Middle East Technical University | Department of Computer Engineering

CENG 499

Introduction to Machine Learning

Fall 2015-2016

Homework 3

Due date: Jan 8, 2015, Fri, 23:55

1 Objectives

To familiarize yourselves with **unsupervised** learning, i.e. **clustering**, by using **k-means**, a *partitional* algorithm and **hierarchical** clustering under **complete** linkage and **Ward's** distance, and also to assess clustering output with respect to **internal** and **external** validity criteria.

2 Cluster Analysis

Cluster analysis aims to discover innate groupings of data mainly through manipulation of geometric positioning of items in the hyperspace. Clustering deals with devising solutions to zero-class classification problems, i.e. the case where no supervised information such as class labels is available.

Assume that you are given the training data set $T = \{(\mathbf{x}_i, y_i) \,|\, i \in [1..m]\}$, in which each $\mathbf{x}_i$ is an n-dimensional multivariate vector, and each y_i holds the supervised class information, whose use will be deferred until external validation task, since we are going to conduct unsupervised analysis of the data.

Clusters are denoted to be C_j which are basically sets that hold particular data instances, with j being a valid index. In other words, $C_j = \{\mathbf{x}_{i,q}\}_{q=1}^{M_j}$ where M_j is the number of elements corresponding cluster has, where i is a valid vector index indicating the position of the data instance in the whole data set and q indices the vector within the cluster it is assigned to.

In this section, extensively-used generic k-means algorithm and agglomerative hierarchical clustering method are briefly described.

2.1 K-means

Generic K-means partitions the data set $X = \{\mathbf{x}_i\}_{i=1}^{m}$, where each data element consists of n-dimensional feature vectors into k clusters $C = \{C_j\}_{j=1}^{k}$, whose centers are represented by n-dimensional vectors as in $\mathbf{c}_j = [c_j^d]_{d=1}^{n}, \forall j \in [1..k]$ by minimizing the following objective function.

$$P(U, C) = \sum_{j=1}^{k} \sum_{i=1}^{m} \sum_{d=1}^{n} u_{ij}\, \theta(x_{id}, c_{jd}) \tag{1}$$

with the constraint

$$\sum_{j=1}^{k} u_{ij} = 1, \quad i \in [1..m] \; and \; u_{ij} \in \{0, 1\} \tag{2}$$

In equations (1) and (2), $U = [u_{ij}]$ is the cluster membership matrix such that $u_{ij} = 1$ if $\mathbf{x}_i \in C_j$, and $\theta(x_{id}, c_{jd})$ is the distance between $\mathbf{x}_i$ and C_j on the d^{th} feature. Assuming that all data elements consist of numeric features, squared Euclidean distance between feature attributes is calculated as follows:

$$\theta(x_{id}, c_{jd}) = (x_{id} - c_{jd})^2 \tag{3}$$

Euclidean distance between two vectors of the same dimensionality is calculated according to the following formula.

$$\begin{aligned} \theta(\mathbf{x}_i, \mathbf{c}_j) &= \|\mathbf{x}_i - \mathbf{c}_j\|_2 \\ &= \left(\sum_{d=1}^{n} \theta(x_{id}, c_{jd}) \right)^{1/2} \end{aligned} \tag{4}$$

Optimization problem in (1) is solved for the two parameters separately, assuming that the other one is fixed. First, cluster centers are fixed and cluster membership matrix is updated for all $q \in [1..k]$, as in the following:

$$\begin{cases} u_{ij} = 1 & \text{if } \theta(\mathbf{x}_i, \mathbf{c}_j) \leq \theta(\mathbf{x}_i, \mathbf{c}_q), \forall q \\ u_{iq} = 0 & \text{for } q \neq j \end{cases} \tag{5}$$

Then, cluster membership matrix is fixed and cluster centers are updated as follows:

$$c_{jd} = \frac{\sum_{i=1}^{m} u_{ij}\, x_{id}}{\sum_{i=1}^{m} u_{ij}}, \quad \forall j \in [1..k] \, \forall d \in [1..n] \tag{6}$$

Updating the cluster membership matrix as in (5) involves assigning data elements to clusters whose centers are located at minimum distances. Cluster centers are updated in (6) by taking feature-by-feature means of the data instances they have.

Pseudocode for generic K-means algorithm is included subsequently.

Algorithm 1 Generic K-means

procedure K-MEANS(X, k)
    Initialize C randomly
    $T \leftarrow 0$
    while $U^{(T)} \neq U^{(T-1)} \wedge T \neq I_{max}$ **do**
        Update U according to (5)
        Update C according to (6)
        $T \leftarrow T + 1$
    end while
    return C and U
end procedure

K-means outlined in Algorithm 1 proceeds in cycles until cluster membership matrix stabilizes, i.e. no new assignment occurs at a particular iteration level or a threshold of iterations, I_{max}, is reached. Note that cluster centers are initialized randomly in the traditional algorithm.

2.2 Hierarchical Clustering

Agglomerative, i.e. bottom-up hierarchical clustering methodology aims to form a hierarchy of clusters, commencing with initial singleton ones dedicated to each data point, which are subsequently merged two at a time according to particular linkage criteria until all data instances are gathered in a single cluster. Yielded structure is called a *dendrogram*, a tree structure storing the generated hierarchy of clusters.

Linkage criteria assesses how dissimilar two clusters are. You are going to employ complete linkage and Ward's distance in this assignment.

