Vol.2018-MUS-120 No.6
2018/8/22

敵対的生成ネットワークを用いた楽曲の自動コード推定法の検討

納庄 貴大[1,a)]　西村 竜一[1,b)]　入野 俊夫[1,c)]

概要：自動コード推定問題で推定精度を上げることは重要な研究課題となっている。先行研究では、隠れマルコフモデルやニューラルネットワーク等を用いた手法が提案されている。これらの手法では、トライアドのうち「メジャー（maj)、マイナー（min)」コードの推定精度は高い。それに比べて、その他のトライアドである「オーギュメント（aug)、ディミニッシュ（dim)」コード及びセブンスコード等の推定精度は低いという問題点がある。原因としては、(a) 和声外音の存在等の音楽的要因、(b) 訓練データの不足等の機械学習手法に起因する問題の二つが関係すると考えられる。そこで本研究では、(b) 訓練データの不足問題の解決を目指し、敵対的生成ネットワーク（GAN）を用いた自動コード推定法を提案する。提案手法では、生成ネットワーク（Generator）により、MIDI の演奏情報から識別器の学習に利用するための追加の訓練データを生成する。その上で、訓練データと生成データの両方を用いて、識別ネットワーク（Discriminator）をクラス分類の識別器として学習する。

1. はじめに

音楽音響信号の自動コード推定は、音楽情報検索分野における重要な研究課題である [1]。これは自動コード推定問題で推定精度を上げることが、自動採譜 [2]、類似楽曲のクラスタリング [3] 、カバー曲同定 [4] 等の応用の性能向上に繋がるためである。

自動コード推定の処理は、音楽音響信号をフレーム単位またはビート単位で切り分けた区間から音響特徴量を抽出する特徴抽出部と、抽出した特徴量を元に識別モデル等を用いて適切なコードを推定する識別部から構成される。特徴抽出部では、音楽音響信号からコード推定のための代表的な特徴量であるクロマベクトル [5] を直接抽出するか、音楽音響信号を調波音や打楽器音に分離した後で特徴量を抽出することが多い [6] [7]。識別部では、生成モデルの隠れマルコフモデル（Hidden Marcov Model, HMM）を用いることが多い [8] [9]。また、近年ではニューラルネットワークを用いて特徴量抽出から識別までを End-to-End で処理する CNN-CRF [10] や BLSTM-CRF[11] といったモデルが提案されている。

しかし、先行研究では、「トライアドのうちメジャー（maj)、マイナー（min）の推定精度は高いが、それに比べて、その他のトライアドであるオーギュメント（aug)、ディミニッシュ（dim）及びセブンス等の推定精度は低い」という問題点が存在する。原因としては、(a) 和声外音の存在等の音楽的要因、(b) 訓練データの不足等の機械学習手法に起因する問題の二つが関係すると考えられる。

このうち、(a) については、次のような要因が考えられる。

- 認識対象の増加による難しさ。認識対象について、トライアド、セブンス、テンションといったコードやその転回形等を考慮すると数百種類を超える。これはコード推定における大語彙問題と呼ばれる [12]。
- 経過音、刺繍音、掛留音、倚音といった和声外音の存在。楽曲中で、その時刻のコードの構成音には無い音がメロデイとして演奏されることがあるため、コードの推定が難しくなる。
- コードの省略音の発生。三和音における 3rd 音の省略やコード区間の定義の異なりによるコードの捉え方の差異等が起こるため省略が生じる。

一方、(b) については、次のような要因が考えられる。

- コードにおけるヒエラルキーの存在。例えば、「C7、Cmaj7、C6」は C の構成音を全て含んだ上で異なる音が追加されており、C とヒエラルキーの関係があると考えることができる。一方「Ddim」には C の構成音は含まれておらず、C とヒエラルキーの関係がないと判断できる。しかし、機械学習手法を用いて分類する際に、「C と C7、C と Cmaj7、C と C6 の間違い」

[1] 和歌山大学 (Wakayama University)
[a)] s185035@center.wakayama-u.ac.jp
[b)] nisimura@sys.wakayama-u.ac.jp
[c)] irino@sys.wakayama-u.ac.jp

