathbf{A}=[\mathbf{a}_1,\mathbf{a}_2,\dots]$、$\mathbf{B}=[\mathbf{b}_1,\mathbf{b}_2,\dots]$と表せる。

ここで $\mathbf{X}=[\mathbf{x}_1,\dots,\mathbf{x}_N]^{\mathrm{T}}$、$\mathbf{Y}=[\mathbf{y}_1,\dots,\mathbf{y}_N]^{\mathrm{T}}$、**W** を観測変数 **y** のスケーリング係数（Grochow et al.（2004））とし、カーネル関数 $k_{\mathrm{x}}(\mathbf{x}_i,\mathbf{x}_j)=\varphi(\mathbf{x}_i)\varphi(\mathbf{x}_j)$のパラメータを $\boldsymbol{\alpha}=\{\alpha_1,\alpha_2,\dots\}$、カーネル関数 $k_{\mathrm{y}}(\mathbf{x}_i,\mathbf{x}_j)=\psi(\mathbf{x}_i)\,\psi(\mathbf{x}_j)$のパラメータを $\boldsymbol{\beta}=\{\beta_1,\beta_2,\dots,\mathbf{W}\}$とすると、

$$p(\mathbf{X}\,|\,\boldsymbol{\alpha}) = \int p(\mathbf{X},\mathbf{A}\,|\,\boldsymbol{\alpha})d\mathbf{A} = \frac{1}{\sqrt{(2\pi)^{(N-1)d}|\mathbf{K}_X|^d}}\exp\left(-\frac{1}{2}tr(\mathbf{K}_X^{-1}\mathbf{X}_{out}\mathbf{X}_{out}^T)\right) \tag{6.2.5}$$

$$p(\mathbf{Y}\,|\,\mathbf{X},\boldsymbol{\beta}) = \int p(\mathbf{Y},\mathbf{B}\,|\,\mathbf{X},\boldsymbol{\beta})d\mathbf{B} = \frac{|\mathbf{W}|^N}{\sqrt{(2\pi)^{ND}|\mathbf{K}_Y|^D}}\exp\left(-\frac{1}{2}tr(\mathbf{K}_Y^{-1}\mathbf{Y}\mathbf{W}^2\mathbf{Y}^T)\right) \tag{6.2.6}$$

となる。$d$、$D$ はそれぞれ $\mathbf{x}_i,\mathbf{y}_i$ の次数、$\mathbf{K}_{\mathrm{x}}$、$\mathbf{K}_{\mathrm{y}}$ はそれぞれ $k_{\mathrm{x}}(\mathbf{x}_i,\mathbf{x}_j)$、$k_{\mathrm{y}}(\mathbf{x}_i,\mathbf{x}_j)$を$(i,j)$要素とするカーネル行列、$\mathbf{X}_{out}=[\mathbf{x}_2,\dots,\mathbf{x}_N]^{\mathrm{T}}$ である。Wang らは、$\boldsymbol{\alpha}$、$\boldsymbol{\beta}$ の事前分布をそれぞれ $p(\boldsymbol{\alpha}) \propto \Pi\alpha_i^{-1}$、 $p(\boldsymbol{\beta}) \propto \Pi\beta_i^{-1}$ とし、以下の負対数事後分布を **X**、$\boldsymbol{\alpha}$、$\boldsymbol{\beta}$ について最小化することによって GPDM の学習を行っている。

$$L = -\ln p(\mathbf{X},\boldsymbol{\alpha},\boldsymbol{\beta}\,|\,\mathbf{Y}) \propto p(\mathbf{Y}\,|\,\mathbf{X},\boldsymbol{\alpha},\boldsymbol{\beta})p(\mathbf{X}\,|\,\boldsymbol{\alpha})p(\boldsymbol{\alpha})p(\boldsymbol{\beta})$$

$$= \frac{d}{2}\ln\left|\mathbf{K}_X\right| + \frac{1}{2}tr(\mathbf{K}_X^{-1}\mathbf{X}_{out}\mathbf{X}_{out}^T) + \sum_i \ln \alpha_i$$

$$-N\ln\left|\mathbf{W}\right| + \frac{D}{2}\ln\left|\mathbf{K}_Y\right| + \frac{1}{2}(\mathbf{K}_Y^{-1}\mathbf{Y}\mathbf{W}^2\mathbf{Y}^T) + \sum_i \ln \beta_i \tag{6.2.7}$$

### 6.3 Shared Gaussian Process Dynamical Model（SGPDM）

GPLVM と同様に、GPDM についても複数の観測変数が同一の潜在変数を共有するモデルを考えることができる（Ek et al.（2007），Deena（2012））。

$$\mathbf{x}_i = f(\mathbf{x}_{i-1};\mathbf{A}) + \boldsymbol{\varepsilon}_{x,i} \tag{6.3.1}$$

$$\mathbf{y}_i = g(\mathbf{x}_i;\mathbf{B}) + \boldsymbol{\varepsilon}_{y,i} \tag{6.3.2}$$

$$\mathbf{z}_i = h(\mathbf{x}_i;\mathbf{C}) + \boldsymbol{\varepsilon}_{z,i} \tag{6.3.3}$$

$\boldsymbol{\varepsilon}_{z,i}$ は平均 **0** のガウス分布に従うノイズであり、また **C** は $h$ のマッピングパラメータである。(6.3.1)式、(6.3.2)式はそれぞれ(6.2.1)式、(6.2.2)式と同一だが、$\mathbf{y}_\mathrm{i}$ とは異なる観測変数 $\mathbf{z}_\mathrm{i}$ が潜在変数 $\mathbf{x}_\mathrm{i}$ を $\mathbf{y}_\mathrm{i}$ と共有している。このモデルは Shared Gaussian Process Dynamical Model（Deena (2012)、以下 SGPDM と表記）と呼ばれ、共有する潜在変数が時間発展する点が SGPLVM と異なっている（図 6.2）。

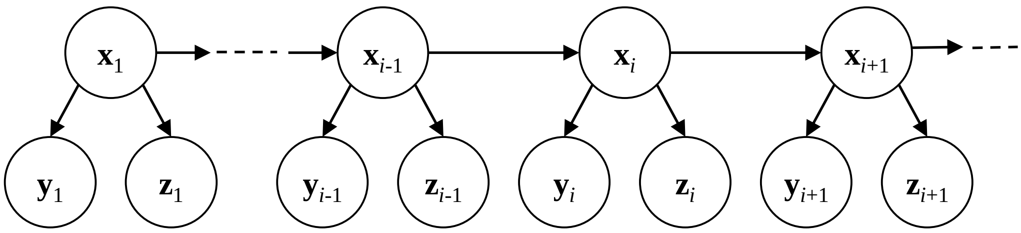



図 6.2　SGPDM のグラフィカルモデル

潜在変数 X から観測変数 Y へのマッピングについての超パラメータを $\mathbf{\Phi}_\mathrm{Y}$、**X** からもう一方の観測変数 **Z** へのマッピングについての超パラメータを $\mathbf{\Phi}_\mathrm{Z}$、**X** の

動特性についての超パラメータを $\mathbf{\Phi}_{dyn}$ とし、$\mathbf{\Phi}$=[$\mathbf{\Phi}_{\mathrm{Y}}$,$\mathbf{\Phi}_{\mathrm{Z}}$,$\mathbf{\Phi}_{dyn}$]とすると、SGPDMの学習は以下の同時尤度の最大化によって行われる。

$$p(\mathbf{Y},\mathbf{Z},\mathbf{X}\,|\,\mathbf{\Phi}) = p(\mathbf{Y}\,|\,\mathbf{X},\mathbf{\Phi}_Y)\,p(\mathbf{Z}\,|\,\mathbf{X},\mathbf{\Phi}_Z)\,p(\mathbf{X}\,|\,\mathbf{\Phi}_{dyn}) \tag{6.3.4}$$

また、未知の観測変数として $\mathbf{y}^*$のみが与えられ、それに対応する $\mathbf{z}^*$を推定することもできる。Ek らは $\mathbf{y}^*$に対応する潜在変数 $\mathbf{x}^*$を以下の最適化によって点推定し、

$$\mathbf{x}^* = \arg\max_{\mathbf{x}}\; p(\mathbf{y}^*,\mathbf{x}\,|\,\mathbf{Y},\mathbf{X},\mathbf{\Phi}_Y,\mathbf{\Phi}_{dyn}) \tag{6.3.5}$$

$\mathbf{x}^*$に対する $\mathbf{z}^*$の推定値を以下のガウス過程回帰により求めている。

$$\mathbf{z}^* = \mathbf{k}_Z^T\mathbf{K}_Z^{-1}\mathbf{Z} \tag{6.3.6}$$

ここで、$\mathbf{k}_Z$=[$k_Z(\mathbf{x}^*,\mathbf{x}_1),\ldots,k_Z(\mathbf{x}^*,\mathbf{x}_N)$]$^{\mathrm{T}}$ である。

### 6.4 Online Shared Gaussian Process Dynamical Model（OSGPDM）

SGPDM は学習の負荷が高く、またバッチ学習であるため、学習データのサイズがコンピュータのメモリ容量により制限されてしまう。これらの点は特に大量のデータを学習に用いる場合に大きな制約となる。本節ではこのような問題に対処すべく、Cubature Kalman Filter による状態推定と、オンラインガウス過程回帰によるオンライン学習を組み合わせた非線形状態空間モデルの学習方法を提案し、自動採譜モデルアルゴリズムへの応用を行う。

#### 6.4.1 アルゴリズムの全体像

(6.2.1)式、(6.2.2)式は状態空間モデルと同一の形式をしており、(6.2.1)式をシステムモデル、(6.2.2)式を観測モデル、$\mathbf{x}_i$ を状態変数とみなすことができる。この時、$\mathbf{y}_{\mathrm{i}}$ の観測の下での $\mathbf{x}_{\mathrm{i}}$ の分布は、

$$p(\mathbf{x}_i\,|\,\mathbf{Y}_i) \propto p(\mathbf{y}_i\,|\,\mathbf{x}_i)\,p(\mathbf{x}_i\,|\,\mathbf{Y}_{i-1}) \tag{6.4.1}$$

