Lecture 6

Nearest Neighbor Methods

Cosine Similarity

```
rom numpy import dot
from numpy.linalg import norm
A = [1, 0]
B = [2, 2]
dot(A, B)/(norm(A) * norm(B))

dot(A, B)/(norm(A) * norm(B))
Out[6]: 0.7071067811865475
```

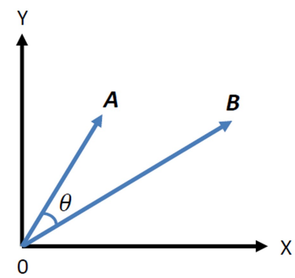
```
from sklearn.metrics.pairwise import cosine_distances
cosine_distances([A], [B])

cosine_distances([A], [B])
Out[8]: array([[0.29289322]])
```

Compute cosine distance between samples in X and Y.

Cosine distance is defined as 1.0 minus the cosine similarity.

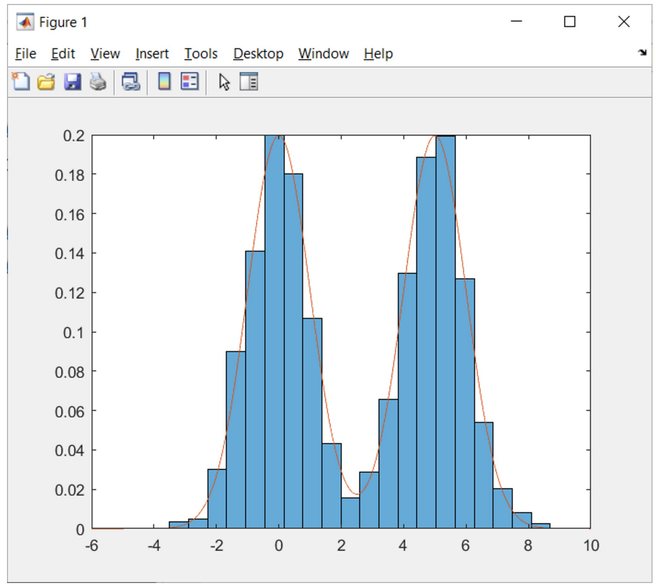
Cosine Similarity

- The cosine of two non-zero vectors can be derived by using the Euclidean dot product formula:
  $$\boldsymbol{A} \cdot \boldsymbol{B} = \|\boldsymbol{A}\|\|\boldsymbol{B}\| \cos(\theta)$$
- *Cosine Similarity* measures the similarity between two vectors of an inner product space as:
  $$S_C(\boldsymbol{A}, \boldsymbol{B}) \triangleq \cos(\theta) = \frac{\boldsymbol{A} \cdot \boldsymbol{B}}{\|\boldsymbol{A}\|\|\boldsymbol{B}\|}$$



Histogram Estimation

```
close all;
clear all;
M = 1000;
X = [randn(M, 1); 5+randn(M, 1)];
h = histogram(X, 20, 'Normalization', 'pdf');

% The true pdf
x = -6: 0.01: 10;
f = 1/2*(normpdf(x, 0, 1) + normpdf(x, 5, 1));
>> hold on;
plot(x,f)

>> trapz(x,f)

ans =

   1.0000
```



- The K-nearest-neighbor technique for density estimation can be extended to the problem of classification, where we apply the K-nearest-neighbor density estimation technique to each class separately and then use the Bayes' theorem.
- Suppose that we have a data set comprising N_i points in class C_i with N points in total, so that $\Sigma_i N_i = N$.
- The class priors (*a priori* probabilities) are given by $p(C_i) = \frac{N_i}{N}$.
- If we want to classify a new point $\boldsymbol{x}$, we draw a sphere centered on $\boldsymbol{x}$ containing precisely K points irrespective of their class.
- Suppose this sphere has volume V and contains K_i points from class C_i, then an estimate of the density associated with each class is $f(\boldsymbol{x}|C_i) = \frac{K_i}{N_i V}$, where $\Sigma_i K_i = K$.
- The unconditional density is given by $f(\boldsymbol{x}) = \frac{K}{NV}$.

Posterior Probability

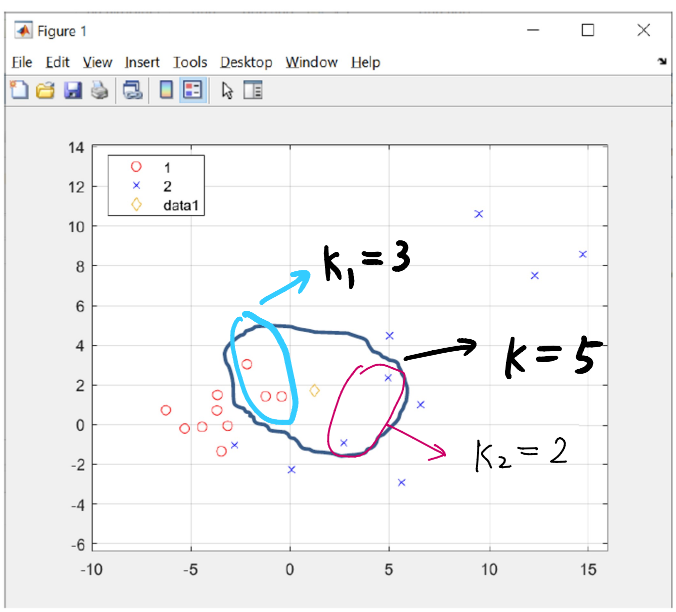
Using Bayes' theorem to obtain the posterior probability of class membership:

$$f(C_i|\boldsymbol{x}) = \frac{f(\boldsymbol{x}|C_i)p(C_i)}{f(\boldsymbol{x})} = \frac{K_i}{K}$$

where

$$f(\boldsymbol{x}|C_i) = \frac{K_i}{N_i V}, p(C_i) = \frac{N_i}{N}, \text{ and } f(\boldsymbol{x}) = \frac{K}{NV}$$

- In order to minimize the probability of misclassification, we assign the test point $\boldsymbol{x}$ to the class having the largest posterior probability, corresponding to the largest value of $\frac{K_i}{K}$.
- Thus to classify a new point, we identify the K nearest points from the training data set and then assign the new point to the class having the *largest* number of representatives amongst this set.



$\frac{3}{5} > \frac{2}{5}$

$\frac{K_1}{K} > \frac{K_2}{K}$

Assign ◇ to class 1