と「C と Ddim の間違い」は等しい損失として扱われるため、学習が適切に処理できないことが起こる。

- 様々なコードに対する訓練データの不足。現在、Isophonics [13] をはじめとして正解コードラベルを有するデータセットはいくつか存在するが、一般的に、多くの楽曲は三和音を中心に作られているため、四和音以上を有するデータセットは少ない状況にある。こうしたデータセットの作成には、音楽的な素養を持った複数の専門家によるラベル及びラベル区間の検証が必要である。ゆえに、画像認識用データセット作成のように、対象に対する「犬」、「猫」のようなシンプルなラベル付けを人手で行い、データセットを一度に多く増やすことが難しい。

そこで本研究では、(b) の訓練データが不足する状況下においても推定精度を確保することを目的に、敵対的生成ネットワーク（Generative Adversarial Nets, GAN）[14] を用いた自動コード推定法を提案する。

2. 提案手法

提案手法の概要を図 1 に示す。提案手法では、2.1 節で説明する GAN の生成ネットワーク（Generator）により、MIDI の演奏情報から識別器の学習に利用するための追加の訓練データを生成する。MIDI の情報は、音高×時間の 128 × 128 行列として扱い、Generator に入力できる。生成データは、2.2 節で説明する CNN-CRF モデルにおける前処理（2.3.1 節参照）の結果から得られる特徴量（105 × 15 行列）に相当する出力となる。その上で、訓練データと生成データの両方を用いて、識別ネットワーク（Discriminator）をクラス分類の識別器として学習する。

2.1 敵対的生成ネットワーク（GAN）

GAN は、深層ニューラルネットワークを用いた生成モデルである。GAN では、生成ネットワーク（Generator）と識別ネットワーク（Discriminator）の 2 つのモデルを相互に学習させる。その過程で、Generator はランダムノイズを入力として限りなく訓練データに近い生成データを生成する能力を高める。一方、Discriminator は訓練データと生成データを正しく識別する能力を高める。Generator と Discriminator には、任意のニューラルネットワークのモデルを利用することができる。

GAN の目的関数は式 (1) のように定義できる。

$$\min_G \max_D V(D,G) = \mathbb{E}_{x\sim p_{data(x)}}[logD(x)] + \mathbb{E}_{z\sim p_{z(z)}}[log(1-D(G(z)))] \quad (1)$$

ここで、G は Generator、D は Discriminator を指す。x は訓練データ、z はノイズ、$D(x)$ は x が訓練データである確率、$G(z)$ は z を入力として生成データを出力する関数、$x \sim p_{data(x)}$ は訓練データの確率分布に従うサンプル、