と推定できる。ただし $\mathbf{Y}_i$=[$\mathbf{y}_i,\ldots,\mathbf{y}_i$]$^{\mathrm{T}}$ である。カルマンフィルタや粒子フィルタは、

1 データずつ与えられる観測データ系列 $\mathbf{y}_i$ に対し、1 期先予測のステップで予測分布 $p(\mathbf{x}_i|\mathbf{Y}_{i\text{-}1})$を、フィルタリングステップで尤度 $p(\mathbf{y}_i|\mathbf{x}_i)$を求め、逐次的にフィルタ分布 $p(\mathbf{x}_i|\mathbf{Y}_i)$を推定する。このような逐次アルゴリズムを総称して逐次ベイズフィルタと呼ぶ（樋口 他（2011））。

通常、システムモデルおよび観測モデルがすでに求められている状況下で逐次ベイズフィルタによるフィルタリングが行われるが、何らかのオンライン学習によってシステムモデル、観測モデルの学習が可能であれば、逐次ベイズフィルタとモデルのオンライン学習を組み合わせて状態空間モデルを同定することが可能であると考えられる。すなわち時刻 $i$ における(6.2.1)式、(6.2.2)式の $f$、$g$ を $f_{i\text{-}1}$、$g_{i\text{-}1}$ で置き換えて、

$$\mathbf{x}_i = f_{i-1}(\mathbf{x}_{i-1};\mathbf{A}) + \boldsymbol{\varepsilon}_{x,i} \tag{6.4.2}$$

$$\mathbf{y}_i = g_{i-1}(\mathbf{x}_i;\mathbf{B}) + \boldsymbol{\varepsilon}_{y,i} \tag{6.4.3}$$

とし、学習用観測データ系列 $\mathbf{y}_i$（$i$=1,...,$N$）を 1 データずつ与えて $p(\mathbf{x}_i|\mathbf{Y}_i)$の推定を行うとともに、$f_{i\text{-}1}$、$g_{i\text{-}1}$ を更新する。さらに未知のテスト用観測データ系列 $\mathbf{y}_i^*$を 1 データずつ与え、以下の状態空間モデル

$$\mathbf{x}_i = f_N(\mathbf{x}_{i-1};\mathbf{A}) + \boldsymbol{\varepsilon}_{x,i} \tag{6.4.4}$$

$$\mathbf{y}_i = g_N(\mathbf{x}_i;\mathbf{B}) + \boldsymbol{\varepsilon}_{y,i} \tag{6.4.5}$$

によって $p(\mathbf{x}_i^* \mid \mathbf{Y}_i^*)$ を推定する。

続いて、一つの状態変数 $\mathbf{x}_i$ を観測変数 $\mathbf{y}_i$、$\mathbf{z}_i$ が共有する(6.3.1)式~(6.3.3)式の場合について考える。この時、学習時には $\mathbf{y}_i$、$\mathbf{z}_i$ が与えられ、またテスト時には $\mathbf{y}_i^*$のみが与えられ、対応する $\mathbf{z}_i^*$を求めるものとする。

学習時には $\mathbf{y}_i$、$\mathbf{z}_i$ を一つにまとめた $[\mathbf{y}_i^T, \mathbf{z}_i^T]^T$ を拡張した観測値とみなして学習を行う。すなわち、拡張した状態空間モデル

$$\mathbf{x}_i = f_{i-1}(\mathbf{x}_{i-1};\mathbf{A}) + \boldsymbol{\varepsilon}_{x,i} \tag{6.4.6}$$

$$\begin{bmatrix} \mathbf{y}_i \\ \mathbf{z}_i \end{bmatrix} = \begin{bmatrix} g_{i-1}(\mathbf{x}_i;\mathbf{B}) \\ h_{i-1}(\mathbf{x}_i;\mathbf{C}) \end{bmatrix} + \begin{bmatrix} \boldsymbol{\varepsilon}_{y,i} \\ \boldsymbol{\varepsilon}_{z,i} \end{bmatrix} \tag{6.4.7}$$

によって $p(\mathbf{x}_i|\mathbf{Y}_i,\mathbf{Z}_i)$を推定する。$f_{i\text{-}1}$ は $\mathbf{x}_{i\text{-}1}$ と $\mathbf{x}_i$ の推定値を、$g_{i\text{-}1}$ は $\mathbf{x}_i$ の推定値と $\mathbf{y}_i$ を、$h_{i\text{-}1}$ は $\mathbf{x}_i$ の推定値と $\mathbf{z}_i$ をそれぞれ使用してモデルの更新を行う。

テスト時においては $\mathbf{z}_i^*$ が与えられないので、まず $g_N$ と未知のテスト用観測データ系列 $\mathbf{y}_i^*$ によってフィルタリングを行い、フィルタ分布 $p(\mathbf{x}_i^* \mid \mathbf{Y}_i^*)$ を求める。さらに $p(\mathbf{x}_i^* \mid \mathbf{Y}_i^*)$ と、$h_N$ により与えられる $p(\mathbf{z}_i^* \mid \mathbf{x}_i^*)$ を用い、以下の周辺化を行って $\mathbf{z}_i^*$ の推定値を得る。

$$p(\mathbf{z}_i^* \mid \mathbf{Y}_i^*) = \int p(\mathbf{z}_i^* \mid \mathbf{x}_i^*) p(\mathbf{x}_i^* \mid \mathbf{Y}_i^*) d\mathbf{x}_i^* \tag{6.4.8}$$

本項で説明した学習アルゴリズム、推定アルゴリズムをそれぞれ図 6.3、図 6.4 に示す。

### 6.4.2 Cubature Kalman Filter

状態空間モデルにおけるシステムモデル、観測モデルが線形モデルで記述され、システムノイズ、観測ノイズがガウス分布に従う場合、カルマンフィルタ（Kalman（1960））によって観測変数 $\mathbf{y}_i$ より状態変数 $\mathbf{x}_i$ を逐次推定することができる。しかしシステムモデル、観測モデルが非線形であったり、ノイズが非ガウス分布に従う場合はカルマンフィルタの適用ができず、このようなケースへの対応として様々な逐次ベイズフィルタのアルゴリズムが提案されてきた。

一つの流れはカルマンフィルタのアルゴリズムを拡張・発展させたもので、拡張カルマンフィルタ（Anderson and Moore（1979））、アンサンブルカルマンフィルタ（Evensen （1994））、Unscented カルマンフィルタ（Julier et al.（2000）、以下 UKF と表記）などが例として挙げられる。

他方、多数の粒子で状態変数やノイズの分布を近似し、粒子のサンプリングとリサンプリングを繰り返して状態変数の推定を行う粒子フィルタ（Gordon et al.（1993）、Kitagawa （1996））も提案されている。粒子フィルタは、扱うモデルの

Train Initial Model

$f_0, g_0, h_0$

for $i$ = 1 to $N$

Prediction Step

$$p(\mathbf{x}_{i-1} \mid \mathbf{Y}_{1:i-1}, \mathbf{Z}_{1:i-1}) \xrightarrow{f_{i-1}} p(\mathbf{x}_i \mid \mathbf{Y}_{1:i-1}, \mathbf{Z}_{1:i-1})$$

Filtering Step

observe $\mathbf{y}_i, \mathbf{z}_i$

$$p(\mathbf{x}_i \mid \mathbf{Y}_{1:i-1}, \mathbf{Z}_{1:i-1}) \xrightarrow{g_{i-1}, h_{i-1}} p(\mathbf{x}_i \mid \mathbf{Y}_{1:i}, \mathbf{Z}_{1:i})$$

Training Step

$$f_{i-1} \rightarrow f_i$$
$$g_{i-1} \rightarrow g_i$$
$$h_{i-1} \rightarrow h_i$$

end

図 6.3　学習アルゴリズム

for $i$ = 1 to $M$

Prediction Step

$$p(\mathbf{x}^*_{i-1} \mid \mathbf{Y}^*_{1:i-1}) \xrightarrow{f_N} p(\mathbf{x}^*_i \mid \mathbf{Y}^*_{1:i-1})$$

Filtering Step

observe $\mathbf{y}^*_i$

$$p(\mathbf{x}^*_i \mid \mathbf{Y}^*_{1:i-1}) \xrightarrow{g_N} p(\mathbf{x}^*_i \mid \mathbf{Y}^*_{1:i})$$

Estimation Step

$$p(\mathbf{x}^*_i \mid \mathbf{Y}^*_{1:i}) \xrightarrow{h_N} p(\mathbf{z}^*_i \mid \mathbf{x}^*_i)$$
$$p(\mathbf{z}^*_i \mid \mathbf{Y}^*_i) = \int p(\mathbf{z}^*_i \mid \mathbf{x}^*_i)\, p(\mathbf{x}^*_i \mid \mathbf{Y}^*_i)\, d\mathbf{x}^*_i$$

end

図 6.4　推定アルゴリズム

形式やノイズの分布に制約が無く、広範なモデルに柔軟に対応できるが、状態変数の次数が高い場合は非常に多くの粒子が必要であり、計算負荷が高くなるという問題点がある。高次元の状態変数に対する粒子フィルタの適用としては、Nakano らの Merging particle filter（Nakano et al. （2007））などの研究例がある。

本研究では逐次ベイズフィルタのアルゴリズムとして、近年 Arasaratnam と Haykin によって提案された Cubature Kalman filter（Arasaratnam and Haykin（2009）、以下 CKF と表記）を用いる。CKF は spherical radial rule と呼ばれる方法によって状態変数の分布を有限個の点の集合で近似し、一つ一つの点に対してカルマンフィルタの処理を施すという点で UKF と類似したアルゴリズムであるが、数値的安定性、推定精度といった点で UKF よりも優れており、また高次元の状態変数への適用も可能であることが示されている。

CKF で非線形状態空間モデルの状態推定を行う場合、数値的な不安定性によって計算プログラムが停止する場合がある。このような場合、Square-Root Cubature Kalman Filter（Arasaratnam and Haykin （2009）、以下 SRCKF と表記）を適用することで数値的不安定性を回避することができる。本研究における自動採譜アルゴリズムの構築にあたっては、SRCKF を逐次ベイズフィルタアルゴリズムとして用いる。

