Lecture 22  Clustering

1. Basics concepts and definitions

2. Clustering algorithms
   •  Sequential algorithms
   •  Cost function optimization clustering algorithms
Concepts and definitions

- **Supervised pattern recognition**
  - class label for each training pattern was known

- **Unsupervised pattern recognition**
  - class label was not available.

- **Definition of Clustering**: Given a set of data vectors $X = \{ x_1, \ldots, x_N \}$, group them such that “more similar” vectors are in the same cluster and “less similar” vectors are in different clusters. The set containing these clusters is called a *clustering of X*. 
Formal definition

- A clustering of a set $X$ is a partition of the $X$ according to certain criteria

$$X = \{x_1, x_2, \ldots, x_N\}. \quad (11.1)$$

We define as an $m$-clustering of $X$, $\mathcal{R}$, the partition of $X$ into $m$ sets (clusters), $C_1, \ldots, C_m$, so that the following three conditions are met:

- $C_i \neq \emptyset, i = 1, \ldots, m$
- $\bigcup_{i=1}^{m} C_i = X$
- $C_i \cap C_j = \emptyset, i \neq j, i, j = 1, \ldots, m$
Example 7.2.1. Consider the data vectors shown in Figure 7.1. Two clusterings that are in line with the definition just given are \( \mathcal{R}_1 = \{\{x_1, x_2\}, \{x_3, x_4\}, \{x_5, x_6, x_7\}\} \) and \( \mathcal{R}_2 = \{\{x_1, x_2, x_3, x_4\}, \{x_5, x_6, x_7\}\} \).
A dissimilarity measure (DM) $d$ on $X$ is a function.

$$d : X \times X \to \mathcal{R}$$

where $\mathcal{R}$ is the set of real numbers, such that

$$\exists d_0 \in \mathcal{R} : -\infty < d_0 \leq d(x, y) < +\infty, \quad \forall x, y \in X$$

$$d(x, x) = d_0, \quad \forall x \in X$$

and

$$d(x, y) = d(y, x), \quad \forall x, y \in X$$

If in addition

$$d(x, y) = d_0 \quad \text{if and only if} \quad x = y$$

and

$$d(x, z) \leq d(x, y) + d(y, z), \quad \forall x, y, z \in X$$

Euclidean distance is a metric dissimilarity measure.
A similarity measure (SM) \( s \) on \( X \) is defined as

\[
s : X \times X \to \mathcal{R}
\]

such that

\[
\exists s_0 \in \mathcal{R} : -\infty < s(x, y) \leq s_0 < +\infty, \quad \forall x, y \in X
\]

\[
s(x, x) = s_0, \quad \forall x \in X
\]

and

\[
s(x, y) = s(y, x), \quad \forall x, y \in X
\]

If in addition

\[
s(x, y) = s_0 \quad \text{if and only if} \quad x = y
\]

and

\[
s(x, y)s(y, z) \leq [s(x, y) + s(y, z)]s(x, z), \quad \forall x, y, z \in X
\]

\( s \) is called a metric SM.

Which quantifies the requirement for more “similar” vectors to be in the same cluster and less “similar” vectors to be in different clusters.
Examples of SM

- **cosine similarity measure,**

\[ s_{\text{cosine}}(x, y) = \frac{x^T y}{\|x\| \|y\|} \]

- **Pearson’s correlation coefficient**

\[ r_{\text{Pearson}}(x, y) = \frac{x_d^T y_d}{\|x_d\| \|y_d\|} \]

where \( x_d = [x_1 - \bar{x}, \ldots, x_l - \bar{x}]^T \) and \( y_d = [y_1 - \bar{y}, \ldots, y_l - \bar{y}]^T \)

- **Tanimoto measure,**

\[ s_T(x, y) = \frac{x^T y}{\|x\|^2 + \|y\|^2 - x^T y} \]
Proximity Functions between a Point and a Set