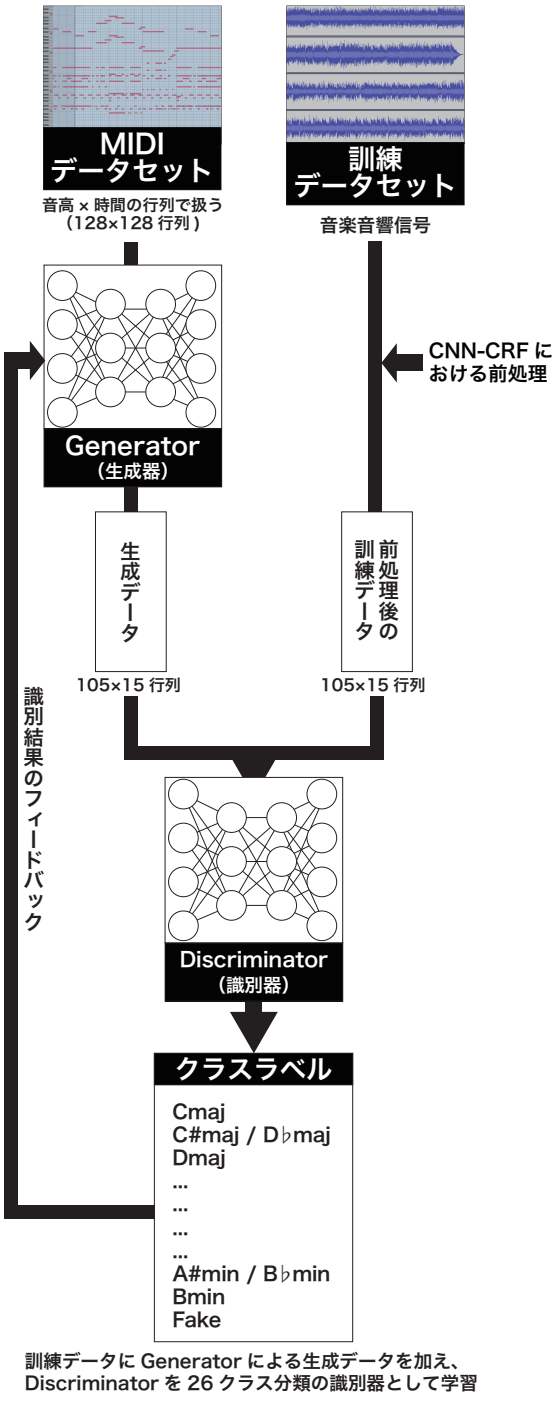

図 1 提案手法の概要図

$z \sim p_{z(z)}$ は生成データの確率分布に従うサンプルである。右辺の第 1 項 $\mathbb{E}_{x\sim p_{data(x)}}[logD(x)]$ は Discriminator が訓練データを訓練データであると正しく識別する期待値であり、第 2 項 $\mathbb{E}_{z\sim p_{z(z)}}[log(1-D(G(z)))]$ は生成データを生成データであると正しく識別する期待値である。Discriminator は訓練データと生成データを正しく識別するように学習を行うため式 (1) を最大化する。一方、Generator は Discriminator に誤識別させるように学習を行うため式 (1) を最小化する。

表 1 DCGAN の Generator（生成ネットワーク）の構造

層	層の種類	パラメータ数	パディング	出力サイズ
1	入力			128 × 128
2	恒等-ReLU			512 × 3 × 3
3	逆畳み込み-ReLU	256 × 2 × 2	有	256 × 4 × 4
4	逆畳み込み-ReLU	128 × 2 × 2	有	128 × 6 × 6
5	逆畳み込み-ReLU	64 × 2 × 2	有	64 × 10 × 10
6	逆畳み込み-シグモイド	1 × 8 × 8	有	105 × 15

表 2 DCGAN の Discriminator（識別ネットワーク）の構造

層	層の種類	パラメータ数	パディング	出力サイズ
1	入力			105 × 15
2	畳み込み-LReLU	64 × 3 × 3	有	64 × 10 × 10
3	畳み込み-LReLU	128 × 2 × 2	有	128 × 6 × 6
4	畳み込み-LReLU	256 × 2 × 2	有	256 × 4 × 4
5	畳み込み-LReLU	512 × 2 × 2	有	512 × 3 × 3
6	恒等			26

表 1 及び表 2 において、表の区切り線の部分で
Batch Normalization が適用される。

2.2 Deep Convolutional GAN（DCGAN）

DCGAN（Deep Convolutional GAN）[15] は、GAN の Generator と Discriminator に畳み込みニューラルネットワーク（Convolutional Neural Network, CNN）を導入したモデルである。GAN の学習時には、Mode Collapse[16] 等が生じる可能性がある。そのため、学習が不安定であるという問題点が存在する。これを受けて、DCGAN では次のような工夫により学習の安定性を高めている。

- CNN における最大プーリング層の替わりに、Generator では Fractionally-strided Convolution を用いてアップサンプリングを行う層、Discriminator では Strided Convolution を用いてダウンサンプリングを行う層を用いる。
- CNN における全結合層の替わりに、Global Average Pooling（GAP）層 [17] を用いる。
- Generator の出力層と Discriminator の入力層以外の層で、Batch Normalization [18] を適用する。
- Generator の出力層以外と Discriminator の全ての層で、活性化関数に Leaky ReLU 関数 $f(x) = max(\alpha x, x)\,(\alpha = 0.2)$ を用いる。

DCGAN の Generator の構造を表 1、Discriminator の構造を表 2 に示す。表 1 及び表 2 において、表の区切り線の部分で Batch Normalization が適用される。