### 6.4.3　オンラインガウス過程回帰

ガウス過程回帰はバッチ式の学習であり、データ数 $N$ に対して計算負荷は $O(N^3)$ となる。そのため、データが一つずつ与えられ、その都度モデルを更新するといった用途に対しては計算負荷の増大により適用が困難となる。このような状況へのガウス過程の適用のため、少ない基底ベクトルでモデルを近似し、オンライン学習によってモデルの更新を行う方法がいくつか提案されている（Csató and Opper（2002）, Van Vaerenbergh et al.（2012）, Bijl et al.（2015））。

本研究では Csató and Opper による Sparse Online Gaussian Processes（以下 SOGP と表記）を用い、状態空間モデルにおけるシステムモデル、観測モデルをオンライン学習する。SOGP では、未知入力 $\mathbf{x}^*$ に対する出力の分布を以下のように推定

する。

$$N(y^* \mid \mathbf{k}(\mathbf{x}^*, \mathbf{X}_I)\boldsymbol{\alpha}_i, k(\mathbf{x}^*, \mathbf{x}^*) + \mathbf{k}(\mathbf{x}^*, \mathbf{X}_I)\mathbf{C}_i\mathbf{k}(\mathbf{X}_I, \mathbf{x}^*)) \tag{6.4.9}$$

ここで $\mathbf{X}_I$ はモデルの基底ベクトルとして取り込んだ学習データ系列$\{\mathbf{x}_1,\ldots,\mathbf{x}_m\}$であり、

$$\mathbf{k}(\mathbf{x}^*, \mathbf{X}_I) = \mathbf{k}(\mathbf{X}_I, \mathbf{x}^*)^T = [k(\mathbf{x}^*, \mathbf{x}_1), \ldots, k(\mathbf{x}^*, \mathbf{x}_m)] \tag{6.4.10}$$

である。学習用データを一つずつ与え、必要な学習データを $\mathbf{X}_I$ に取り込みながら $\boldsymbol{\alpha}_i$, $\mathbf{C}_i$ を逐次更新する。また $m$ はあらかじめ定めた最大基底ベクトル数であるが、取り込んだ基底ベクトルがこの値を超えると、$\mathbf{X}_I$ の中から最も推定値に対する影響の弱い基底ベクトルを削除して基底ベクトル数を $m$ に保つ。

SOGP は 1 出力の回帰モデルなので、入力 $\mathbf{x}^*$に対して $l$ 次元の出力 $\mathbf{Y}^* = [y_1^*, \ldots, y_l^*]^T$ を推定したい場合は $l$ 個の SOGP を用いることで対応できる。この時、$\mathbf{X}_I$, $\boldsymbol{\alpha}_i$, $\mathbf{C}_i$ を $l$ 個の SOGP で共有することで計算負荷を下げることができる。以下、このような方法で多次元化した SOGP を MSOGP（Multi-output Sparse Online Gaussian Process）と呼ぶ。(6.4.9)式より明らかなように、MSOGP では各次元の分散は同じ値となる。

## 6.5 OSGPDM による自動採譜アルゴリズム

本節ではこれまでに説明した SRCKF および MSOGP を用い、時系列モデルとして自動採譜モデルを学習、推定する自動採譜アルゴリズムについて説明する。

### 6.5.1 モデルの学習

状態変数、音響データ、演奏データをそれぞれ $\mathbf{x}_i$、$\mathbf{y}_i$、$\mathbf{z}i$ とする。前章と同様に $\mathbf{y}_i$、$\mathbf{z}_i$ は連続する $r$ フレームを連結して特徴量とする。学習においては$[\mathbf{y}_i^T, \mathbf{z}_i^T]^T$を観測値とし、それに対応した $\mathbf{x}_i$ を SRCKF で推定しながら MSOGP によりシステムモデル、観測モデルのオンライン学習を行う。

まずシステムモデルについて説明する。システムモデルは 1 期前の状態変数 $\mathbf{x}_{i\text{-}1}$

を入力し、$\mathbf{x}_i$ を出力する回帰モデルである。この時、システムノイズ $\boldsymbol{\varepsilon}_{\mathrm{x},i}$ が重畳するが、6.4.3 項で述べたように MSOGP では各次元の分散が等しくなってしまうため、$\mathbf{x}_i$ の平均 $\boldsymbol{\mu}_{\mathrm{x},i}$ と分散 $\boldsymbol{\Sigma}_x = [\sigma_1^2, \ldots, \sigma_d^2]^T$ をそれぞれ別の回帰モデルで推定する。

$$\boldsymbol{\mu}_{x,i} = f_{\mu,i-1}(\mathbf{x}_{i-1}; \mathbf{A}_\mu) \tag{6.5.1}$$

$$\overline{\mathbf{S}}_i = f_{\sigma,i-1}(\mathbf{x}_{i-1}; \mathbf{A}_\sigma) \tag{6.5.2}$$

ここで、$\overline{\mathbf{S}}_i = [\ln(\sigma_{1,i}^2), \ldots, \ln(\sigma_{d,i}^2)]^T$ である。(6.5.1)式、(6.5.2)式から推定される状態変数 $\mathbf{x}_i$ は $N(\mathbf{x}_i|\boldsymbol{\mu}_{\mathrm{x}.i},\exp(\mathrm{diag}(\bar{\mathbf{S}}_i)))$に従う。このように平均、分散がそれぞれ潜在変数（ここでは状態変数）に依存するモデルは異分散モデルとして知られている（Goldberg et al. (1998)）。Wang らは SOGP によって異分散モデルを構成し、deep gaussian process（Damianou and Lawrence (2013)）のオンライン学習を行っている (Wang et al. (2016)）。

(6.5.1)式は各フレームにおいて、1 期前の状態変数のフィルタ分布平均値 $\bar{\mathbf{x}}_{i-1}$ を入力、現時点のフィルタ分布平均値 $\bar{\mathbf{x}}_i$ を出力として学習を行う。また、(6.5.2)式は各フレームにおいて 1 期前の状態変数のフィルタ分布平均値 $\bar{\mathbf{x}}_{i-1}$ を入力とし、各次元の値が、

$$s_i^{(l)} = \ln(\bar{x}_i^{(l)} - \bar{x}_{i-1}^{(l)})^2 \tag{6.5.3}$$

であるベクトル $\mathbf{S}_i$ を出力として学習を行う。

続いて観測モデルについて説明する。観測モデルは現在の状態変数 $\mathbf{x}_i$ を入力し、観測値 $[\mathbf{y}_i^T, \mathbf{z}_i^T]^T$ の推定値を出力する回帰モデルであり、システムモデル同様に MSOGP による異分散モデルによって実現する。ただし、システムモデルの場合と異なり、分散の対数ではなく観測値と推定値の誤差ベクトル $\boldsymbol{\varepsilon}_{\mathrm{y},i}, \boldsymbol{\varepsilon}_{\mathrm{z},i}$ を推定する。

$$\boldsymbol{\mu}_{y,i} = g_{\mu,i-1}(\mathbf{x}_i; \mathbf{B}_\mu) \tag{6.5.4}$$

$$\boldsymbol{\varepsilon}_{y,i} = g_{e,i-1}(\mathbf{x}_i; \mathbf{B}_e) \tag{6.5.5}$$

$$\boldsymbol{\mu}_{z,i} = h_{\mu,i-1}(\mathbf{x}_i; \mathbf{C}_\mu) \tag{6.5.6}$$

$$\boldsymbol{\varepsilon}_{z,i} = h_{e,i-1}(\mathbf{x}_i;\mathbf{C}_e) \tag{6.5.7}$$

従って、観測モデルは以下のように構成できる。

$$\begin{bmatrix}\boldsymbol{\mu}_{y,i}\\ \boldsymbol{\mu}_{z,i}\end{bmatrix} = \begin{bmatrix} g_{\mu,i-1}(\mathbf{x}_i;\mathbf{B}_\mu) \\ h_{\mu,i-1}(\mathbf{x}_i;\mathbf{C}_\mu)\end{bmatrix} + \mathbf{W}_i \tag{6.5.8}$$

$\mathbf{W}_i$ は平均 $\mathbf{0}$、分散共分散行列が $\mathbf{R}_i$ のガウス性ノイズである。(6.5.5)式、(6.5.7)式で与えられる $\boldsymbol{\varepsilon}_{\mathrm{y},i},\boldsymbol{\varepsilon}_{\mathrm{z},i}$ の各次元は $\mathbf{x}_i$ で条件付けられた誤差の期待値であり、$\mathbf{x}_i$ についてtail to tailの有効グラフィカルモデルとして図6.5のように表すことができる。従って各次元は条件付き独立となり、$\mathbf{x}_i$ が与えられた状況下ではそれぞれの次元間の統計的依存性は遮断される。この性質を利用すると、$\mathbf{R}_i$ は以下のように近似することができる。$c_0\mathbf{I}$ は正則化項である。

$$\mathbf{R}_i = [\boldsymbol{\varepsilon}_{y,i}^T, \boldsymbol{\varepsilon}_{z,i}^T]^T [\boldsymbol{\varepsilon}_{y,i}^T, \boldsymbol{\varepsilon}_{z,i}^T] + c_0\mathbf{I} \tag{6.5.9}$$

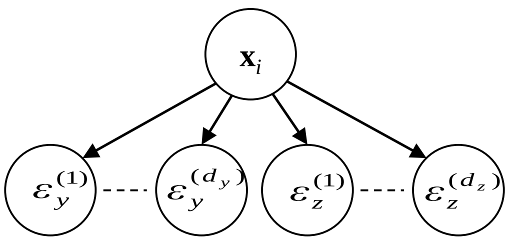



図 6.5　$\boldsymbol{\varepsilon}_{\mathrm{y},i},\boldsymbol{\varepsilon}_{\mathrm{z},i}$ のグラフィカルモデル

学習は、(6.5.1)式および(6.5.2)式で与えられるシステムモデルと、(6.5.4)式~(6.5.9)式で与えられる観測モデルを用い、SRCKFで各フレームの観測値 $[\mathbf{y}_i^T, \mathbf{z}_i^T]^T$ に対する状態変数のフィルタ分布 $p(\mathbf{x}_i|\mathbf{Y}_i)$ を推定し、その後にMSOGPによって $f_\mu$、$f_\sigma$、$g_\mu$、$g_\mathrm{e}$、$h_\mu$、$h_\mathrm{e}$ を更新するといった手順で進める。この処理を全学習用楽曲の全フレームに対して行い、システムモデル、観測モデルの学習を行う。

