CONTENTS

サンプルの Experiment – ICA Demo	2
クイックノート	2
デモ(1D)の操作手順	3
デモ(2D)の操作手順	4
ICA のヘルプ	5

クイックノート

⋆ = - File \rightarrow Example Experiments \rightarrow Analysis \rightarrow ICA Demo

この Experiment では、2つまたは3つの1次元ウェーブから合成信号を作成します。 ウェーブは、正弦波または鋸歯波のいずれでもかまいません。 ウェーブの振幅と周期を設定できます。 デモでは、2つのウェーブを混合するためのものと、3つのウェーブを混合するためのもの、の2つの定数混合マト リックスを使っています。 混合信号は、PCA(主成分分析)とICA(独立成分分析)によって解析されます。 PCA とICA の成分がともに表示されます。 ICA 成分のスペクトル表現も表示されています。 ICA デモパネルを使って、さまざまなオプションを調整し、結果を確認してください。

このプロシージャは、2列または3列のウェーブ SS を作ることから始まります。 これは、Amp3 の左にあるチェックボックスの状態によって異なります。 ウェーブ SS は、次のいずれかで乗算されます。

AA[0][0]= {0.3,0.7,-0.5} AA[0][1]= {-0.5,3,0.5} AA[0][2]= {0.7,0.2,-1}

または

AA[0][0]= {1,-1} AA[0][1]= {1,1}

乗算式は次のとおりです:

MatrixOP/O xx=SS x AA

注記: 文献では、次のような表現がよく見られます。

MatrixOP/O Mixed=A x S

したがって、n×n 行列Aとn×m 行列Sの「混合」行列は、次元がn×m(通常n<m)となります。

Igor の自然なデータ配置は列であるため(例えば、EEG の記録では各チャンネルは2次元ウェーブの列で表されます)、ICA コマンドのデフォルトは、行数が列数よりも多い混合用に設計されています。

それ以外の場合は、混合ウェーブを転置するか、ICAの/Tフラグを使ってください。

「SVD failed for converge」というエラーメッセージが表示された場合は、行数が列数よりも少ない可能性が高いので注意してください。

3つのグラフの列は「チャネル」を示しています。 最も左のグラフは、入力を作成するために使う基本成分を示しています。

電り上のクラクは、ヘルを作成するために使う基本成分を示していよ 2列目は混合入力の3つのチャンネルを表示しています。

3列目は PCA によって計算された成分を表示しています。

計算された成分は、元のチャンネルと一致する必要がないことに注意してください。

これは、ICA の最初のステップが PCA であり、PCA の最初の成分が最も大きな固有値を持つもの(この例では最も 大きなソース振幅を持つもの)であるためです。

4列目は ICA によって計算された成分を表示し、5列目は対応する成分のスペクトル表現を示します。

いくつかの興味深い観察結果:

- 1. 2つの成分のみを混合する場合、PCAは ICA と等価です。
- 2. 2つ以上の信号を混合すると、PCAの結果は通常、混合成分となります。
- 3. 同じ周期を持つ信号を混合すると、PCA(主成分分析)とICA(独立成分分析)の両方とも性能が低下します。
- 4. 抽出された鋸歯成分が反転して表示される場合がありますが、ICA 成分にはスケーリングが適用されていないため、-1 を乗算することで元の向きに戻すことができます。

デモ(1D)の操作手順

新しい Experiment を作成したところからの手順で確認します。

※デモ Experiment の Procedure Windows を開き、プロシージャをコピーし、コンパイルしておきます。

1. Data Browser の左の New Data Folder ボタンをクリッ クします。	New Data Folder X
newAA というデータフォルダーを作ります。	newAA
	OK Cancel
2.Data Browser の左の Browse Expt ボタンをクリックしま す。	Current Das Folder: root
ICA Demo.pxp を選択します。	Strings Brids Pick C Analysis V Analysis New Data Enform
	Seven Core ■ <th< th=""></th<>

3. ICA Demo.pxp から次のウェーブを取り込みます。	Data Browser		
M_ICAComponents M_PCA sp0 sp1 sp2 SS xx	Current Data Folde Display Waves Variables Strings Info Piot New Data Folder Save Copy Browse Ept_ Delete Execute Cmd_	r root Name Toot Michomponents Sp0 Sp1 Sp2 Ss Ss Ss Ss Ss Ss Ss Ss Ss Ss	Done Browsing Done Browsing CAD Demo CAD DE