2.3 CNN-CRF モデル

本稿では、F. Korzeniowski らによって提案された自動コード推定モデル [10] を CNN-CRF モデルと呼ぶこととする。CNN-CRF モデルは、3 つのコード推定処理（前処理部、特徴抽出部、和音複合部）から構成される。

2.3.1 前処理部

前処理部では、特徴量の抽出を行う。まず、入力音楽音響信号に対して短時間フーリエ変換（STFT）を適用する。

表 3 CNN-CRF モデルの CNN の構造

層	層の種類	パラメータ数	パディング	出力サイズ
1	入力			105 × 15
2	畳み込み-ReLU	32 × 3 × 3	有	32 × 105 × 15
3	畳み込み-ReLU	32 × 3 × 3	有	32 × 105 × 15
4	畳み込み-ReLU	32 × 3 × 3	有	32 × 105 × 15
5	畳み込み-ReLU	32 × 3 × 3	有	32 × 105 × 15
6	最大プーリング	2 × 1		32 × 52 × 15
7	畳み込み-ReLU	64 × 3 × 3	無	64 × 50 × 13
8	畳み込み-ReLU	64 × 3 × 3	無	64 × 48 × 11
9	最大プーリング	2 × 1		64 × 24 × 11
10	畳み込み-ReLU	128 × 12 × 9	無	128 × 13 × 3
11	畳み込み-恒等	26 × 1 × 1	無	26 × 13 × 3
12	平均プーリング	13 × 3		26 × 1 × 1
13	ソフトマックス			26

表 3 において、全ての畳み込み層の後に Batch Normalization、
表の区切り線の部分で確率 0.5 の Dropout が適用される。

この際、フレームサイズは 8192 サンプル（0.19 秒）、窓サイズは 8192 サンプル（0.19 秒）、フレームシフトは 4410 サンプル（0.1 秒）、窓関数は Hann 窓とする。次に、メルフィルタバンクを適用する。メルフィルタバンクは、65Hz（C2）から 2100Hz（C7）までの間で 1 オクターブあたり 24 チャンネルで構成される。最後に、対数変換を適用する。

これらの処理によって得られる音楽音響信号の時間周波数表現 L は、式 (2) のように定義できる。

$$L = \log\left(1 + B^{\triangle}_{Log}\,|S|\right) \tag{2}$$

ここで、S は音楽音響信号の STFT、$B^{\triangle}_{Log}$ はメルフィルタバンクを指す。

この L を系列に関する文脈情報と共に特徴抽出部に送る。具体的には、L の列 l_i を要素とする行列 $X_i = [l_{i-C}, ..., l_i, ..., l_{i+C}]$ を扱う。ここで、i は対象フレームのインデックス、C は文脈サイズである。本稿では、元論文 [10] と同様に $C = 7$ とした。ゆえに、X_i は対象フレーム i と前後 7 フレームを合わせた合計 15 フレームの前処理結果を有する $X_i \in \mathbb{R}^{105\times15}$ の行列となる。この X_i を特徴量として扱う。

2.3.2 特徴抽出部

特徴抽出部では、前処理部で抽出した特徴量を CNN に入力することで高次の特徴量を得る。CNN-CRF モデルにおける CNN の構造を表 3 に示す。表 3 では、全ての畳み込み層の後に Batch Normalization が適用される。加えて、表の区切り線の部分で Dropout [19] が確率 0.5 で適用される。最後の三層は、CNN において全結合層の変わりに用いられる GAP である。

2.3.3 和音複合部

和音複合部では、特徴抽出部で抽出した高次の特徴量を条件付き確率場（Linear-Chain Conditional Random Fields, CRF）[20] に入力することで、最終的なコード推定結果を得る。CRF は対数線形モデルを自然言語処理をはじめとした系列ラベリング問題に適用したモデル [21] であり、入