なお、$f_\mu$、$f_\sigma$、$g_\mu$、$g_\mathrm{e}$、$h_\mu$、$h_\mathrm{e}$ のカーネル関数は以下のものを用いる。

$$k_{x\mu}(\mathbf{x}_i,\mathbf{x}_j)=\theta_{x\mu 1}\exp\left(-\frac{\|\mathbf{x}_i-\mathbf{x}_j\|^2}{2\theta_{x\mu 2}^2}\right) \tag{6.5.10}$$

$$k_{x\sigma}(\mathbf{x}_i,\mathbf{x}_j)=\theta_{x\sigma 1}\exp\left(-\frac{\|\mathbf{x}_i-\mathbf{x}_j\|^2}{2\theta_{x\sigma 2}^2}\right) \tag{6.5.11}$$

$$k_{y\mu}(\mathbf{x}_i,\mathbf{x}_j)=\theta_{y\mu 1}\exp\left(-\frac{\|\mathbf{x}_i-\mathbf{x}_j\|^2}{2\theta_{y\mu 2}^2}\right) \tag{6.5.12}$$

$$k_{ye}(\mathbf{x}_i,\mathbf{x}_j)=\theta_{ye1}\exp\left(-\frac{\|\mathbf{x}_i-\mathbf{x}_j\|^2}{2\theta_{ye2}^2}\right) \tag{6.5.13}$$

$$k_{z\mu}(\mathbf{x}_i,\mathbf{x}_j)=\theta_{z\mu 1}\exp\left(-\frac{\|\mathbf{x}_i-\mathbf{x}_j\|^2}{2\theta_{z\mu 2}^2}\right) \tag{6.5.14}$$

$$k_{ze}(\mathbf{x}_i,\mathbf{x}_j)=\theta_{ze1}\exp\left(-\frac{\|\mathbf{x}_i-\mathbf{x}_j\|^2}{2\theta_{ze2}^2}\right) \tag{6.5.15}$$

### 6.5.2 各音高ベロシティ値の推定

システムモデル、観測モデルの学習が完了した後、テスト用楽曲の音響データ $\mathbf{y}_i^*$ を与え、演奏データ（各音高のベロシティ値） $\mathbf{z}_i^*$ の推定を行う。学習時と異なり、観測値は $\mathbf{y}_i^*$ のみとなるので、(6.5.1)式~(6.5.5)式を用いて $\mathbf{x}_i^*$ のフィルタ分布 $p(\mathbf{x}_i^* \mid \mathbf{y}_i^*)$ を推定する。$p(\mathbf{z}_i^* \mid \mathbf{Y}_i^*)$ は(6.4.8)式の周辺化により求めることができるが、分割されたガウス分布の性質から以下のように $p(\mathbf{z}_i^* \mid \mathbf{Y}_i^*)$ の平均 $\hat{\boldsymbol{\mu}}_{zi}^*$ および分散共分散行列 $\hat{\boldsymbol{\Sigma}}_{zz,i}^*$ を求めることができる（Bishop（2008））。

$$\hat{\boldsymbol{\mu}}_{z,i}^* = \boldsymbol{\mu}_z^* - \boldsymbol{\Sigma}_{zy,i}\boldsymbol{\Sigma}_{yy,i}^{-1}(\mathbf{y}_i^* - \boldsymbol{\mu}_y^*) \tag{6.5.16}$$

$$\hat{\boldsymbol{\Sigma}}^*_{zz,i} = \boldsymbol{\Sigma}_{zz,i} - \boldsymbol{\Sigma}_{zy,i}\boldsymbol{\Sigma}^{-1}_{yy,i}\boldsymbol{\Sigma}_{yz,i} \tag{6.5.17}$$

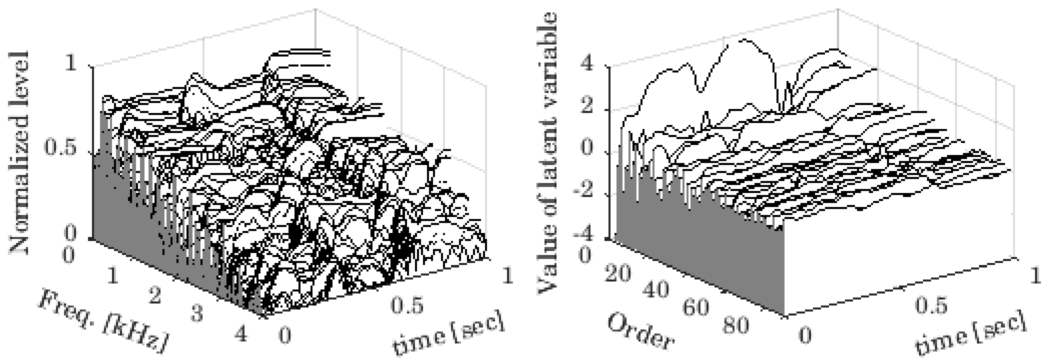
ただし、

$$\boldsymbol{\mu}^*_y = g_{\mu,N}(\mathbf{x}^*_i;\mathbf{B}_\mu) \tag{6.5.18}$$

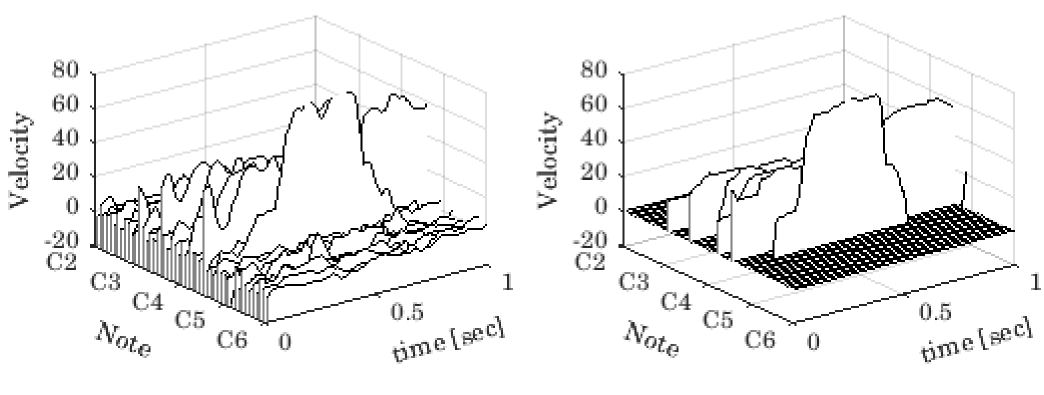
$$\boldsymbol{\mu}^*_z = h_{\mu,N}(\mathbf{x}^*_i;\mathbf{C}_\mu) \tag{6.5.19}$$

$$\begin{bmatrix} \boldsymbol{\Sigma}_{yy,i} & \boldsymbol{\Sigma}_{yz,i} \\ \boldsymbol{\Sigma}_{zy,i} & \boldsymbol{\Sigma}_{zz,i} \end{bmatrix} = [g_{e,N}(\mathbf{x}^*_i;\mathbf{B}_e)^T, h_{e,N}(\mathbf{x}^*_i;\mathbf{C}_e)^T]^T [g_{e,N}(\mathbf{x}^*_i;\mathbf{B}_e)^T, h_{e,N}(\mathbf{x}^*_i;\mathbf{C}_e)^T] + c_0\mathbf{I} \tag{6.5.20}$$

である。




(a)音響データ　　(b)潜在変数推定値




(c)演奏データ推定値　　(d)後処理後の演奏データ推定値

図 6.6 音響データと各段階の推定値

### 6.5.3　推定結果の後処理

本章の推定結果についても、第 3 章、第 4 章と同様の後処理を施す。図 6.6 に各段階の推定値の例を示す。

## 6.6　　実験結果

実験にあたり、モデルのパラメータを表 6.1 のように設定した。後処理として、第 4 章、第 5 章と同様に rank order filter による平滑化処理およびレベル閾値、継続フレーム数閾値による枝刈り処理を行った。後処理のパラメータは表 6.2 に示すものを採用した。

図 6.7 は後処理前の OSGPDM 出力における打鍵時のベロシティ値正解値と推定値の対応を示したものである。学習データは学習用楽曲の全フレームを 1 フレームずつ与えた。推定値は各音高における打鍵タイミング（オンセット）前後 5 フレーム中の最大値を取り出した。RMSE はガウス過程回帰の場合と同様に(4.4.1)式より求めた。ガウス過程回帰，SGPLVM の場合と同様、回帰の結果は演奏時の打鍵の強さを反映したものであることが分かる。推定値が 0 前後となっているも

表 6.1　OSGPDM による自動採譜アルゴリズムのパラメータ

| Parameter | | Value | Parameter | | Value |
|---|---|---|---|---|---|
| $\mathbf{A}_\mu$ | $\theta_{\mathrm{x}\mu1}$ | 1.0 | $\mathbf{C}_\mu$ | $\theta_{\mathrm{z}\mu1}$ | 1.0 |
| | $\theta_{\mathrm{x}\mu2}$ | 0.025 | | $\theta_{\mathrm{z}\mu2}$ | 0.025 |
| $\mathbf{A}_\sigma$ | $\theta_{\mathrm{x}\sigma1}$ | 1.0 | $\mathbf{C}_e$ | $\theta_{\mathrm{ze}1}$ | 1.0 |
| | $\theta_{\mathrm{x}\sigma2}$ | 0.025 | | $\theta_{\mathrm{ze}2}$ | 0.025 |
| $\mathbf{B}_\mu$ | $\theta_{\mathrm{y}\mu1}$ | 1.0 | $c_0$ | | 0.02 |
| | $\theta_{\mathrm{y}\mu2}$ | 0.025 | $m$ | | 1000 |
| $\mathbf{B}_e$ | $\theta_{\mathrm{ye}1}$ | 1.0 | | | |
| | $\theta_{\mathrm{ye}2}$ | 0.025 | | | |

表 6.2 OSGPDM による自動採譜アルゴリズムの後処理のパラメータ

| | |
|---|---|
| Start point $L_1$ of rank order filter (Eq.(4.3.4)) | 5 |
| End point $L_2$ of rank order filter (Eq.(4.3.4)) | 12 |
| Order $s$ of rank order filter (Eq.(4.3.4)) | 8 |
| Level threshold $h$ of pruning | 17 |
| Duration threshold $\tau$ of pruning | 9 |

