プログラミング基礎演習レポート 2023-24 独立成分分析による音声・画像・データの分離

東京大学 工学部 電子情報工学科 2年 J4-220215 瀧川 雄理

キーワード

直交行列による対角化、 線型変換、 音声分離、 合成画像の分離

I Introduction

本レポートにおいては、複数の元データを、異なる割合で合成して得た複数の新たなデータから、元データを復元することを目的とする。特に、数字としてのデータ列に限らず、音声や画像といった様々な形式のデータに対する適用方法を模索する。まずは、ICA そのものの手法について紹介し、その後、与えられたデータに対して適用した際の工夫や結果・考察をそれぞれ述べる。

II 手法 ICA __ util.ipynb における関数 ICA の説明

関数 ICA は、独立成分分析の手法に基づいて、大きく次の 3 つに分けられる。

① datanum 個の「datasize 時点のデータから成る時系列データ」の共分散行列 Σ の対角化をする過程で得られた、対角行列 D と直交行列 E を用いて、行列 $Z = ED^{-\frac{1}{2}}E^TX$ を定義する (白色化)。

② Z の各列ベクトルの線型変換により datadim 個の独立な成分 (列ベクトルとして並べた行列が Y) に分離する、その変換行列 W の各行ベクトルを、レポートの出題文で与えられたアルゴリズムに従って、更新する

③ 得られた独立な成分を matplotlib パッケージの pyplot ライブラリを用いて図示する

以下、ソースコード ICA_util.ipynb の流れに沿って、手法を説明する。

1. 行列 Z を求める

まず、各時系列データの平均を 0 にし、その後二次元配列の転置をとって、同時点におけるデータが同じベクトル $X[:, j]$ に格納されるようにする。次に、$datasize$ 時点それぞれについて、ベクトル $X[:, j]$ どうしの直積を計算し行列として展開し、その平均となる行列 Σ を共分散行列として得る。

一般的な、対称行列の直交行列による対角化関数 Eigenvalue_decomp を定義し、それを用いて、共分散行列を対角行列 D へと対角化する変換行列である、直交行列 E を得る。続いて、出題文の流れに沿って、行列 $Z = ED^{-\frac{1}{2}}E^TX$ を得る。この Z の各ベクトルは、その相関行列が対角行列つまり無相関になっている。

2. 独立成分への変換行列 W の更新

得られる $datadim$ 個の独立成分が全て互いに独立であるかを、成分間の相関係数の大きさで雑に判定し、全て互いに独立であると判定されるまで、W の初期値を生成し直し再度学習を行うアルゴリズムとした。

W の各ベクトルを、出題文中の繰り返しアルゴリズムにより更新していく。

ただし、更新の進み方によっては、途中から W が振動してしまいいつまでも収束しない[*1]ため、それを防ぐために、W の列ベクトルの第一成分が必ず正になるようにしている。

最終的には、$Y = WZ$ を計算し、Y をなすベクトルの一つ一つが独立な信号源を表しているので、それを図示し、行列 Y を返す。

[*1] 実際に、dat1.txt と dat2.txt を対象にすると振動が起こってしまう。

III　課題 1 合成波形の分離

ソースは、ICA1.ipynb にある。

1.　題材と ICA 以外の処理

dat1.txt と dat2.txt は 2 つの信号源を異なる比率で混合したデータだと考え、ICA によりその信号源を同定する。open(ファイル名) によりファイルを開いて、.readline() により、一行ずつデータを読み込んだ。$datanum = 2, datadim = 2$ の独立成分分析をおこなった。

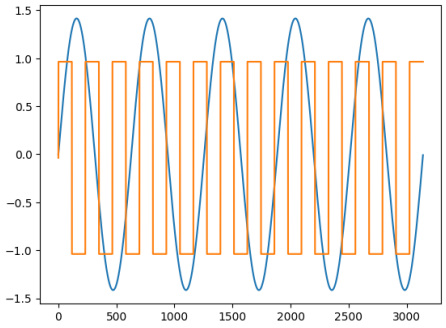
2.　結果

下右のような独立な 2 信号源が得られた。

各信号源の波形を見ると、方形波と正弦波という独立な 2 信号源からの信号を異なる比率で混合していたことが確認できる。

3.　考察 (および考えたこと)