力系列を元に Viterbi アルゴリズム [22] により予測系列を出力する。CRF は式 (3) のように定義される。

$$P(Y|X) = \frac{\exp E(X,Y)}{\sum_{Y'} \exp(X,Y')} \tag{3}$$

ここで、Y はラベル系列 $\{y_0, ..., y_N\}$ であり、X は Y と同じ系列長の特徴量系列である。また、素性関数 $E(Y,X)$ は式 (4) のように定義される。

$$E(X,Y) = \sum_i \left(x_{i_{y_i}} + c_{y_{i-1}y_i} \right) \tag{4}$$

ここで、$x_{i_{y_i}}$ はクラス損失、$c_{y_{i-1}y_i}$ はラベル遷移コスト行列である。ラベル遷移コスト行列 $c_{y_{i-1}y_i}$ は、クラス分類におけるクラス数を K とすると $c_{y_{i-1}y_i} \in \mathbb{R}^{K \times K}$ の行列となり、CRF の学習時にはパラメータとして行列の各成分が調整される。さらに、損失関数は式 (5) のように定義される。

$$L = -\left(\sum_{i=1}^{l} x_{i_{y_i}} + \sum_{i=1}^{l-1} c_{y_{i-1}y_i} - log(Z) \right) \tag{5}$$

ここで、l は入力系列長、Z は正規化定数である。なお、素性関数と損失関数は Chainer の実装 [23] に従った。

3. 評価実験

提案手法のモデルを評価するために、次の 3 条件において、8 分割交差検証によるコード推定実験を行なった。

- 従来法（CNN-CRF モデル：モデルの構造は表 3 を使用）
- 提案手法 1（DCGAN 準拠のモデル：Generator は表 1、Discriminator は表 2 の構造を使用）
- 提案手法 2（DCGAN + CNN-CRF のモデル：Generator は表 1、Discriminator は表 3 の構造を使用）

3.1 実験条件

実験には、TheBeatles のアルバム 12 枚（Please Please Me, With the Beatles, A Hard Day's Night, Beatles for Sale, Help!, Rubber Soul, Revolver, Sgt. Pepper's Lonely Hearts Club Band, Magical Mystery Tour, The Beatles, Abbey Road, Let It Be）に含まれる楽曲 180 曲と、各楽曲に対応したコードの正解ラベルデータセット [13] 及び MIDI ファイル 180 個を用いた。楽曲のチャンネル数は 1ch、標本化周波数は 44.1kHz、量子化ビット数は 16bit とした。正解ラベルには、楽曲の各時刻における正解コードが記述されている。

提案手法のモデルの訓練時のエポック数は 15、ミニバッチサイズは 1024、最適化手法は Adam [24]（$\alpha = 0.0002$、$\beta\ 1 = 0.5$、$\beta\ 2 = 0.999$、$\varepsilon = 0.00000001$）、Generator の損失関数は生成データに対する予測結果の負の softplus 関数誤差、Discriminator の損失関数は訓練データと生成データに対する予測結果の交差エントロピー誤差とした。

表 4 Chord Symbol Recall (%)

	従来法 CNN-CRF	提案手法 1 DCGAN 準拠	提案手法 2 DCGAN + CNN-CRF
訓練データ	95.5	98.2	86.5
テストデータ	79.5	73.2	44.7

分類クラス数は、25 個のコードクラス（メジャー 12 個、マイナー 12 個、非コード 1 個）[25] に 1 個の生成データクラスを加えた 26 個とした。

コード推定の正解率は、全楽曲に対して推定結果が正解である区間の割合（Chord Symbol Recall, CSR [26]）とした。

3.2 実験結果

実験結果を表 4 に示す。訓練データに対する CSR については、提案手法 1 が従来法を上回り、提案手法 2 が従来法を下回ることを確認した。一方、テストデータに対する CSR については、提案手法 1 及び提案手法 2 が従来法を下回ることを確認した。

提案手法 1 に関する結果は、テストデータに対する CSR と訓練データに対する CSR に 25%の差があるため、過学習の傾向にあると考えられる。

また、提案手法 2 に関する結果から、従来法である CNN-CRF モデルに GAN の仕組みを導入したことで CSR が下がったことがわかった。これは、生成データと訓練データのパラメータ分布に乖離があることで、生成データが追加の訓練データとして機能しなかったことが原因であると考えられる。よって、3.3 節では生成データと訓練データのパラメータ分布の可視化を行うことで原因を考察する。