のについてはガウス過程回帰，SGPLVM の場合よりも大幅に減少しているが、これらは後述するように低音域および高音域における推定精度が SGPLVM と比べてもさらに改善されたためと考えられる。ただし、全体の RMSE は 65.34 とガウス過程回帰の場合よりも改善されているが、SGPLVM よりも大きくなっている。

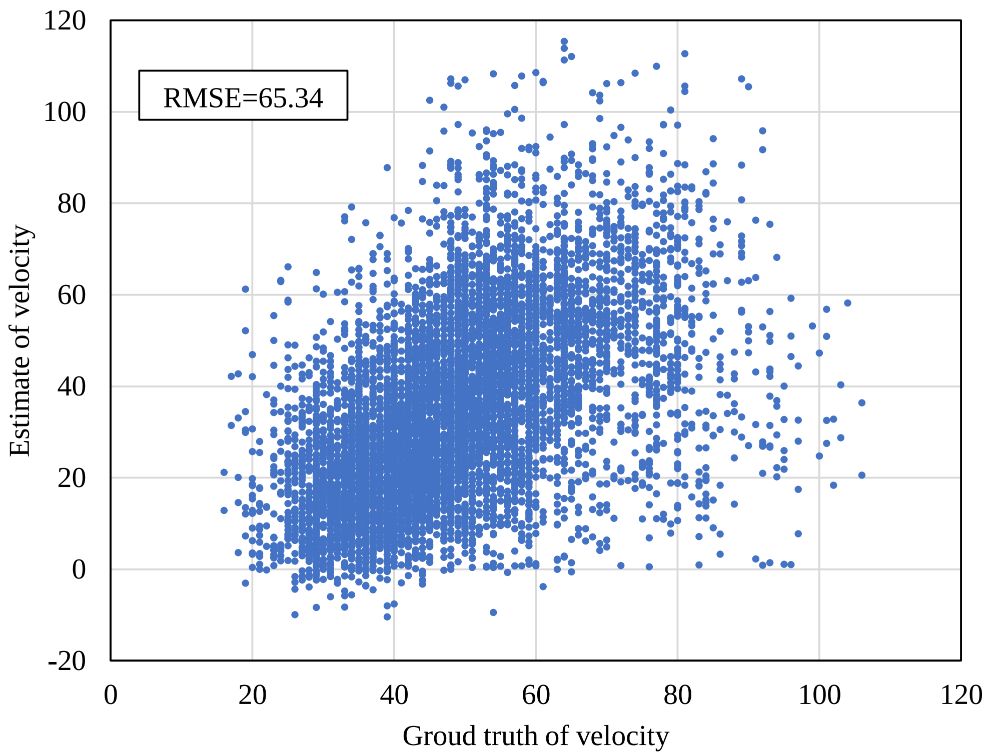
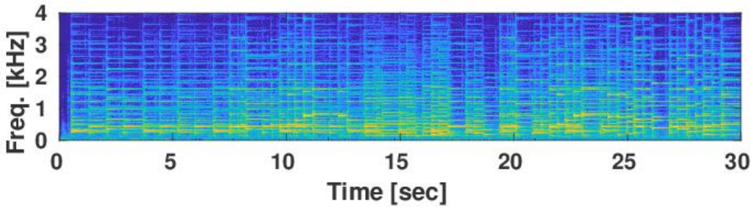
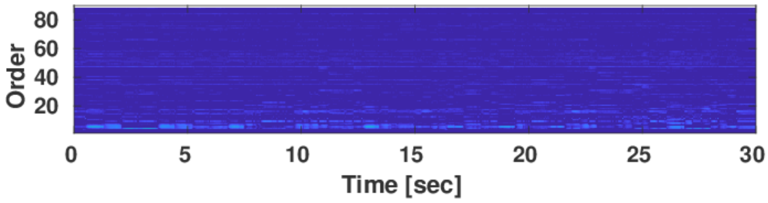


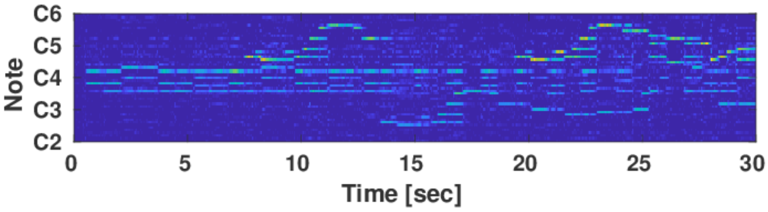

図 6.7　OSGPDM における正解データと推定値



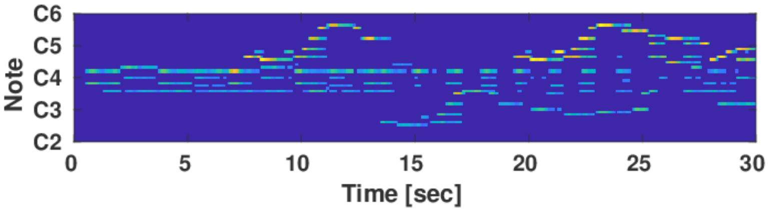

(a)音響データ



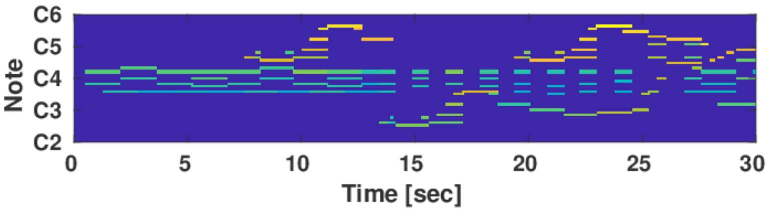

(b)潜在変数推定値




(c)演奏データ推定値

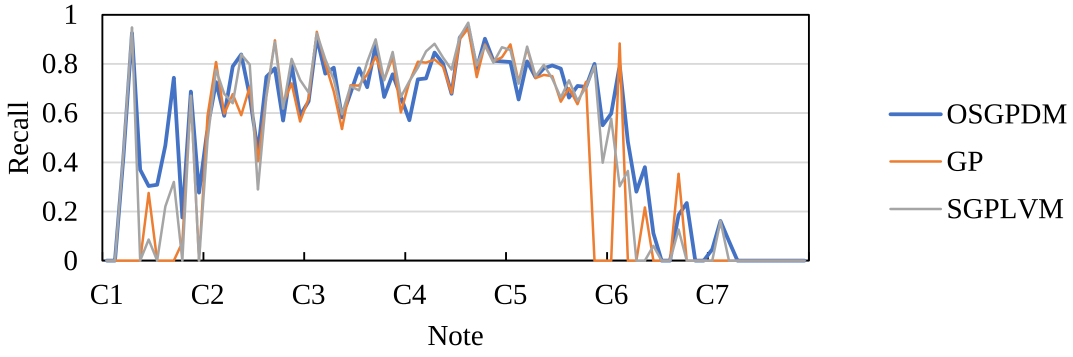



(d)後処理後の演奏データ推定値

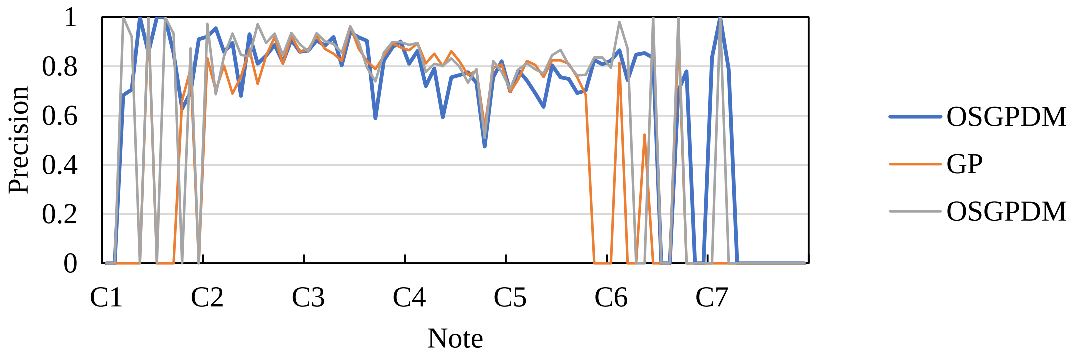



(e)正解データ

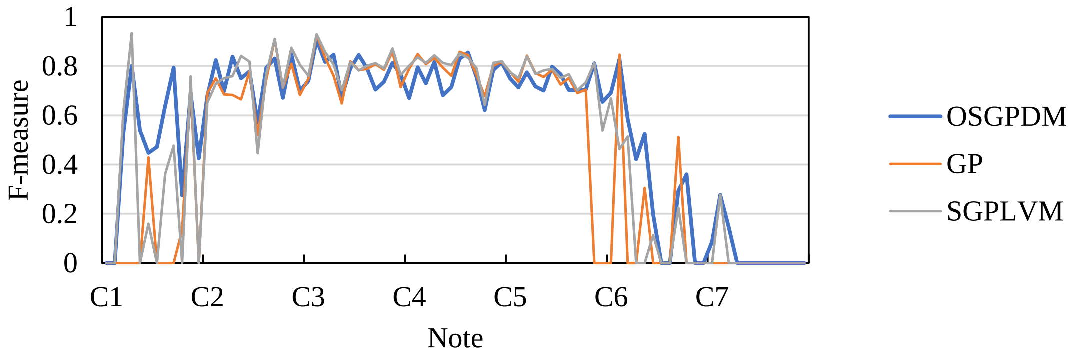
図 6.8 テスト用楽曲に対する推定例




(a)再現率




(b)適合率




(c)F 値

図 6.9　OSGPDM 自動採譜アルゴリズムの音高別推定精度

図 6.8 にテストデータの楽曲に対する推定例を示す。演奏データ推定値（図 6.8(c)）は演奏データ正解値（図 6.8(e)）と近いもとのなっていることが分かる。このことは、OSGPDM の学習によって音響データ、演奏データが共有する状態変数を獲得することができ、さらに未知の音響データに対応する状態変数の推定を介し、演奏データを推定することが可能であることを示している。演奏データ推定値に後処理を施したものが図 6.8(d)である。演奏データ推定値に残っていた微小な変動が除去され、押鍵の有無がより明確になっている。