はじめ、方形波と正弦波にきちんと分離できるときと、そうでないときがあったため、どういった理由があるか検討した。その結果、W の更新は、ベクトルごとに完全に独立に行われていたため、各ベクトルの更新の収束先が同じもしくはほとんど同じベクトルになる可能性が全く排除されていなかったことに気がついた。そこで、W についてすでに更新したベクトルと類似度が高いベクトルを更新の結果取ってしまった場合には、その行ベクトルの初期値を再度乱数により生成し直し、再び更新をやり直すように修正した。その結果、独立な成分どうしのみが出力されるようになった。



IV　課題 2 複数人の会話の分離

1.　題材と ICA 以外の処理

第一に、二人の話者が同時に話したものを，（仮想的に）異なる場所で録音したデータが入っている、speechA1.wav, speechA2.wav に対して、ICA によりその独立な信号源となる音声を同定し、分離する。ソースは、ICA2.ipynb にある。scipy.io.wavfile.read により、.wav ファイルを読み込んだ。その内容は、サンプリング周波数、データの一次元配列、データ型から構成される。まずは、各ファイルのデータの一次元配列を一つの配列にまとめる。次に、$datanum = 2, datadim = 2$ の ICA を実施し、その結果を元の音声データと同じスケールに復元した上で、キャスト (今回は int16 型) する。最後に、この波形を図示して、音声ファイル product_data/speechA1_ica.wav ならびに、product_data/speechA2_ica.wav に保存した。

第二に、二つの曲を同時に演奏したものを，（仮想的に）異なる場所で録音したデータが入っている、music1.wav, music2.wav に対して、ICA によりその独立な信号源となる音楽を同定し、分離する。ソースは、ICA2_music.ipynb にある。$datanum = 2, datadim = 2$ の ICA を実施し、その結果を元の音声データと同じスケールに復元した上で、キャスト (今回は int16 型) した。分離した音楽ファイルは、product_data/music1_ica.wav ならびに、product_data/music2_ica.wav に保存した。

2.　結果

二人の話者が同時に話した音声の分離については、およそ男声と女声でよく分離されていた。ただ、一部小さくお互いの声が混在している箇所が存在した。

3.　考察 (および考えたこと)

分離しきれていない箇所が存在した理由について、そもそも今回の分析は、二人の話者が話しているのを異なる場所で録音していた音声を対象にしていた。つまり、各信号源から観測点に到達する信号のスケールならびに比率は異なるという前提が存在していた。しかし、例えば、声量が小さいなどの理由により、各信号源から 2 観測点に到達する信号のスケールが全く変わらない時点については、そのデータを分離することは困難になりうる。また、今回白色化した Z を線形変換することで、独立な信号源 Y を同定しているわけであり、結局のところ、各信号源の時系列ベクトルは、Z の列ベクトルの極めて尤もらしい線形結合にすぎないため、分離には限界があ

るということだともいえる。また、全体的に子音が聞き取りづらくなっているような印象を受けたが、評価指標が存在しないので定量的な評価は不可能である。

V　課題 3 合成画像の分離

ソースコードは、ICA3.ipynb にある。

1. 題材と ICA 以外の処理

二つの画像の異なる比率の重ね合わせである、 image1.png と image2.png に対して、ICA を適用して、合成する前の 2 つの画像を同定し、分離する。

pillow ライブラリから Image クラスを利用し、png ファイルを読み込む。2 次元配列として読み込まれるため、この 2 次元配列の形状を、$horizontal, vertical$ として保存した上で、一次元配列に変形し、まとめて original_data に格納した。

