MATLAB の使用方法

コマンドラインに

```
>> a = 1
```

と入力すると、変数 'a' に実数値'1' が代入される。MATLAB の変数は基本的に行列やベクトルであり、例えば

```
>> b = [ 1 2 3 ]
```

によって横ベクトル

```
>> c = [ 1; 2; 3 ]
```

によって縦ベクトル

```
>> D = [ 1 2 3; 4 5 6; 7 8 9 ]
```

によって行列を代入することができる。
行列の各成分には、

```
>> e = D( 3, 1 )
```

のようにしてアクセスできる。
等差数列成分をもつベクトルは次のような作りかたができる。

```
>> b = 1:1:3
```

これは 1 からスタートして 1 ステップ刻みで 3 までの数列ベクトル、を意味し、

```
>> b = [ 1 2 3 ]
```

と同じ意味である。
ステップが 1 であるときは省略して、

```
>> b = 1:3
```

と書くこともできる。
行列の成分へのインデックスに、自然数値ベクトルを使うことができる。

```
>> b0 = D( 1, 1:3 )
>> b1 = D( 2, [1 2 3] )
>> B2 = D( [1 3], [1 3] )
```

行列の第 1 行目の横ベクトル全体を、

```
>> b = D( 1, : )
```

と書くこともできる。

```
>> D( :, : )
```

とすれば行列全体である。行列やベクトル同士のかけ算は成分に分解せずに、

```
>> f = b * c
>> g = D * c
```

のように行列まるごとでおこなうことができる。また、

```
>> h = D * b
```

のような不可能な演算にはエラーメッセージを出す。
次元の等しい行列 (ベクトル) の各要素に関するかけ算割り算が、

```
>> a = [ 1 2 ]
>> b = [ 1 3 ]
>> c = a ./ b
>> d = a .* b
```

のようにして可能である。
行列 (ベクトル) にスカラーを足すと、全成分に足したことになる。

```
>> a + 1
```

関数はひととおり揃っている。

```
>> sin( 2*pi )
>> log( exp( 5 ) )
>> sqrt( 2 )
```

スカラー値関数に行列を入れると、各成分が演算される。

```
>> log10( [ 10 100; 1 0.1] )
```

行列関数というのもいろいろある。

```
>> A = [ 8 2; 2 10 ]
>> sum( A )  % 和
>> A'        % 転置
>> min( A )  % 最小値
>> inv( A )  % 逆行列
>> eig( A )  % 固有値
```

行列をつくる関数もいろいろ。

```
>> eye( 3 )     % 単位行列
>> zeros( 3,2 ) % ゼロ値行列
>> rand( 3,3 )  % 一様乱数行列
>> randn( 2,4 ) % 正規乱数行列
```

ベクトルのプロットも簡単。

```
>> x = -pi:0.1:pi
>> y = sin( x )
>> plot( x, y )
```

計算結果画面出力の抑制には、行末にセミコロンをつける。変数の中身が知りたいときは、変数名を打ちこめばよい。

```
>> X = 1:10 ;
>> X
```

オンラインヘルプによって各種関数の解説が得られる。

```
>> help sum
>> help plot
```

MATLABのソースファイルはその名前に拡張子'.m'がつき、m-filesと呼ばれる。その実体はテキストファイルであり、メモ帳などのエディタで編集ができる。MATLAB標準のソース編集ソフトウェアを使うのも良い。

m-filesには、scriptとfunctionの二種類がある。
script fileはそのfile名から'.m'を除いたものをコマンドラインに入力することで実行される。script fileの実行は、fileに書かれた各行をコマンドラインから入力することと同じ意味を持つ。
function fileも基本的に同様だが、引数を持つ。また出力以外の内部変数は隠蔽されており外から参照できない。
詳しくはサンプルを参照のこと。

演習 1：RBF

サンプルプログラムの解説

コマンドプロンプトで

```
>> RBF1demo
```

と入力すると、グラフが出力される。横軸が x 軸、縦軸が y 軸であり、青色の分布点は学習データセット、赤色の折れ線はパラメータ学習後のRBFの出力である。緑色の線分は基底関数の位置と大きさを表しており、中心が基底中心 μ_j、長さが 2σ である。

ここで見ていただいた'RBF1demo'の実体は、ソースファイル'RBF1demo.m'である。ソースファイルをエディタで開いて読みながら、以下の解説を読みすすめられたい。

ファイル'RBF1demo.m'の前半では、学習データ $(x(t), y(t))$, $t = 1, \ldots, T$ と基底の中心位置 μ_j と基底の分散半径 σ が定義される。デフォルトとして学習データはabs(sin)曲線に小さな正規ノイズが乗ったもの。基底個数 $M = 10$ 個、基底中心はその x-rangeを等間隔に切ったもの、基底分散半径は x-rangeに比例、基底個数に反比例する値、にそれぞれ設定されるようになっている。これらは任意に変更可能である。まず、基底の個数 M の値を変えることによって、関数近似の具合の変化を見る。また、基底分散半径の影響も見てみる。

後半では、これらを引数とする関数'RBF1learn'によって最小二乗法にもとづくパラメータ a_j の推定を行い、'RBF1out'によってその関数形を計算し、'RBF1plot'によってグラフにプロットする。

まず'RBF1out.m'を読んでこれがテキスト(6)式で定義される出力を行うことを確かめよ。

'RBF1learn'は最小二乗法によるパラメータ推定を行うメインのアルゴリズムである。

演習 2: AR法

デモ'AR.m'を実行せよ。

青の分布点は学習データセット。赤のラインは、学習データセットの前半部のみにもとづいて、AR法によって後半部を予測したものである。

演習 3: EM法によるクラスタリング

デモ'EM.m'を実行せよ。スペースキーを押すと、E-stepとM-stepが1 epochずつ進行する。収束したと思ったら、ctrl-Cによって止めること。

青の分布点は、2次元散布点の学習データセット $\{x(t)|t = 1, \ldots, T\}$ であり、赤円は各クラスタ中心を中心点とし標準偏差 σ を半径とした円を表す。

演習 4: Kalman filter

```
>> cd pendulum
```

によってディレクトリ移動したのち、デモ'pendulum_demo.m'を実行する。