図 6.9 に OSGPDM 自動採譜アルゴリズムの音高別の推定精度を示す。また 4 章，5 章との比較のため、ガウス過程回帰，SGPLVM による推定結果も併せて示す。OSGPDM 自動採譜アルゴリズムの全音域を通じての推定結果は再現率 0.742、精度 0.755、F 値は 0.748 であり、ガウス過程回帰自動採譜アルゴリズム，SGPLVM 自動採譜アルゴリズムの推定結果の精度を超えるには至らなかった。図 6.8 より、低音域（C2 以下）および高音域（A#5 以上）での推定精度が SGPLVM よりもさらに向上したことが示されているが、中音域での推定精度が若干低下している。図 4.10 に示した通り、学習用楽曲，テスト用楽曲ともに低音域および高音域の音高の出現頻度は低く、中音域の音高の出現頻度は高い。そのために OSGPDM では全体の推定精度が向上しなかったものと考えられる。

表 6.3 に本研究と同じテスト用楽曲を使用した先行研究における採譜結果とともに、本章で提案した手法による採譜結果を示す。

### 6.7　まとめと考察

本章では、SGPLVM で導入した潜在変数に時間的な依存性を持たせて非線形状態空間モデルを構成し、SRCKF による状態推定を行いながら MSOGP によってシステムモデル、観測モデルをオンライン学習する OSGPDM を提案した。音の時間的連続性を考慮し、オンライン学習により大量の音楽データを処理する自動採譜アルゴリズムを実現した。本章の目的は、以下の二点であった。

表 6.3　先行研究および本研究での採譜結果（フレームレベル）

| System | Precision | Recall | F-measure |
|---|---|---|---|
| Benetos and Weyde (2013) | - | - | 0.680 |
| Vincent et al. (2010)* | 0.796 | 0.636 | 0.707 |
| O'Hanlon and Plumbley (2014) | 0.755 | 0.705 | 0.729 |
| Berg-Kirkpatrick et al. (2014) | 0.691 | 0.807 | 0.744 |
| Cheng et al. (2015) | 0.854 | 0.729 | 0.777 |
| Cheng et al. (2016) | - | - | **0.790** |
| Gaussian Process Regression (Chapter 4) | 0.716 | 0.810 | 0.760 |
| SGPLVM (Chapter 5) | 0.751 | 0.810 | 0.780 |
| OSGPDM (Chapter 6) | 0.697 | 0.793 | 0.742 |

* 原文ではテスト用楽曲の構成が本研究とは異なるため、Berg-Kirkpatrick et al. (2014)で評価された結果を記載。

・潜在変数に時間的な依存性を導入して状態変数とし、自動採譜の問題を非線形状態空モデルにおける状態推定の問題として扱うことで、フレームごとに音響データから演奏データを推定する困難さを緩和し、推定精度の向上を図る。

・オンライン学習による大量の音響データ、演奏データの学習、および発生頻度の低い現象や楽曲中の出現頻度が低い音域の音高の効率的な学習によって、推定精度の向上を図る。

後処理前の OSGPDM の推定結果の評価では、ガウス過程回帰、SGPLVM の場合と同様、各音高のベロシティ値を反映した推定値が得られることを確認した。RSME はガウス過程回帰の場合よりも改善されたが、SGPLVM の場合よりは大きな値となった。さらに rank order filter による平滑化処理およびレベル閾値、継続フレーム数閾値による枝刈り処理を行って後処理を施すことで、自動採譜アルゴリズムを構築した。全音域を通じた F 値は 0.748 となり、ガウス過程回帰および

SGPLVM を超えるには至らなかった。図 6.9 に示した音高別の推定結果を見ると、C2 以下の低音域および A#5 以上の高音域で推定精度の大幅な改善が見られるが、中音域についてはガウス過程回帰、SGPLVM よりも若干推定精度が低下している。出現頻度の低い音域については OSGPDM の有効性が示されたが、出現頻度の高い中音域で推定精度が低下したことが全音域の推定精度を下げてしまったと考えられる。

表 6.3 に記載した先行研究との比較では、中位の Berg-Kirkpatrick et al. (2014) の推定結果とほぼ同程度の推定性能となった。

現時点では OSGPDM のモデルのパラメータをオンラインで調整する方法が確立していないため、各パラメータはグリッドサーチにより求めている。そのため、モデルの性能を最適化できていないことが採譜性能を落としている原因であると考えられ、部分的には本手法の有効性が確認できたものの、本章の目的を十分に達成するには至っていない。しかし低音域、高音域での推定精度の改善が見られたことから、パラメータの最適化によってさらなる推定精度の向上が期待できる。モデルパラメータについての最適化手法の確立が今後の課題である。

なお近年、ガウス過程回帰モデルを複数連結してより高度な推定を行う Deep Gaussian Process が提案されている（Damianou and Lawrence（2013），Wang et. al.（2016））。ガウス過程回帰モデルを連結するという点で本手法と Deep Gaussian Process との類似点も見受けられる。今後、Deep Gaussian Process で得られた知見も参考にして本手法の改善を図りたい。

# 第 7 章　結論

本章ではこれまでの議論を踏まえて全体のまとめと考察を行い、さらに今後の課題を述べて、本論文の結論とする。

## 7.1　全体のまとめと考察

第 3 章で問題提起したように、本研究では自動採譜の問題において押鍵の有無を 2 値識別として推定するのではなく、演奏の表情についての情報も抽出できるよう、打鍵の強さ（ベロシティ値）を連続値として推定する手法の確立に取り組んだ。第 4 章では音響データから演奏データのベロシティ値を推定するモデルをガウス過程回帰により実現し、1 出力の回帰モデルを音高ごとに置くことで、各音高のベロシティ値を反映した推定値が得られることを確認した。第 5 章、第 6 章ではそれぞれ SGPLVM、OSGPDM による自動採譜アルゴリズムを提案し、推定値の RMSE を改善することができた。ただし現状では回帰の推定結果のばらつきが大きく、このばらつきの縮小が今後の課題と考えられる。演奏データであるベロシティ値は打鍵時から離鍵時まで一定値をとるのに対し、音響データである振幅スペクトルは打鍵の直後から減衰する信号である。一定のベロシティ値に対し、減衰途中の様々な大きさの振幅スペクトルが対応するといった状況で学習が行われることが、ばらつきが発生する原因であると考えられる。

第 5 章では音響データと演奏データが持つ情報を共通のより低次元な潜在変数に縮約し、推定精度の向上を図ることを目的に SGPLVM による自動採譜アルゴリズムを提案した。潜在変数次数の評価結果では、潜在変数の次元がピアノの鍵盤数と同じ 88 次元の場合に最も高い推定精度を示すことを確認した。また学習データのフレーム数の評価では、学習フレーム数を 3000 フレームとした時点で、5000 フレームの学習データを用いたガウス過程回帰の推定結果を上回り、さらに出現頻度の低い低音域および高音域について推定精度の改善が見られた。これらの点から、SGPLVM が少ない学習データで効率的に学習を行えることが示された。rank order filter による平滑化処理とレベル閾値、継続フレーム数閾値による枝刈り処理の組み合わせによる後処理後の F 値は 0.780 となって、ガウス過程回帰による手法の 0.760 を上回り、観測変数の共有、および潜在変数の導入についての有効性が示された。

第 6 章では自動採譜の問題を時系列における推定問題として扱い、さらにオンラインで多様なデータを大量に学習可能とすることを目的に OSGPDM による自動採譜アルゴリズムを提案した。後処理後の推定精度評価では、C2 以下の低音域および A#5 以上の高音域で推定精度の大幅な改善が見られ、時間的な情報の利用、およびオンライン学習の有効性が示された。しかし、出現頻度の高い中音域で推定精度が低下したことから全音域での F 値は 0.748 に留まった。現時点では OSGPDM のモデルのパラメータをオンラインで調整する方法が確立していないが、低音域、高音域での推定精度の改善が見られたことから、パラメータの最適化によってさらなる推定精度の向上が期待できるものと考えられる。なお、ガウス過程回帰モデルを連結するという点で OSGPDM と Deep Gaussian Process（Damianou and Lawrence（2013）, Wang et. al.（2016））との類似点も見受けられる。今後、Deep Gaussian Process で得られた知見も参考にして本手法の改善を図りたい。

前述の通り、本研究で提案した手法の中では SGPLVM が最も高い推定精度を示しており、音響データ、演奏データに共通の潜在変数を導入することの有効性を示すことができた。OSGPDM については、現時点では SGPLVM を超える採譜性能には至っていないが、音楽データの時間的連続性を考慮し、かつオンラインで採譜アルゴリズムのモデルを更新することによって大量の音楽データを学習できる自動採譜アルゴリズム実現の可能性を示した。

先行研究との比較では最新のものにわずかに及ばなかったが、本研究での提案手法はいずれも上位に位置することができた。今回比較した事例の中で最も高い推定精度を示した Cheng et al.（2016）は、NMF と楽音の立ち上がりおよび減衰部分のモデルを組み合わせた手法であり、OSGPDM と同様に時間方向の情報を利用した手法と考えることができる。この例からも時間方向の情報を利用すること

の有効性が示されており、今後の OSGPDM の改善による推定精度の向上に期待を持たせるものであると考えられる。

なお近年、機械学習の分野で深層学習が注目を集めており、自動採譜のタスクにおいても深層学習を用いた手法が提案されている。実験の条件や用いたデータセットが本研究とは異なっていたため、論文中での推定精度の比較は行わなかったが、報告されている事例では、音響モデルと音楽言語モデルの組み合わせや（Sigtia et. al.（2015））、オンセット検出のような時間情報の利用（Wang and Yan（2017））を行うことで推定精度が向上している。深層学習を用いた事例においても先験的情報の利用が高い推定精度の実現と結びついていると考えられ、推定アルゴリズムの高度化のみならず、本研究で議論したように、どのような構造の中でいかに先験的情報を利用するかといった点が、今後の自動採譜の推定精度の向上とって重要であると考えられる。

