An Introduction to Matlab

For CSE 802: Pattern Recognition

Feng Kang
kangfeng@cse.msu.edu
Start Matlab

► You can access it from CSE lab but it’s more easy to go to engineering lab and use Matlab there.

► Machines in engineering building’s labs.
  - Start->All Programs->Matlab 7.0.

► License issues, especially for some toolbox.
  - Exit Matlab if you do not use it.
Topics

► Data structure of Matlab.

► Some useful Matlab functions for this course.

► Plotting of data.

► Two examples:
  - Plotting of multivariate Gaussian data.
  - PCA: compute PCA and plot the data of reduced dimensionality.
 Scalars, Vectors and Matrices

► Scalar:
  - Just a number: \( a = 1; \ b = 3; \)

► Vector:
  - Column vector: \( a = [1; \ 2; \ 3; \ 4]. \)
  - Row vector: \( b = [1 \ 2 \ 3 \ 4]. \)
  - Transpose: \( a = b'; \)

► Matrix:
  - \( A = [1 \ 2 \ 3; \ 4 \ 5 \ 6; \ 7 \ 8 \ 9]; \)
Access Elements in Matrices

► Access a single element.
  - A[row index, column index]
  - A[1,3] = 3;

► Access a sub-matrix.
  - Extract out part of rows: B = A[1:2, :];
  - Extract out part of columns: C = A[:,1:2];
Operations on Matrix

► Cell by cell operation.
  - ‘.’
  - E.g. B = A. ^2;
  - B= [1 4 9; 16 25 36; 49; 64; 81];

► Matrix operation.
  - ‘+’, ‘-’, ‘*’. 
Control Structures for Matlab(1)

- Conditional statements.
  
  ```matlab
  if expression1
      statements1
  elseif expression2
      statements2
  else
      statements3
  end
  
  if (a>3)
      b=4;
  end;
  ```

- Example
Control Structure for Matlab(2)

► Loop structure: for loop

```matlab
j=0;
for i=1:10
    j = j+i;
end
```

► Loop structure: while loop

```matlab
while expression
    statements
end
```
Symbolic Toolbox(1)

- Declare a symbol object.
  - Not a number but a symbol.
  - Syntax: `syms arg1 arg2 ... real`.
  - Use symbols to represent a function.

```matlab
syms x u real
syms s positive
f = exp(-(x-u)^2/s^2);
```
Symbolic Toolbox(2)

- Manipulate the function.
  - Compute integration.
  - \( g = \int f(x, -\infty, \infty); \text{ result: } g = s \cdot \pi^{(1/2)} \)
  - Gaussian distribution: \( f/g \)

- There are many other ways to manipulate the functions: e.g. differentiation.
Load and save data

► Load data:
  - Matrix format: load('file path');

► Save data:
  - Matrix format: save('file path', 'matrix name', '-ascii');
Common Functions in CSE 802

- Functions related to Multivariate Gaussian distribution.
  - mean(A)
  - cov(x), cov(x,y); x, y are vectors.
  - inv(A): inverse of the matrix.
  - det(A): determinant of the matrix.
  - mvnrnd(mu, sigma, num of data.)

- Functions related to dimensionality reduction.
  - eigs(A): compute eigenvector of A.
Plotting

► Plot function:
  - Plot one line: plot(X1,Y1,LineSpec).

  - Plot several lines on the same figure:
    - figure(1);
    - hold on;
    - plot(x1, y1, LineSpec1);
    - plot(x2, y2, LineSpec2);
    - ... 
    - hold off;
    - legend('line 1', 'line 2', ...);
    - xlabel('description of x axis'); ylabel('description of y axis');
x = 1:10;
y = 3*x;
z = x.^2;

figure(1)
hold on;
plot(x, y, '-ro');
plot(x, z, '-b*');
hold off;
legend('y=3*x', 'z=x.^2');
xlabel('x'); ylabel('function values');
Ezplot

- Mainly used for implicitly defined functions.
  - Sometimes, it’s more convenient to plot the implicit form of the functions.
    - E.g. \( x^2 + y^2 = 1 \)
  - Function format: \( \text{ezplot}(f,[\text{xmin},\text{ xmax},\text{ ymin},\text{ ymax}]) \) plots \( f(x,y) = 0 \) over \( \text{xmin} < x < \text{xmax} \) and \( \text{ymin} < y < \text{ymax} \).
    - The first parameter \( f \) is passed as a string.
figure(1)
ezplot('x^2+y^2-1',[-1,1,-1,1]);
xlabel('x');
ylabel('function values');
Generate Multivariate Gaussian Data

Generate multivariate Gaussian data.

- \( \text{rand\_data} = \text{mvnrnd(mu, sigma, num of data.)} \)
- \( \text{E.g.} \)

\[
\begin{align*}
\text{mu1} &= [0 \ 0]; \\
\text{sigma1} &= [1 \ 0; \ 0 \ 1]; \\
\text{r1} &= \text{mvnrnd(mu1, sigma1, 50)}; \\
\text{plot(r1(:,1), r1(:,2), '*');}
\end{align*}
\]
PCA to Extract the Major Information in Data and Plot it.

- PCA to reduce dimensionality of data and plot them in 2-D space.

Example: IRIS data:
- four dimensional data. Hard to visualize.
- Apply PCA to reduce to two dimensional data and plot them.
Example codes of PCA

Codes:

```matlab
X = load('iris_data');
c = mean(X);
X = X - repmat(c, size(X,1), 1);
covar = cov(X);
opt.disp = 0;
[p, D] = eigs(covar, 2, 'LA', opt);
reduced = X*p;
figure(1)
hold on;
plot(reduced(1:50, 1), reduced(1:50, 2), 'o');
plot(reduced(51:100, 1), reduced(51:100, 2), '*');
plot(reduced(101:150, 1), reduced(101:150, 2), '+');
hold off;
legend('Setosa', 'Versicolour', 'Virginica');
```
Plot of PCA