その後 $datanum = 2, datadim = 2$ の ICA を適用し、その結果を元の画像データと同じスケールに復元した上で、キャスト (今回は unsigned int8 型) する。最後に、保存していた元の画像の形状情報をもとに配列を変形して、product_data/image1_re+.png、product_data/image2_re+.png に保存した。

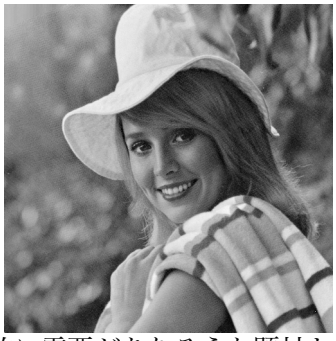
2. 結果

はじめ、ICA の結果を.astype により強引にキャストして得られた、信号源とおぼしき画像を見るに、輪郭は分離することができている反面、輪郭の内側の色調 (濃淡) に関しては、真っ黒に塗りつぶされてしまっていたりするなどしていた。(右の画像は product_data/image1_re.png)

しかし、キャストする前に、$Y+ = 128$ という式を追加して、ICA の冒頭で時系列データベクトルの平均が 0 になるように各ベクトルを平行移動させていたのを、もとに戻したところ、輪郭の内側の濃淡も自然になった。

3. 考察

現在はグレースケールの画像データであったが、RGB データなどの場合は、時系列データベクトルの数を 3 倍にすればよいと考えられるが、同じ時系列データの異なる色のベクトルどうしを無相関化することに意味はなくむしろ分析結果を狂わせることになると考えられるため、別の手法を考案する必要があるだろう。



VI　課題 4 複数人の足音の分離

課題 2 においては、複数人の声の分離を行った。ここでは、より困難だが、社会的に需要がありそうな題材として、「足音」を考えた。

1. 実験と ICA 以外の処理

まず、志賀高原「白樺 https://www.shirakaba.co.jp の地下室において、J5-220066 山本大智の協力のもと、部屋の対角に 1 台ずつ iPhone13 を配置し、一人がサンダルをもう一人がブーツを履いて 10 秒間歩く音を録音した。

その際、時間軸を揃えるために、録音の冒頭に、手を叩く音を入れてある。

ICA 以前の処理は、以下の手順で行った。

① Apple Music アプリケーション上において、MP4 ファイルを.wav 形式に変換

② jupyter notebook に scipy ライブラリを用いて読み込み、手を叩く音を基準に時間軸を揃えた。なお、手の叩く音の入っている「コマ」は、int16 型のデータの値が 20000 を超える「コマ」を抽出することで特定した。

③ 得た二つの音声データ data/footsteps1.wav, data/footsteps2.wavを一つの 2 次元 Array にまとめた。

その後、ICA にかけた。

2. 結果

- ICA における、信号源の独立性を判定するための、信号源ベクトルどうしの相関係数の閾値を 0.5 にしたところ、学習が終了しなかった。
- それを受けて、閾値を 0.8 に緩和したところ、学習は終了したが、得られた信号源は、全く足音を分離できていなかった。
- 再度、閾値を 0.6 に引き上げたところ、学習は終了し、得られた信号源 product_data/footstep1_ica.wav, product_data/footstep2_ica.wav については、部分的に足音が分離されていた。具体的には、product_data/footstep1_ica.wav では、6 秒手前までサンダルの足音のみになっており、それ以降はブーツの足音が支配的ではあるが 2 種類の足音が混在している。product_data/footstep2_ica.wav では、8 秒すぎまでブーツの足音のみになっている。

3. 考察

まず、足音というデータの特性として、声や音楽とは異なり、波形が鋭い δ 関数の重ね合わせのようになっていることが挙げられる。これが、独立成分の分析の妨げとなっている側面があると考えられる。また、音声の後半部分のようにどちらの録音データにおいても、同じ人の足音が支配的な場合、最適化関数においてもう一人の足音についての項の影響度が下がり、分離が困難であるという性質があると推測される。

参考文献

① なし