## 7.2 今後の課題

本論文ではガウス過程回帰、SGPLVM、OSGPDM による採譜アルゴリズムを検討したが、前節でも触れたように現状では回帰の推定結果のばらつきが大きい。ばらつきが発生する原因として、演奏データであるベロシティ値は打鍵時から離鍵時まで一定値をとるのに対し、音響データである振幅スペクトルは打鍵の直後から減衰する信号であり、一定のベロシティ値に対して減衰途中の様々な大きさの振幅スペクトルが対応した形で学習が行われることが考えられる。OSGPDM におけるシステムモデルの改善等により、今後この問題に対処する必要がある。

OSGPDM については現時点ではモデルのパラメータを調整する方法が確立していないため、十分な採譜性能を引き出せていない。しかし本研究での実験や先行研究での知見から、音楽データの動特性を考慮することによって音の時間的連続性を考慮し、推定精度の向上を図ることは可能であると考えられる。モデルパラメータについての最適化手法の確立による採譜性能の向上が今後の課題である。

また本研究では主にフレームレベルの推定精度向上に注力してきたが、近年の

自動採譜の研究ではノートトラッキングあるいはオンセット検出の評価も行われている。参考として、推定値がレベル閾値 $h$ 未満から $h$ 以上の値へと変化した時点をオンセット（音符イベントの開始時点）とし、既出論文（Benetos and Weyde（2013）、Berg-Kirkpatrick et al.（2014）、Cheng et al.（2015）、Sigtia et al.（2015））と同様に、オンセット検出時点の前後 50msec 以内の範囲に正解データのオンセットが含まれれば正解として SGPLVM によるノートトラッキングの評価を行ったところ、F 値は 0.643 であった（精度：0.580、再現率：0.721）。例えば Berg-Kirkpatrick らの手法（Berg-Kirkpatrick et al.（2014））では 0.764 の F 値（精度：0.781、再現率：0.747）が得られており、ノートトラッキングの推定性能についてはまだ改善の余地があると考えている。本論文での提案手法は回帰モデルであり、推定値が連続値として得られるため、離散値としてオンセットの有無を推定するためには別途推定処理が必要となる。上述の手法では単純なレベル閾値によるオンセット検出を行ったため、特に同音連打時の検出性能が低下したが、今後は音響データや演奏データ、潜在変数の形状等も考慮し、検出性能の改善を図る必要があると考えられる。

なお、OSGPDM については自動採譜のタスクのみでなく、制御工学やデータ同化、マーケティングといった時系列を扱う種々の問題に応用できるものと考えられる。今後は自動採譜以外の分野に対しても適用を検討したい。

# 謝辞

本研究の遂行と博士論文の作成にあたり、主任指導教員である統計数理研究所の松井知子教授よりご指導をいただきました。社会人として仕事を持ちながらの研究であったために思うように捗らないことも多々ありましたが、丁寧な対応を長期に渡り行っていただきました。途中で投げ出さずに何とかここまで研究を進めることができましたのは、何よりも先生のご指導の賜物と考えております。心よりの感謝を表明いたします。

本研究を開始するにあたり、産業技術総合研究所情報技術研究部門の後藤真孝首席研究員からは貴重なご意見、ご指摘をいただきました。また音楽情報処理の最先端にいらっしゃる研究者と接点を持てたことは､何よりの刺激となりました。ここに感謝を申し上げます。

政策研究大学院大学の土谷隆教授には、先生が統計数理研究所にご在籍されていた当時、線形代数を叩き直していただくとともに、カルマンフィルタの理論等を丁寧に解説していただきました。先生からご指導いただいた内容は本研究の中でも生かされたと考えております。ここに深く御礼申し上げます。

本論文の審査は、統計数理研究所の福水健次教授、南和宏准教授、持橋大地准教授、奈良先端大学院大学の鹿野清宏名誉教授に引き受けていただきました。福水先生は博士課程在籍当時の副指導教員でもありました。先生方にはご多忙な中、審査のためのお時間を割いていただき、また論文の内容について貴重なご指摘を行っていただきました。これらのご指摘により、この度何とか博士論文の完成に辿り着けたものと考えております。心より感謝申し上げます。

著者が初めて自動採譜の研究に触れたのは 1990 年､大分大学工学部電子工学科（当時）での卒業研究でしたが、当時はこれ程長期に渡り関わることになるとは思ってもみませんでした。学部および修士課程での研究をご指導いただき、自動採譜の研究のきっかけを与えて下さいました大分大学工学部電子工学科（当時）の森田泰次教授（当時）にあらためて深く感謝を申し上げます。

社会人として仕事を持ちながら、業務内容とは異なる研究に従事するのは当初

の予想以上に大変であり、周囲の皆様のご協力なしには進めることができませんでした。公私に渡って様々なご協力を賜りました統計数理研究所の教職員の皆様、総合研究大学院大学複合科学研究科統計科学専攻でともに在学した当時の学生の皆様、前職場である花王株式会社川崎工場の皆様、現職場である同社情報システム部門の皆様に御礼申し上げます。

# 参考文献


Anderson, B. D., and Moore, J. B. (1979). Optimal filtering. Englewood Cliffs, 21, 22-95.

Arasaratnam, I., and Haykin, S. (2009). Cubature kalman filters. IEEE Transactions on automatic control, 54(6), 1254-1269.

Benetos, E., Dixon, S., Giannoulis, D., Kirchhoff, H., and Klapuri, A. (2013). Automatic music transcription: challenges and future directions. Journal of Intelligent Information Systems, 41(3), 407-434.

Benetos, E., and Weyde, T. (2013). Explicit duration hidden markov models for multiple-instrument polyphonic music transcription. In International Society for Information Music Retrieval

Berg-Kirkpatrick, T., Andreas, J., and Klein, D. (2014). Unsupervised transcription of piano music. In Advances in neural information processing systems. 1538-1546

Bertin, N., Badeau, R., and Vincent, E. (2010). Enforcing harmonicity and smoothness in Bayesian non-negative matrix factorization applied to polyphonic music transcription. IEEE Transactions on Audio, Speech, and Language Processing, 18(3), 538-549.

Bijl, H., van Wingerden, J. W., Schön, T. B., and Verhaegen, M. (2015). Online sparse Gaussian process regression using FITC and PITC approximations. IFAC-Papers On Line, 48(28), 703-708.

Bishop, C. M. (2008). パターン認識と機械学習（上,下）. シュプリンガー・ジャパン株式会社 (元田浩. 栗田多喜夫, 樋口知之, 松本裕治, 村田昇 (監訳))

Chang, C. C. LIBSVM -- A Library for Support Vector Machines. https://www.csie.ntu.edu.tw/~cjlin/libsvm/. (accessed 2010-10-2)

Cheng, T., Dixon, S., and Mauch, M. (2015). Improving piano note tracking by HMM smoothing. In Signal Processing Conference (EUSIPCO), 2015 23rd European (pp. 2009-2013). IEEE.

Cheng, T., Mauch, M., Benetos, E., and Dixon, S. (2016, August). An attack/decay model for piano transcription. ISMIR.

Cristianini, N., & Shawe-Taylor, J. (2005). サポートベクターマシン入門. 共立出版. (大北剛 (訳))

Csató, L., and Opper, M. (2002). Sparse on-line Gaussian processes. Neural computation, 14(3), 641-668.

Damianou, A., and Lawrence, N. (2013). Deep gaussian processes. In Artificial Intelligence and Statistics (pp. 207-215).

Dannenberg, R. B. (1984). An on-line algorithm for real-time accompaniment. In ICMC (Vol. 84, pp. 193-198).

Deena, S. P. (2012). Visual speech synthesis by learning joint probabilistic models of audio and video.

Ek, C. H., and Lawrence, P. H. T. N. D. (2009). Shared Gaussian process latent variable models (Doctoral dissertation, PhD thesis).

Ek, C. H., Torr, P. H., and Lawrence, N. D. (2007). Gaussian process latent variable models for human pose estimation. In International workshop on machine learning for multimodal interaction (pp. 132-143). Springer, Berlin, Heidelberg.

Emiya, V., Badeau, R., and David, B. (2010). Multipitch estimation of piano sounds using a new probabilistic spectral smoothness principle. IEEE Transactions on Audio, Speech, and Language Processing, 18(6), 1643-1654.

Evensen, G. (1994). Sequential data assimilation with a nonlinear quasi-geostrophic model using Monte Carlo methods to forecast error statistics. Journal of Geophysical Research: Oceans, 99(C5), 10143-10162.

Fujishima, T. (1999). Realtime Chord Recognition of Musical Sound: a System Using Common Lisp Music. In ICMC(pp. 464-467).

深山幸穂, 田中大介. (2009). 部分空間同定法を用いた採譜システムにおける楽器と調性の判別. 研究報告音楽情報科学 (MUS), 2009(8), 1-6.

Fukayama, Y., and Tanaka, D. (2013). A Music Transcription Algorithm Applying State Estimation and Parameter Identification on the Time-frequency Plane. Transactions of the Institute of Systems, Control and Information Engineers, 26(1), 1-7.

Goldberg, P. W., Williams, C. K., and Bishop, C. M. (1998). Regression with input-dependent noise: A Gaussian process treatment. In Advances in neural information processing systems(pp. 493-499).

Gordon, N. J., Salmond, D. J., and Smith, A. F. (1993). Novel approach to nonlinear/non-Gaussian Bayesian state estimation. In IEE Proceedings F (Radar and Signal Processing) (Vol. 140, No. 2, pp. 107-113). IET Digital Library.

Goto, M. (2001). An audio-based real-time beat tracking system for music with or without drum-sounds. Journal of New Music Research, 30(2), 159-171.

Grosche, P., Müller, M., and Kurth, F. (2010). Cyclic tempogram—a mid-level tempo representation for musicsignals. In Acoustics Speech and Signal Processing (ICASSP), 2010 IEEE International Conference on (pp. 5522-5525). IEEE.

Goto, M., and Hayamizu, S. (1999). A real-time music scene description system: Detecting melody and bass lines in audio signals. In Working Notes of the IJCAI-99 Workshop on Computational Auditory Scene Analysis (pp. 31-40).

Grochow, K., Martin, S. L., Hertzmann, A., and Popović, Z. (2004). Style-based inverse kinematics. In ACM transactions on graphics (TOG) (Vol. 23, No. 3, pp. 522-531). ACM.