In many clustering schemes, a vector $x$ is assigned to a cluster $C$ taking into account the proximity between $x$ and $C$, $\varphi(x, C)$. There are two general directions for the definition of $\varphi(x, C)$. According to the first one, all points of $C$ contribute to $\varphi(x, C)$. Typical examples of this case include:

- The **max proximity function**:
  
  \[ \varphi_{\text{max}}^{ps}(x, C) = \max_{y \in C} \varphi(x, y) \]

- The **min proximity function**:
  
  \[ \varphi_{\text{min}}^{ps}(x, C) = \min_{y \in C} \varphi(x, y) \]

- The **average proximity function**:
  
  \[ \varphi_{\text{avg}}^{ps}(x, C) = \frac{1}{n_C} \sum_{y \in C} \varphi(x, y) \]

  where $n_C$ is the cardinality of $C$. 
Proximity Functions between Two Sets

- 

Proximity functions \( \varphi^{ss} \) used for the comparison of sets are based on proximity measures, \( \varphi \), between vectors. If \( D_i, D_j \) are two sets of vectors.

- The mean proximity function:

\[
\varphi_{\text{mean}}^{ss}(D_i, D_j) = \varphi(m_{D_i}, m_{D_j})
\]

- The average proximity function:

\[
\varphi_{\text{avg}}^{ss}(D_i, D_j) = \frac{1}{n_{D_i} n_{D_j}} \sum_{x \in D_i} \sum_{y \in D_j} \varphi(x, y)
\]
Proximity Functions between Two Sets

The *max* proximity function:

$$\varphi_{\text{max}}^{ss}(D_i, D_j) = \max_{x \in D_i, y \in D_j} \varphi(x, y)$$

The *min* proximity function:

$$\varphi_{\text{min}}^{ss}(D_i, D_j) = \min_{x \in D_i, y \in D_j} \varphi(x, y)$$
Basic Sequential Algorithmic Scheme (BSAS)

- The BSAS algorithm performs a single pass on a given data set. In addition, each cluster is represented by the mean of the vectors that have been assigned to it.

**Algorithm**

1. For each new vector \( x \),

   *compute* its distance from the already formed clusters

   If the distance is larger than a (user-defined) threshold of dissimilarity, and if the maximum allowable number of clusters, \( q \), have not been reached, a new cluster containing \( x \) is created; Otherwise, \( x \) is assigned to its closest cluster and the corresponding representative is updated

2. The algorithm terminates when all data vectors have been considered once.

See matlab implementation
Refinement procedure

- This procedure is applied on a clustering that has already been obtained.

- It performs a single pass over the data set. The closest cluster for each vector, \( x \), is determined. After all vectors have been considered, each cluster is redefined using the vectors identified as closest to it. If cluster representatives are used, they are re-estimated accordingly (a usual representative is the mean of all the vectors in a cluster)
Consider the 2-dimensional data set $X$ consisting of the following vectors:

$$x_1 = (2, 5), \ x_2 = (6, 4), \ x_3 = (5, 3), \ x_4 = (2, 2), \ x_5 = (1, 4),$$
$$x_6 = (5, 4), \ x_7 = (3, 3), \ x_8 = (2, 3), \ x_9 = (2, 4), \ x_{10} = (8, 2),$$
$$x_{11} = (9, 2), \ x_{12} = (10, 2), \ x_{13} = (11, 2), \ x_{14} = (10, 3), \ x_{15} = (9, 1)$$
Algorithm

1. Apply the BSAS algorithm on X, presenting its elements in the order
   $x_8, x_6, x_{11}, x_1, x_{15}, x_2, x_3, x_{14}, x_7, x_{10}, x_9, x_{12}, x_{13}, x_4, x_{15}$
   for threshold $= 2.5$ and $q = 15$

Order and threshold matter!!!