3.3 生成データの可視化

提案手法の Generator が出力する生成データ（105 × 15 行列）を可視化した。提案手法通り MIDI の演奏情報（128 × 128 行列）を入力とした時の生成データを図 2、提案手法で用いた MIDI の演奏情報の替わりにランダムノイズ（128 × 128 行列）を入力とした時の生成データを図 3 に示す。図 2 の生成データは、MIDI データセットから無作為に抽出した区間の演奏情報を元に出力されたものである。左図が 1 エポック終了時点、右図が 15 エポック終了時点での生成結果である。また、訓練データに CNN-CRF モデルの前処理を適用して得た特徴量を図 4 に示す。ここでは例として、左図に Emaj（E メジャー）、右図に Gmin（G マイナー）の区間を示した。他のどのコード区間でもこれらと同様の分布となった。図 2、3、4 における縦軸は行列の行、横軸は行列の列である。

MIDI の演奏情報を入力とした場合（図 2）とランダムノイズを入力とした場合（図 3）のどちらの結果も、濃淡

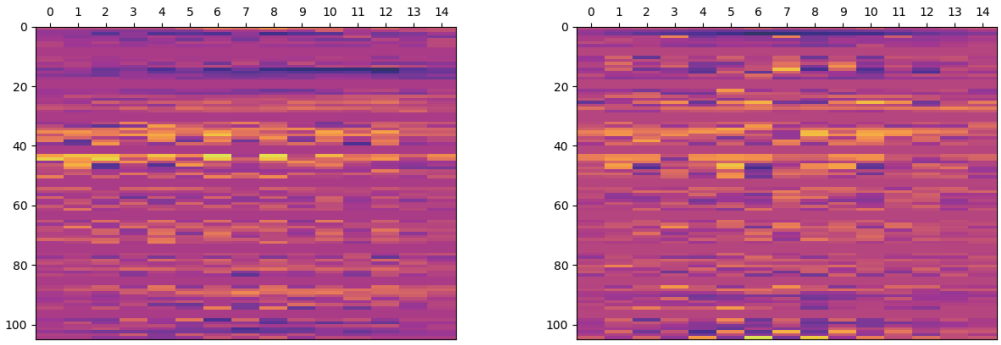

図 2 MIDI の演奏情報を提案手法 1（DCGAN 準拠のモデル）の Generator に入力した時の生成データ（105 × 15 行列）を濃淡画像として可視化した結果。MIDI データセットから無作為に抽出した区間の演奏情報を元に出力されたものである。縦軸が行、横軸が列。左図は 1 エポックの学習終了時点、右図は 15 エポックの学習終了時点の結果。

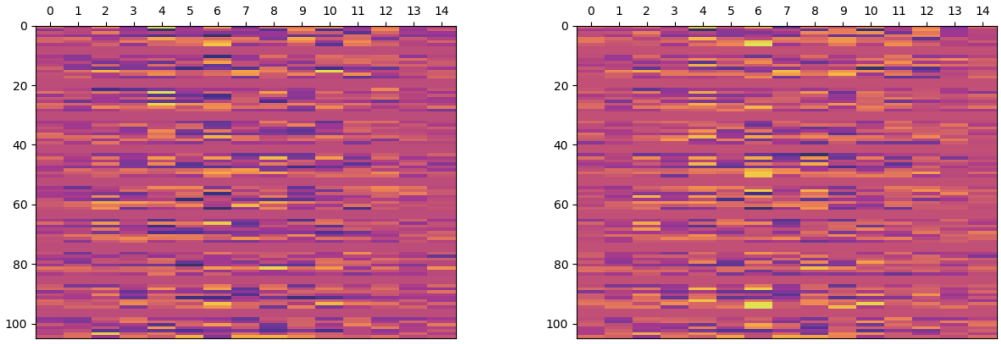

図 3 ランダムノイズを提案手法 1（DCGAN 準拠のモデル）の Generator に入力した時の生成データ（105 × 15 行列）を濃淡画像として可視化した結果。縦軸が行、横軸が列。左図は 1 エポックの学習終了時点、右図は 15 エポックの学習終了時点の結果。