樋口知之 (編), 上野玄太, 中野慎也, 中村和幸, 吉田亮 (著). (2011). データ同化入門: 次世代のシミュレーション技術. 朝倉書店

一般社団法人音楽電子事業協会. (2016). MIDI1.0 規格書 PDF 版. http://amei.or.jp/midistandardcommittee/MIDIspcj.html. (参照 2017-10-20)

今村武史, 松井知子. (2017). Shared Gaussian Process Latent Variable Model によるピアノ楽曲の自動採譜. 電子情報通信学会論文誌 D, 100(10), 882-891.

International Organization for Standardization. (1975). ISO 16:1975 Acoustics -- Standard tuning frequency (Standard musical pitch).

Julier, S., Uhlmann, J., and Durrant-Whyte, H. F. (2000). A new method for the nonlinear transformation of means and covariances in filters and estimators. IEEE Transactions on automatic control, 45(3), 477-482.

Kalman, R. E. (1960). A new approach to linear filtering and prediction problems. Journal of basic Engineering, 82(1), 35-45.

Kameoka, H. (2007). Statistical approach to multipitch analysis. Ph. D. Thesis, The University of Tokyo.

亀岡弘和, 中村友彦, 高宗典玄. (2015). 音楽音響信号処理技術の最先端. 電子情報通信学会誌, 98(6), 467-474.

亀岡弘和, 西本卓也, 嵯峨山茂樹. (2005). 調波時間構造化クラスタリング (HTC) による音楽音響特徴量の同時推定. 情報処理学会研究報告音楽情報科学 (MUS), 2005(82 (2005-MUS-061)), 71-78.

Kameoka, H., Nishimoto, T., and Sagayama, S. (2007). A multipitch analyzer based on harmonic temporal structured clustering. IEEE Transactions on Audio, Speech, and Language Processing, 15(3), 982-994.

Kameoka, H., Ochiai, K., Nakano, M., Tsuchiya, M., and Sagayama, S. (2012). Context-free 2D Tree Structure Model of Musical Notes for Bayesian Modeling of Polyphonic Spectrograms. In ISMIR (Vol. 2012, pp. 307-312).

亀岡弘和, 嵯峨山茂樹. (2009). 音楽情報処理技術の最前線: 1. 多重音解析と自動採譜. 情報処理, 50(8), 711-716.

Kashino, K. (1994). Computational auditory scene analysis for music signals. PhD thesis, University of Tokyo.

Kashino, K., Nakadai, K., Kinoshita, T., and Tanaka, H. (1995). Organization of hierarchical perceptual sounds. In Proceedings of the 14th Int. Joint Conf. on Artificial Intelligence (IJCAI-95) (Vol. 1, pp. 158-164).

片山徹. (2004). システム同定―部分空間法からのアプローチ―. 朝倉書店

河原英紀. (1991). ウェーブレット解析の聴覚研究への応用 (< 小特集> 新しい信号処理の理論とその応用: ウェーブレット解析とその周辺). 日本音響学会誌, 47(6), 424-429.

Kitagawa, G. (1996). Monte Carlo filter and smoother for non-Gaussian nonlinear state space models. Journal of computational and graphical statistics, 5(1), 1-25.

北川源四郎. (2005). 時系列解析入門. 岩波書店

Klapuri, A. P. (2003). Multiple fundamental frequency estimation based on harmonicity and spectral smoothness. IEEE Transactions on Speech and Audio Processing, 11(6), 804-816.

Krueger, B. Classical Piano Midi Page. http://www.piano-midi.de/.(accessed 2008-7-31)

Lawrence, N. (2005). Probabilistic non-linear principal component analysis with Gaussian process latent variable models. Journal of machine learning research, 6(Nov), 1783-1816.

Lawrence, N. D. (2007). Learning for larger datasets with the Gaussian process latent variable model. In Artificial Intelligence and Statistics (pp. 243-250).

Lawrence,N. D. Shared GP-LVM Softwar ,
http://inverseprobability.com/sgplvm/, (accessed 2015-4-18)

Lawrence, N. D., and Quiñonero-Candela, J. (2006). Local distance preservation in the GP-LVM through back constraints. In Proceedings of the 23rd international conference on Machine learning (pp. 513-520). ACM.

Lee, D. D., and Seung, H. S. (2001). Algorithms for non-negative matrix factorization. In Advances in neural information processing systems (pp. 556-562).

Moorer, J. A. (1975). On the segmentation and analysis of continuous musical sound by digital computer. Ph.D. dissertation, Stanford Department of Music Report No. STAN-M3

棟安実治，田口亮. (1999). 非線形ディジタル信号処理，朝倉書店

Nakamura, E., Hamanaka, M., Hirata, K., and Yoshii, K. (2016). Tree-structured probabilistic model of monophonic written music based on the generative theory of tonal music. In Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on (pp. 276-280). IEEE.

Nakano, S., Ueno, G., and Higuchi, T. (2007). Merging particle filter for sequential data assimilation. Nonlinear Processes in Geophysics, 14(4), 395-408.

O'Hanlon, K., and Plumbley, M. D. (2014). Polyphonic piano transcription using non-negative matrix factorisation with group sparsity. In Acoustics, speech and signal processing (icassp), 2014 ieee international conference on (pp. 3112-3116). IEEE.

Paulus, J., Müller, M., and Klapuri, A. (2010). State of the Art Report: Audio-Based Music Structure Analysis. In ISMIR (pp. 625-636).

Pertusa, A., and Inesta, J. M. (2008). Multiple fundamental frequency estimation using Gaussian smoothness. In Acoustics, Speech and Signal Processing, 2008. ICASSP 2008. IEEE International Conference on (pp. 105-108). IEEE.

Piszczalski, M., and Galler, B. A. (1979). Predicting musical pitch from component frequency ratios. The Journal of the Acoustical Society of America, 66(3), 710-720.

Poliner, G. E., and Ellis, D. P. (2007). A discriminative model for polyphonic piano transcription. EURASIP Journal on Applied Signal Processing, 2007(1), 154-162.

Quiñonero-Candela, J., and Rasmussen, C. E. (2005). A unifying view of sparse approximate Gaussian process regression. Journal of Machine Learning Research, 6(Dec), 1939-1959.

Rasmussen, C. E., and Williams, C. K. (2006). Gaussian processes for machine learning. Cambridge: MIT press.

Ryynanen, M. P., and Klapuri, A. (2005). Polyphonic music transcription using note event modeling. In Applications of Signal Processing to Audio and Acoustics, 2005. IEEE Workshop on (pp. 319-322). IEEE.

斉藤収三, 中田和男.(1981). 音声情報処理の基礎. オーム社

Scholkopf, B., and Smola, A. J. (2001). Learning with kernels: support vector machines, regularization, optimization, and beyond. MIT press.

Sheh, A., and Ellis, D. P. (2003). Chord segmentation and recognition using EM-trained hidden Markov models.

Shon, A., Grochow, K., Hertzmann, A., and Rao, R. P. (2006). Learning shared latent structure for image synthesis and robotic imitation. In Advances in neural information processing systems. (pp. 1233-1240).

Sigtia, S., Benetos, E., and Dixon, S. (2015). An end-to-end neural network for polyphonic music transcription. arXiv:1508.01774

Smaragdis, P., and Brown, J. C. (2003). Non-negative matrix factorization for polyphonic music transcription. In Applications of Signal Processing to Audio and Acoustics, 2003 IEEE Workshop on. (pp. 177-180). IEEE.

高宗典玄, 亀岡弘和, 嵯峨山茂樹. (2014). 2 次元 LR パーサによる音楽演奏 MIDI 信号からの自動採譜. 日本音響学会研究発表会講演論文集 日本音響学会 編, 1039-1042.

武田晴登, 西本卓也, 嵯峨山茂樹. (2004). 確率モデルによる多声音楽演奏の MIDI 信号のリズム認識. 情報処理学会論文誌, 45(3), 670-679.

Tipping, M. E., and Bishop, C. M. (1999). Probabilistic principal component analysis. Journal of the Royal Statistical Society: Series B (Statistical Methodology), 61(3), 611-622.

Van Vaerenbergh, S., Lázaro-Gredilla, M., and Santamaría, I. (2012). Kernel recursive least-squares tracker for time-varying regression. IEEE transactions on neural networks and learning systems, 23(8), 1313-1326.

Vapnik, V. N. (1995). The Nature of Statistical Learning Theory, Springer, New York

Vercoe, B. (1984). The synthetic performer in the context of live performance. In Proc. ICMC (pp. 199-200).

Vincent, E., Bertin, N., and Badeau, R. (2010). Adaptive harmonic spectral decomposition for multiple pitch estimation. IEEE Transactions on Audio, Speech, and Language Processing, 18(3), 528-537.

Wakefield, G. H. (1999). Mathematical representation of joint time-chroma distributions. In International Symposium on Optical Science, Engineering, and Instrumentation, SPIE (Vol. 99, pp. 18-23).

Wang, J., Hertzmann, A., and Blei, D. M. (2006). Gaussian process dynamical models. In Advances in neural information processing systems (pp. 1441-1448).

Wang, Q., Zhou, R., and Yan, Y. (2017). A two-stage approach to note-level transcription of a specific piano. Applied Sciences, 7(9), 901.

Wang, Y., Brubaker, M., Chaib-Draa, B., and Urtasun, R. (2016). Sequential inference for deep Gaussian process. In Artificial Intelligence and Statistics (pp. 694-703).

Yeh, C., (2008). Multiple fundamental frequency estimation of polyphonic recordings (Doctoral dissertation, Ph. D. dissertation, University Paris 6).

吉井和佳. (2016). 音楽を軸に拡がる情報科学: 5. 音楽と機械学習. 情報処理, 57(6), 519-522.

Yoshii, K., and Goto, M. (2011). A Vocabulary-Free Infinity-Gram Model for Nonparametric Bayesian Chord Progression Analysis. In ISMIR (pp. 645-650).

吉井和佳, 糸山克寿. (2015). 1. 統計的音響信号処理の新展開 (< 特集> メディア処理のための機械学習~ ビッグデータ活用を支えるキーテクノロジー~). 映像情報メディア学会誌: 映像情報メディア, 69(2), 111-116.