Matlab Functions and Tools

- Some functions are given. There is no need to implement them. Please refer to the links in the next slide.
- You can also use other resources, with proper citation in your report.

Useful Matlab Functions

- Naïve Bayes classifier
  - PredictClass = classify(Xtest,Xtrain,Ytrain,'diaglinear');
- Randomly split data
  - p = randperm(n,k)
  - Indices = crossvalind('Kfold', N, K)
- plotImages
  - plotImages(digitsImages, xy_coord, scale, skip);
- LLE:
  - http://www.cs.nyu.edu/~roweis/lle/code.html
- ISOMAP:
  - http://isomap.stanford.edu/
- LDA-dimension reduction
  - http://homepage.tudelft.nl/19j49/Matlab_Toolbox_for_Dimensionality_Reduction.html

decide the model to learn

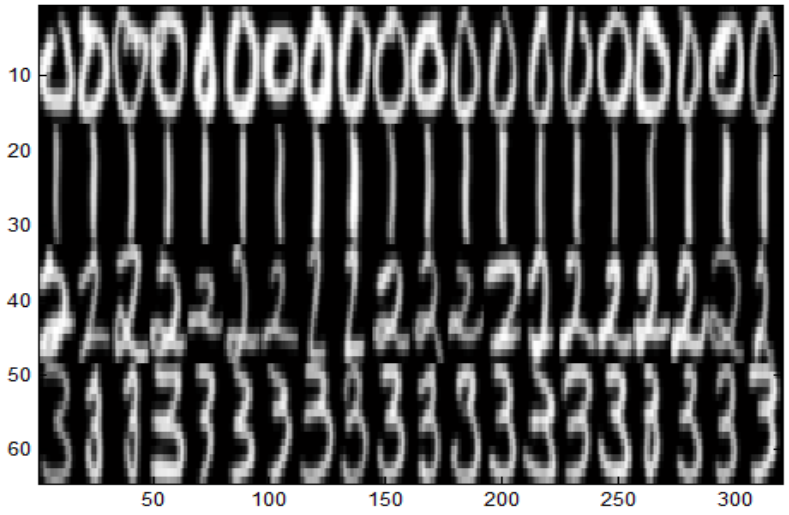
n: total # of samples
k: select k samples by permutation

Datasets

- Dataset A (record activity sensors):
  - Sample-feature matrix: fea (19,000 x 81)
    - Features: readings of 81 sensors
    - The data is in **time-series**, given in time order
  - Missing values
    - ‘NaN’
  - Outliers
    - Negative readings are not outliers

Datasets

- Dataset B (image data of handwritten digits)
  - Sample-feature matrix: fea (2066 x 784)
    - Features: 28 x 28 gray-scale images, in column-wise
  - Ground truth labels: gnd (2066 x 1)
    - Labels: 0, 1, 2, 3, 4



Datasets

- Dataset C (clinic data)
  - Sample-feature matrix:
    - fea (2100 x 21)
    - Need to be normalized (min-max) before further processing
  - Ground truth labels
    - gnd (2100 x 1)
    - 3 classes: normal(1), suspect(2), pathologic(3)

Q1: Data Cleaning and Preprocessing

- Missing values; Outliers
  - Detect and fix them
- Normalization:
  - Min-max
  - Z-score
- Plot histograms
- Observations and comments!

Q2&3: Feature Extraction

- Linear methods:
  - PCA
  - LDA
- Nonlinear methods:
  - LLE
  - ISOMAP
- Supervised vs. non-supervised dimensionality reduction

Q4: Feature Selection

- Search strategy
  - SFS
  - SBS
- Objective function
  - Filter based
  - Wrapper based

basics

- ‘Hello World!’
  - a = 3;
  - b = 4;
  - c = a + b
- end each statement with semicolon, if you do not like to see the result in the command window

basics

- arithmetic operators:
  - addition: A+B
  - subtraction: A-B
  - multiplication: A*B
  - right division: A/B = A*inv(B)
  - left division: A\B = inv(A)*B
  - power: A^b
  - transpose: A’
  - colon operator:
    - to create vectors: a:b
    - array subscripting: A(:,b)

basics

- dot operators (a.k.a element-wise operators)
  - A.*B, A./B, A.\B, and A.^B
- relational operators
  - a<b, a<=b, a>b, a>=b, a==b, and a~=b
- logical operations
  - a||b (or), a&&b (and), ~ a(not)
- element-wise logical operators
  - A|B, A&B, ~A

basics

- operator precedence
  - Parentheses
  - transpose and power
  - unary plus, unary minus, and logical negation
  - multiplication(s) and division(s)
  - addition and subtract
  - …

basics

- flow control
  - conditional control
    - if, else, and elseif
    - switch and case
  - loop control
    - for
    - while
    - break
    - continue

basics

- if

```
if expression1
   statements1
elseif expression2
   statements2
else
   statements3
end
```

basics

- for

```
for index = values
   program statements
end
```

- while

```
while expression
   program statements
end
```

basics

- function definition
  - `function [output_variables] = fcn_name(input_variables)`
  - the name of a function should be consistent with the file name

an example

- Given the corresponding coefficients of two lines (ax+by+c=0), calculate the intersection point and plot the lines on a figure.
- Function:

```
[intersection,Runtime] = myPlot(line1,line2)
```

- Script to call this function:

```
clear all;clc
load coeffs
line1 = coeff(1,:);
line2 = coeff(2,:);
[intersection,RunTime] = myPlot(line1,line2)
```

how to access?

- have a license
- Nexus computers
  - on campus
  - remotely
- Octave

help

- where to look for answers?
  - Matlab Help
  - Mathworks website
  - Online forums
  - TAs

refs

- www.mathworks.com
- www.gnu.org/software/octave/
- saw.uwaterloo.ca/matlab/