2. Repeat step 1, now with the order of presentation of the data vectors to
   the algorithm as $x_7, x_3, x_1, x_{15}, x_9, x_6, x_8, x_4, x_{10}, x_{15}, x_{13}, x_14, x_{11}, x_{12}$

3. Repeat 1, now with threshold $= 1.4$

Matlab test
Merging Procedure

• This procedure is applied on a clustering of a given data set to merge pairs of clusters that exhibit high “similarity” (low “dissimilarity”)

• Two clusters are merged and the procedure is repeated on the resulting clustering

• Procedure terminates if the dissimilarity is greater than a (user-defined) cluster-dissimilarity threshold

• Merging is sensitive to the value of parameter threshold, the choice of which mostly depends on the problem at hand.
Cost function optimization clustering algorithm

• Each cluster, $S_j$, *in a clustering is parameterized by a vector of parameters* $\theta_j$.

• *The aim* is to identify the values of these parameter vectors, which characterize the clustering structure of $X$ *in* an optimal sense.

• This is carried out via the optimization of appropriately defined functions.
Optimization problem of k-mean alg.

- Given a set of observations \((x_1, x_2, \ldots, x_n)\), where each observation is a \(d\)-dimensional real vector, \(k\)-means clustering aims to partition the \(n\) observations into \(k\) sets \((k \leq n)\) \(S = \{S_1, S_2, \ldots, S_k\}\) so as to minimize the within-cluster sum of squares (WCSS):

\[
\arg\min_{S} \sum_{i=1}^{k} \sum_{x_j \in S_i} \|x_j - \mu_i\|^2
\]

- where \(\mu_i\) is the mean of points in \(S_i\).
**k-Means Algorithm**

- This is the most widely known clustering algorithm.

- The parameter vectors $\theta_j$ (also called cluster representatives or simply representatives) correspond to points in the $l$-dimensional space, where the vectors of data set $X$ live.

- *k*-means assumes that the number of clusters underlying $X$, $m$, is known. Its aim is to move the points $\theta_j$, $j = 1, \ldots, m$, into regions that are dense in points of $X$ (clusters).
**k-Means Algorithm**

Algorithm
1. Randomly choose k items and make them as initial centers
2. For each point, find the nearest center and assign the point to the cluster associated with the nearest center
3. Update the center of each cluster based on the items in that cluster. The new center will be the mean of all points in the cluster
4. Repeats steps 2 and 3, till no point switches clusters.
Example

- Generate and plot a data set, $X_3$, that consists of $N = 100$ 2-dimensional points. These points form four equally sized groups. Each group contains vectors that stem from Gaussian distributions with means $m_1 = (0, 0)$, $m_2 = (10, 0)$, $m_3 = (0, 9)$, and $m_4 = (9, 8)$, respectively, and respective, covariance matrices $S$.

```matlab
randn('seed',0)
m=[0 0; 4 0; 0 4; 5 4];
S(:,:,1)=eye(2);
S(:,:,2)=[1.0 .2; .2 1.5];
S(:,:,3)=[1.0 .4; .4 1.1];
S(:,:,4)=[.3 .2; .2 .5];
n_points=100*ones(1,4); %Number of points per group
X3=[];
for i=1:4
    X3=[X3; mvnrnd(m(:,i),S(:,:,i),n_points(i))];
end
X3=X3';
```

See *matlab k-mean alg. Test*.
m=4;
[l,N]=size(X3);
rand('seed',0)
theta_ini=rand(l,m);
[theta,bel,J]=k_means(X3,theta_ini);
figure(1), hold on
figure(1) plot(X3(1,bel==1),X3(2,bel==1),'r',... X3(1,bel==2),X3(2,bel==2),'g*',X3(1,bel==3),X3(2,bel==3),'bo',... X3(1,bel==4),X3(2,bel==4),'cx',X3(1,bel==5),X3(2,bel==5),'md',... X3(1,bel==6),X3(2,bel==6),'yp',X3(1,bel==7),X3(2,bel==7),'ks')
figure(1), plot(theta(1,:),theta(2,:),'k+')
figure(1), axis equal