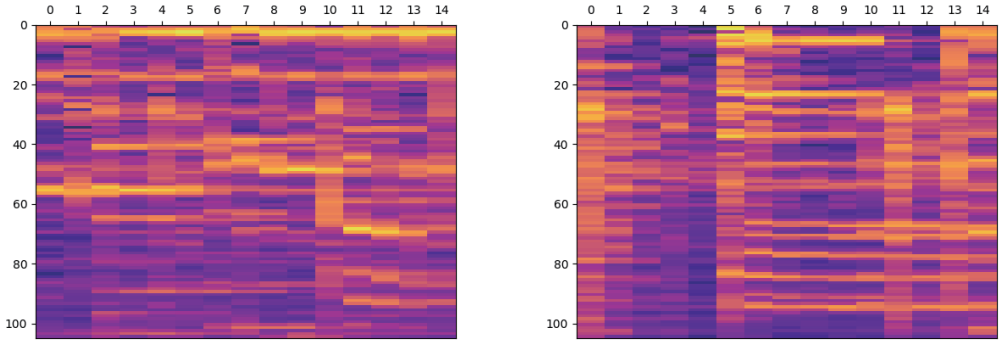

図 4 訓練データの音楽音響信号に 2.3.1 節で説明した CNN-CRF モデルの前処理を適用して得た特徴量（105 × 15 行列）を濃淡画像として可視化した結果。縦軸が行、横軸が列。ここでは例として、左図に Emaj 区間、右図に Gmin 区間に対する結果を示す。他のどのコード区間でもこれらと同様の分布となった。図 2、3 と比べて濃淡の差がはっきりしている。

の分布傾向が訓練データ（図 4）と異なることが確認できる。具体的には、生成データの濃淡分布は全体的に濃淡が均一な部分が多いが、訓練データの濃淡分布は濃い部分と淡い部分がはっきりしている。加えて、訓練データの濃淡分布には行方向に同程度の濃さが連続する傾向もある。このことから、Generator が訓練データのパラメータ分布に近い生成データを出力できていないことを確認した。

また、MIDI の演奏情報を入力とした場合（図 2）の濃淡分布には、訓練データの濃淡分布のように行方向に同程度の濃さが連続している箇所が存在するため、Generator への入力としてはランダムノイズよりも MIDI の演奏情報の方が適切であると考えられる。

4. おわりに

本稿では、敵対的生成ネットワークを用いた自動コード推定法を提案し、モデルの評価実験を行なった。今後は、生成データの可視化結果を踏まえて、Generator のモデル構造等を再検討する。また、現状はどのコードクラスに似たデータを生成するかの条件付けが無いため、Discriminator の学習に用いるにはデータが不完全であったり、1 つのコードクラスに対応したデータばかりを生成する Mode Collapse が起きる可能性がある。ゆえに、コードラベルに合わせて特定のコードクラスに対応した生成データを出力するように Generator を改良し、半教師あり学習を行う。その上で、推定対象コードの種類数を増やして同様の実験を行う予定である。

