Appendix 1

IATLAB

オブジェクト指向プログラミン

オブジェクト指向プログラミングは,大規模なアプリケー ション・ソフトウェアを開発する際に効率化をもたらします. ソフトウェアをオブジェクト(部品)の組み合わせとして設計す ることで,開発期間の短縮と信頼性の向上が期待できます. C++やJavaと同じように,MATLABでも「状態」と「振る舞 い」をもつオブジェクトを基本としたアプリケーション・ソフ トウェア開発が可能です.

MATLAB オブジェクト指向プログラミングへの誘い

汎用性のある安全なオブジェクトを設計することで,再利用 しやすい部品が整います.さらに,保守性を向上させるために, MATLABには単体テストの環境が用意されています(右掲の **コラム**1を参照).

MATLAB におけるオブジェクト指向プログラミングのポイントを以下にまとめます.

- ・オブジェクトの型枠となるクラスの定義は, @ で始まるディレクトリで行う
- ●実体(インスタンス)オブジェクトを生成する関数(コンストラクタ)は, @を除いたディレクトリ名と同じ名前のMファイルとする
- コンストラクタは状態を保持する変数(フィールド)を構造
 体として定義し,関数 class で実体オブジェクトに変換
 する
- ●オブジェクトの振る舞い(メソッド)は,@で始まるディレ

リスト1 EmSteps コンストラクタ

66



クトリの下に関数として定義する

 メソッドは,引き数にインスタンス・オブジェクトをもつ. フィールドを更新するメソッドは,インスタンス・オブ ジェクトを戻り値とする

村松 正吾

では早速,本特集第1章の例題10のEステップとMス テップをオブジェクト指向でプログラムしてみましょう.

<mark>例題1</mark> クラス定義とオブジェクト生成

本特集 第1章のリスト8(p.64)に示した繰り返し演算解法 (EM アルゴリズム)をオブジェクト指向プログラミングで 実装してみましょう.

 ・野まず,型枠となるクラスを定義します.ディレクトリ
 @EmStepsを作成しましょう.ディレクトリ @EmStepsの下に,以下の三つの関数を準備します.

@EmSteps-EmSteps.m(コンストラクタ)

-process.m(EM ステップ実行メソッド)

└get.m (フィールド取得メソッド)

Mファイル EmSteps.m はインスタンス・オブジェクトを生

リスト2 process メソッド

function emSteps = process(emSt	ceps)
。 % process メソッド	
<pre>% emSteps = process(emSteps) </pre>	
* * フィールドの抽出	
X = emSteps.X; % テータ setPi = emSteps.setPi; setMu = emSteps.setMu;	❀ 混合係数群 ❀ 平均ベクトル群
<pre>setSigma = emSteps.setSigma;</pre>	% 共分散行列群
その他変数 nDists = length(setPi); nSamples = size(X,2); res = zeros(nSamples,nDists);	% 混合数кの抽出 % 標本数Nの抽出 % 分布の重み (Zik)
% EMステップ開始 % : % 省略(リスト 8 と同じ) % : % EMステップ終了	
8 フィールド(パラメータ)の更新 emSteps.setPi = setPi; emSteps.setMu = setMu; emSteps.setSigma = setSigma;	 % 混合係数群 % 平均ベクトル群 ※ 共分散行列群

Interface Oct. 2007

コラム1 単体テスト環境

部品でもシステムでも何かモノを作るときには,それが正しく機 能するか否かを確認する必要があります.これが「テスト」です.一 言で「テスト」といっても,さまざまな手法があります.作ろうとす るモノの仕様を定め,それを満足するか否かを実装後にテストする 方法もあれば,小さな機能をテストしながら実装を進める方法もあ ります.

小さな機能テストは,簡単に即座に繰り返し実行できます.必要 最小限の機能のテストだけを定義し,それを満足する機能を実装し, これを発展的に繰り返すことで部品やシステムを作り上げる「テスト 駆動型開発」という開発手法があります.

テスト駆動型開発では,仕様は発展的に変化します.実装の変化 も容認します.実装の変更は正しい機能への脅威となりますが,細 かいテストの定義が,開発者に変更の勇気を与えてくれます.単体 テスト環境は,このテスト駆動型開発を支援するツールです.

MATLABにもいくつかの単体テスト環境があります.MATLAB Central File Exchange サイト(http://www.mathworks.com/ma tlabcentral/fileexchange/)で"unit testing "というキーワー ドを検索してみてください.

MATLAB R2007a に対応した単体テスト環境に mlUnit があります (図A). mlUnit の Web サイト(http://mlunit.dohmke.de/ Main_Page)からツールをダウンロードできます.本稿執筆時点の バージョンは 1.5.1 です.なお,同ツール内のメソッド assert が R2007a にあらかじめ用意されている組み込み関数と重複するため,

成するコンストラクタを定義します.ディレクトリ名から@を 取った名前にします.

コンストラクタ EmSteps のソース・コードをリスト1に示 します.データ群 xを引き数として受け, 混合係数群,平均 ベクトル群,共分散行列群の初期パラメータをフィールドとし て設定しています.これらをいったん,構造体として定義し, 関数 class でインスタンス・オブジェクトに変換しています. MATLABにおける構造体の定義は,.(ドット)に続けてメン バ変数名を記述するだけです.

process.mは, EM ステップ(1回分)のメソッドを実装し ます.第1章のリスト8と同じ部分は省略して**リスト**2にソー ス・コードを記載します.メソッドprocessでは,入力にク ラス EmStepsのインスタンス・オブジェクト emStepsを受け ます.このメソッドは,フィールドを更新するので,更新した 後のインスタンス・オブジェクトを出力する必要があります.

なお,オブジェクトの状態(フィールド)は隠ぺいされてお り,外部からアクセスできません.外部から状態を観測するた めに,メソッドgetを定義します.**リスト**3にソース・コード を記載します.メソッドgetでは,インスタンス・オブジェク ト emStepsを受けます.このメソッドは,引き数 fieldName で指定されたフィールドの値を switch 文による条件分岐を利 利用前に以下の手続きが必要になります.

1)ファイルmlunit1.5.1.zipを任意のディレクトリ \$HOME に展開 2)MATLAB コマンド・ウィンドウ上でディレクトリを移動

1

App1

2

App2

3

App3

5

>> cd \$HOME/mlunit/src

3)インストール・メソッドを実行

>> install(mlunit);
4)MATLABパスにディレクトリを追加

>> addpath(' \$HOME/mlunit/src ');

上記の手続きのあと,メソッド assert は mlunit_assert に改 名されます.



用して出力します.

クラス EmSteps を利用することで,第1章のリスト8の関数 emmixmvncg はリスト4のように修正できます.クラス EmSteps のオブジェクトは部品として利用できるようになります.

オブジェクト指向プログラミングでは,既に作成したクラス を拡張できる「継承」の仕掛けが特徴の一つといえます.MAT LABにおけるクラスの継承は,関数 class で次のように実現 します.

childObj =...

リスト3 get メソッド

<pre>function value = get(emSteps,fieldName)</pre>	
8	
% get メソッド	
8	
<pre>% value = get(emStepObj,fieldName)</pre>	
8	
switch fieldName % 分岐処 理	
case 'setPi' % 混合係数群	
value = emSteps.setPi;	
case 'setMu' % 平均ベクトル群	
<pre>value = emSteps.setMu;</pre>	
case 'setSigma' % 共分散行列群	
<pre>value = emSteps.setSigma;</pre>	
otherwise % その他(エラー)	
error([fieldName, '有効なフィールド名ではありません '])	
end	

Interface Oct. 2007