4. メニュー Macros \rightarrow ICA for 1D Signals を選択します。

Macros Windows Not ICA for 1D signals ICA for 2D signals **5.** 右のような一連のグラフが表示されます。 また、コントロールパネルが表示されます。

それぞれの意味は上記で説明しています。

 コントロールパネルの上部は3つのウェーブ(グラフウィンド ウでは Signal1、Signal2、Optional Signal3 に対応)の振幅 (AmpX)、波形生成の関数(Function)、周期(PeriodX)を設定 します。

下部では ICA のオプションを設定します。

各設定は ICA コマンドのフラグに対応しているため、ICA のヘルプを確認してください。

デモ(2D)の操作手順

新しい Experiment を作成したところからの手順で確認します。

※デモ Experiment の Procedure Windows を開き、プロシージャをコピーし、コンパイルしておきます。

1. Data Browser の左の Browse Expt ボタンをクリックしま す。

ICA Demo.pxp (Examples → Analysis フォルダー内)を選択 します。





			10 10	
ICA Demo	o Data: D			
Amr	1: 2 = Fund	JU 🗣 Num lype	Period1:	12 2
Amp	o2: 1	ction: saw-tooth 💌	Period2:	40 🗘
V Am	53: 1 🗘 Fun	ction: sine 💌	Period3:	17 🗘

Non Quad. Func: exp 🔻 Alpha: 1

Tolerance 0.0001 \$

Update

ICA Operation Options

✓ Cols Flag

Data Browser

Deflate Method

デモを開始するには、Macros メニューから ICA for 2D Signals を選択します。

4. 2D の例は、左の列に表示されている3つの画像の組み合わせから構成されています。

この例のすべての画像は、100 行 100 列の単精度浮動小数点ウェ ーブです。

各画像は1列にまとめられ(Redimension を使用)、3つの画像は3列のウェーブ(10000行)に結合されます。 後者は、前の例と同じ混合行列を使って混合されます。

混合ウェーブの3つの列は、2番目の画像列に表示されています (混合ウェーブの各列は 100x100 ウェーブにリサイズされてい ます)。

3番目の画像列には、ICA コンポーネントが表示されています。



ICA のヘルプ

ICA [flags] srcWave

ICA コマンドは、FastICA アルゴリズムを使って独立成分分析を実行します。 入力データは2次元ウェーブの形式であり、各列は1つのデータ取得チャネルに相当します。 コマンドの結果は、現在のデータフォルダー内のウェーブ M_ICAComponents、M_ICAUnMix、および M_matrixW に保存されます。

ICA コマンドは、Igor Pro 7.0 で追加されました。

フラグ

/A=alpha	alpha は、logCosh 関数の引数として使われる定数です。
	この定数は、exp 関数とは組み合わせて使われません。

Alpha は [1,2] の範囲にあり、デフォルト値は1です。

収束速度や結果の品質に影響を与えるために、この値を変更する必要はほとんどありません。

/CF=num ICA で使うコントラスト関数(非二次関数とも呼ばれる)を指定します。

num=0: logCosh (デフォルト)

num=1: exp

/COLS 入力データを列ごとに平均を引いた後、正規化します。 このフラグを使うと、アルゴリズムは収束し、より良い結果を生成します。

Igor Pro 9.0 以降、入力はデフォルトで前処理されるため、/COLS は不要となり、使用も推奨されなくなりました。

/DCMP=destCAComp

ICA コンポーネントの宛先ウェーブを指定します。 このフラグを使わない場合、ICA コンポーネントは、現在のデータフォルダーの M_ICAComponents というウェーブに保存されます。

/DEVW=destPCAEV

入力行列の SVD の特異値の宛先ウェーブを指定します。 保存先が指定されていない場合、現在のデータフォルダーの W_PCAEV にウェーブが 保存されます。 特異値は実数であり、大きい順に並べられています。 このフラグを使うには、/PCA フラグも指定する必要があります。