謝辞 本研究の一部は、JSPS 科研費 JP18K02862、JP16H01734 の助成を受けて実施した。

参考文献

[1] Brian McFee, Juan Pablo Bello, Structured Training for Large-vocabulary Chord Recognition, Proc. ISMIR, pp. 188–194, 2017.
[2] J. Weil, T. Sikora, J. L. Durrieu, G. Richard, Automatic Generation of Lead Sheets from Polyphonic Music Signals, Proc. ISMIR, pp. 603–608, 2009.
[3] 長澤槙子, 渡辺知恵美, 伊藤貴之, 定型コード進行パターンに着目したポピュラー音楽クラスタリング手法の提案, 情報処理学会研究報告, 2007-DBS-143-65, pp. 375–380, 2007.
[4] 中沢彰吾, 三河正彦, 田中和世, 伊藤慶明, 隠れマルコフモデルによる自動和音認識を用いたカバー演奏ストリームからの楽曲同定手法の検討, 電子情報通信学会技術研究報告, vol. 112, no. 358, HIP2012-59-72, pp. 1–6, 2012.
[5] T. Fujishima, Realtime Chord Recognition of Musical Sound: A System using Common Lisp Music, Proc. ICMC, pp. 464–467, 1999.
[6] 上田雄, 小野順貴, 嵯峨山茂樹, 機能和声モデルによる音楽信号からの和声推定, 情報処理学会研究報告, 2010-MUS-86-13, 2010.
[7] 丸尾智志, 池宮由楽, 糸山克寿, 吉井和佳, 音楽音響信号に対する歌声・伴奏音・打楽器音分離に基づくコード認識, 情報処理学会研究報告, 2015-MUS-108-1, 2015.
[8] A. Sheh, D. P. W. Ellis, Chord Segmentation and Recognition using EM-trained Hidden Markov Models, Proc. ISMIR, pp. 183–189, 2003.
[9] H. Papadopoulos, G. Peeters, Large-Scale Study of Chord Estimation Algorithms Based on Chroma Representation and HMM, Proc. International Workshop on Content-Based Multimedia Indexing, pp. 53–60, 2007.
[10] F. Korzeniowski, G. Widmer, A Fully Convolutional Deep Auditory Model for Musical Chord Recognition, Proc. IEEE International Workshop on Machine Learning for Signal, 2016.
[11] Yiming Wu, Wei Li, Music Chord Recognition Based on MIDI-Trained Deep Feature and BLSTM-CRF Hybird Decoding, Proc. ICASSP, pp. 376–380, 2018.
[12] Junqi Deng, Yu-Kwong Kwok, Large Vocabulary Automatic Chord Estimation using Bidirectional Long Short-Term Memory Recurrent Neural Network with Even Chance Training, Proc. ISMIR, pp. 531–536, 2017.
[13] M. Mauch, C. Cannam, M. Davies, S. Dixon, C. Harte, S. Kolozali, D. Tidhar, M. Sandler, OMRAS2 Metadata Project 2009, Proc. ISMIR, 2009.
[14] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozir, Aaron Courville, Yoshua Bengio, Generative Adversarial Nets, Proc. NIPS, pp. 2672–2680, 2014.
[15] A. Radford, L. Metz, S. Chintala, Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks, arXiv:1511.06434 [cs.LG], 2015.
[16] Ian J. Goodfellow, NIPS 2016 Tutorial: Generative Adversarial Networks, arXiv:1701.00160 [cs.LG], 2016.
[17] M. Lin, Q. Chen, S. Yan, Network in network, Proc. ICLR, 2014.
[18] S. Ioffe, C. Szegedy, Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift, Proc. ICML, pp. 448–456, 2015.
[19] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, R. Salakhutdinov, Dropout: A Simple Way to Prevent Neural Networks from Overfitting, Journal of Machine Learning Research, vol. 15, pp. 1929–1958, 2014.
[20] J. D. Lafferty, A. McCallum, F. C. N. Pereira, Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data, Proc. ICML, pp. 282–289, 2001.
[21] 奥村学, 高村大地, 言語処理のための機械学習入門, コロナ社, 2010.
[22] A. Viterbi, Error bounds for convolutional codes and an asymptotically optimum decoding algorithm, IEEE Transactions on Information Theory, vol. 13, no. 2, pp. 260–269, 1967.
[23] chainer.functions.crf1d — Chainer 4.2.0 documentation, `https://docs.chainer.org/en/stable/reference/generated/chainer.functions.crf1d.html#chainer.functions.crf1d`, (2018.07.30. アクセス確認)
[24] D. Kingma, J. Ba, Adam: A method for stochastic optimization, Proc. ICLR, 2014.
[25] M. McVicar, R. Santos-Rodriguez, Y. Ni, T. D. Bie, Automatic Chord Estimation from Audio: A Review of the State of the Art, IEEE/ACM Transactions on Audio, Speech, and Language Processing, vol. 22, no. 2, pp. 556–575, 2014.
[26] Harte Christopher, Towards Automatic Extraction of Harmony Information from Music Signals, Ph.D. diss. Queen Mary, University of London, 2010.