Lecture 12

Midterm Exam: Oct. 4

Closed-book and closed-notes
Formula sheet (one-page, two-sided)
Calculator allowed

Bayes Classifiers
- Bayes Theorem

- Decision Function formula

$$d_j(x) = p(x|c_j)P(c_j) = \frac{1}{\sqrt{2\pi}\sigma_j} e^{-\frac{(x-m_j)^2}{2\sigma_j^2}} P(c_j)$$

where $j = 1, 2$

$$p(\mathbf{x}/\omega_j) = \frac{1}{(2\pi)^{n/2}|\mathbf{C}_j|^{1/2}} e^{-\frac{1}{2}(\mathbf{x}-\mathbf{m}_j)^T \mathbf{C}_j^{-1}(\mathbf{x}-\mathbf{m}_j)}$$

Pages 30- 36 of Slides "Bayes Classifiers": Naïve Bayes decision functions and decision boundary

Matrix/Vector Calculus
Review HW 2 and HW 3 problems.

Mahalanobis distance

Given the sample mean and covariance matrix, determine the Mahal distance.

$$Cov(X_1, X_2) = \begin{bmatrix} Var(X_1) & ? \\ ? & Var(X_2) \end{bmatrix}$$

```
>> N = 1000000;
>> m1 = [-4, 0];   % Mean vector
c1 = [1,0; 0,1]; % Covariance matrix
rng default  % For reproducibility
r1 = mvnrnd(m1,c1,N);
>> x1 = r1(:,1);
>> x2 = r1(:,2);
```

```
>> var(x1)

ans =

    0.9994
```

```
>> var(x2)

ans =

    1.0013
```

```
>> cov(r1)

ans =

    0.9994   -0.0002
   -0.0002    1.0013
```

```
>> mean((x1-mean(x1)).*(x2-mean(x2)))

ans =

   -2.0185e-04
```

KNN method

Review the algorithm and the underlying theory (Bayes Theorem)

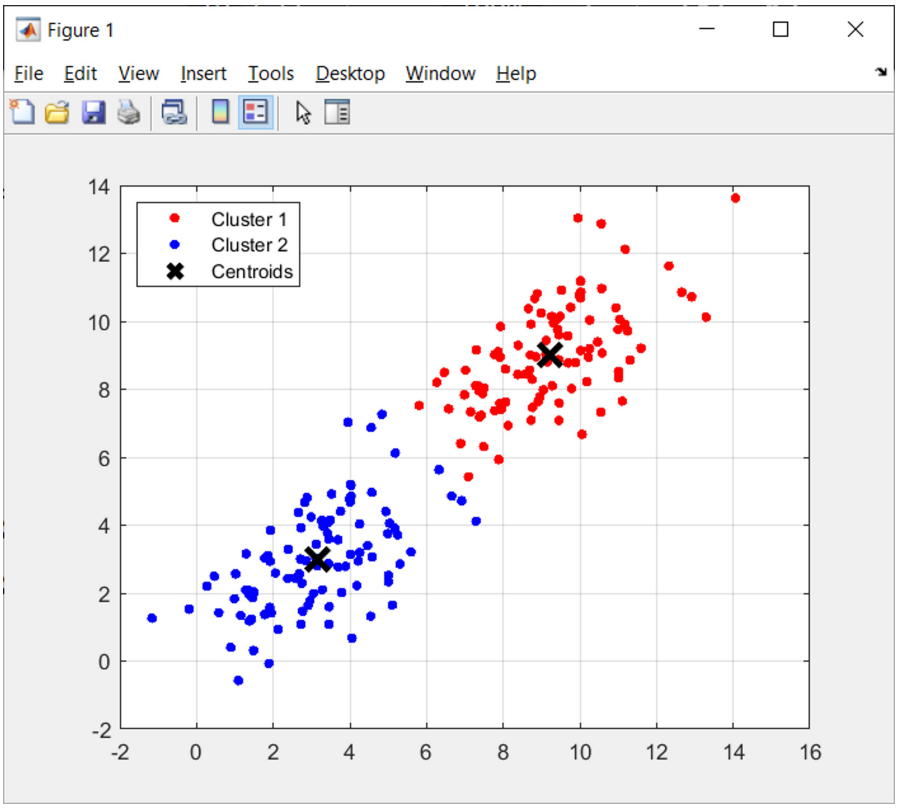
Performance metrics

Generate Confusion Matrix chart

Calculate the metrics:

Accuracy, Sensitivity, Specificity, Precision, F1 score

==========




```
% kmeans_demo.m

…

>> [idx,C,sd] = kmeans(X,2);

>> % sum of Euclidean distance squared
sd
d1 = pdist2(X(idx==1,:), C(1,:)).^2;
d2 = pdist2(X(idx==2,:), C(2,:)).^2;
sum(d1)
sum(d2)

sd =

  480.7163
  475.0740


ans =

  480.7163


ans =

  475.0740
```

'kmeans_demo.py'

k-means method

```
import numpy as np
infile = r"C:\Users\...\kmeans.csv"
dataset = np.loadtxt(infile, delimiter=',')
X = dataset[:, 0:2]

from sklearn.cluster import KMeans

cluster = KMeans(n_clusters=2, random_state=0).fit(X)
cluster.cluster_centers_
```

```
Out[9]:
array([[9.20631359, 9.00165323],
       [3.14182431, 2.99438324]])
```

```
xtest = [[6,4]]
cluster.predict(xtest)
xtest = [[6,8]]
cluster.predict(xtest)
```