/DFLT 反復計算で1回ごとに「分離」行列の1つのベクトルを求めるデフレーション/ベクトル 法を使います。 デフォルトでは、このコマンドは行列法を使い、一度に完全な分離行列を解きます。

/DMXW=destWMat

行列 W の宛先ウェーブを指定します。 追加の反復を実行する場合に、この行列を使用できます (以下の /WINT を参照)。

/DUNX=destUnMix

分離行列の宛先ウェーブを指定します。 このフラグを使わない場合、コマンドは現在のデータフォルダー内のウェーブ M_ICAUnMix にデータを保存します。

/FREE すべての宛先ウェーブをフリーウェーブとして作成します。 このフラグは、ユーザー指定の宛先がない場合にデフォルトで作成される出力ウェーブ には影響しません。

> /FREE は、関数内でのみ、/DEST で指定された destWave が単純な名前またはウェ ーブ参照構造体フィールドである場合にのみ使用できます。

詳細については、ヘルプ Free Waves (Programming.ihf) を参照してください。

/FREE フラグは、Igor Pro 10.0 で追加されました。

/PCA[=pcaDestWave]

「PCA」段階の出力結果を、fastICA 反復処理前のデータの形式で保存します。 データは、現在のデータフォルダーまたはユーザー指定の pcaDestWave のウェーブ M_PCA に保存されます。

/Q 静寂モード。履歴に何も出力しません。

/TOL=tolerance 許容値は、反復が収束するタイミングを決定するために使われます。 デフレクション/ベクトル法では、許容誤差は連続する反復におけるベクトルの値の差を 測定します。

行列法では、許容誤差はすべての成分の平均偏差を測定します。

デフォルトでは、両方の方法において tolerance = 1e-5 です。

/WINT=w 初期の分解行列 W を指定します。 この行列を指定しない場合、アルゴリズムは enoise を使って初期化されます。 ウェーブ w は、srcWave と同じ数値型で、次元 nRows x nCols の2次元でなければ なりません。 ここで、nCols は srcWave の列数です。 初期行列を指定することは、以前の反復計算セットから得たもので、不適切な許容誤差 で収束した可能性がある場合、有用です。

詳細

srcWave は、nRows×nCols の 2 次元ウェーブです。 NaN または INF を含まない、単精度または倍精度の実数値のウェーブでなければなりません。 srcWave の各列は、独立した成分の線形重ね合わせから成る1つのデータ取得チャネルに対応していま す。

これは次の行列積として表すことができます。

X=A (S^t)

S は nRows×nCols の独立成分行列で、[^]t は転置を表します。 A は nCols×nCols の混合行列で、X は「混合」入力です。 ICA(独立成分分析)コマンドは、次の変換から

S=W X

Sの列間の相互情報が最小化されるように、Sの独立した成分を抽出しようします。 相互情報は、成分をスカラー定数で乗算しても影響を受けないため、結果の独立成分はスカラー因子まで指 定することができます。

このコマンドでは、FastICA アルゴリズムを使って独立成分を計算します。

アルゴリズムには、計算に利用できる2つの方法があります。 デフォルトでは、W 行列全体を一度に評価しようとします。 第2の方法(/DFLT フラグ)は「デフレーション」とも呼ばれ、W の行を1行ずつ計算します。 デフレーション法は、入力の列数よりも独立した成分が少ない場合、利点がある可能性があります。

例

// ソースを作成

Make/O/N=(1000,3) ddd
ddd[][0]=sin(2*pi*x/13)
ddd[][1]=sin(2*pi*x/17)
ddd[][2]=sin(2*pi*x/23)

// 混合行列を作成

Make/O/N=(3,3) AA AA[0][0]= {0.291,0.6557,-0.5439} AA[0][1]= {0.5572,0.3,-0.2} AA[0][2]= {-0.1,-0.7,0.4}

GraphOddd Composition Composition



// 混合を実行

MatrixOp/O xx=ddd x AA

// ICA の実行

ICA/DFLT/COLS xx Display M_ICAComponents[][0] Display M_ICAComponents[][1] Display M ICAComponents[][2]



参考

A. Hyvarinen and E. Oja (2000) Independent Component Analysis: Algorithms and Applications, Neural Networks, (13)411-430.