Given two clusters C_i and C_j having multidimensional data vectors $\{\mathbf{x}_{.,f}\}_{f=1}^{N_i}$ and $\{\mathbf{x}_{.,s}\}_{s=1}^{N_j}$ respectively at any level of the hierarchy such that $i \neq j$, complete linkage criterion evaluates their dissimilarity according to the following formula.

$$\Lambda_c(C_i, C_j) = \max_{\mathbf{x}_{.,f} \in C_i \wedge \mathbf{x}_{.,s} \in C_j} \theta(\mathbf{x}_{.,f}, \mathbf{x}_{.,s}) \tag{7}$$

where $\theta(\mathbf{x}_{.,f}, \mathbf{x}_{.,s}) = \|\mathbf{x}_{.,f} - \mathbf{x}_{.,s}\|_2^2$ is the squared Euclidean distance between data vectors $\mathbf{x}_{.,f}$ and $\mathbf{x}_{.,s}$.

Complete linkage criterion formulated in (7) decides on how distant two clusters are by evaluating the largest distance between the any two members of these clusters.

Ward's minimum variance method imposes a general agglomerative hierarchical clustering procedure in which the linkage criterion is based on the optimal value of an objective function, which is the error sum of squares in this case.

Distance between clusters to be merged C_i and C_j should be computed according to the following formula.

$$\begin{aligned} \Lambda_w(C_i, C_j) = \sum_{\mathbf{x}_{.,r} \in C_i \cup C_j} \|\mathbf{x}_{.,r} - \mathbf{c}_p\|_2^2 - \sum_{f=1}^{M_i} \|\mathbf{x}_{.,f} - \mathbf{c}_i\|_2^2 \\ - \sum_{s=1}^{M_j} \|\mathbf{x}_{.,s} - \mathbf{c}_j\|_2^2 \end{aligned} \tag{8}$$

where $\mathbf{c}_p = (1/(M_i + M_j)) \sum_{\mathbf{x}_{\cdot,r} \in C_i \cup C_j} \mathbf{x}_{\cdot,r}$ as the centroid of the merged cluster, $\mathbf{c}_i = (1/M_i) \sum_{f=1}^{M_i} \mathbf{x}_{\cdot,f}$ as the centroid of C_i and $\mathbf{c}_j = (1/M_j) \sum_{s=1}^{M_j} \mathbf{x}_{\cdot,s}$ as the centroid of C_j.

Ward's linkage criterion formulated in (8) assesses the dissimilarity of two clusters depending on the minimization of within-cluster variances.

Outline of the agglomerative clustering algorithm is included subsequently.

Algorithm 2 Agglomerative Hierarchical Clustering

procedure HIERARCHICAL(X)
    Form singleton clusters, C_i for each $\mathbf{x_i}$
    Form $C \leftarrow \bigcup C_i$
    $T \leftarrow 0$
    while $|C| \neq 1$ **do**
        Find C_i, C_j that minimizes (7) or (8)
        Merge $C_k \leftarrow C_i \cup C_j$
        Remove $C \leftarrow C - \{C_i, C_j\}$
        Add $C \leftarrow C \cup C_k$
        $T \leftarrow T + 1$
    end while
    return C
end procedure

Hierarchical clustering algorithm outlined in Algorithm 2 yields a hierarchy of clusters by iteratively merging clusters that minimizes adopted linkage criterion until all points are gathered in a single cluster. Particular number of clusters can be sought through the appropriate levels in the generated hierarchy.

2.3 Clustering Validation

Cluster analysis conducted in the previous section is not complete without validation phase with respect to several mathematical indices. Assessment via these criteria helps the data analyst to decide on which clustering attains the highest quality.

As class labels are available for your data set, first an external evaluation will be carried out to reveal to which extent generated clusters represent classes formed by domain specialists. For **only** this case, number of clusters should be assumed to be the number of classes.

Assume that you have two distinct partitioning of the data set X as $U^{(1)}$ and $U^{(2)}$. **Class purity index** is the external validation criterion you are going to utilize. For that, first you should map each cluster to the class that occurs most frequently within the cluster. Then the index proceeds the computation according to the subsequent formula.

$$PI(U^{(1)}, U^{(2)}) = \frac{1}{m} \sum_{i=1}^{k} \max_{j} |C_i^{(1)} \cap C_j^{(2)}| \tag{9}$$

where m is the number of elements in the data set, $C_i^{(1)}$ is the cluster indexed by i in the first partition, and $C_j^{(2)}$ is the cluster indexed by j in the second partition.

Purity in (9) is utilized for comparing clusters and classes of any size and takes values in the interval $[0, 1]$, where the highest degree of purity is unity.

Regarding external validation, **variation of information** (VI) is another criterion to be employed. VI relies on the concepts of entropy and mutual information. Entropy of a clustering is computed as follows.

$$H(U) = -\sum_{j=1}^{k} P(j) \log P(j) \tag{10}$$

where $P(j) = M_k/m$ is the probability of encountering a point in cluster C_j.

Entropy in (10) is non-negative and zero only when there is a single cluster. Otherwise, its value depends on the cluster compositions. Mutual information between two partitions is computed in accordance with the subsequent formula.

$$I(U^{(1)}, U^{(2)}) = \sum_{j=1}^{k} \sum_{i=1}^{k} P(j,i) \log \frac{P(j,i)}{P(j)P'(i)} \tag{11}$$

where $P(j,i) = |C_j \cap C_i'|/m$ is the probability of existence of a data point in cluster C_j in the first partition and within cluster C_i' in the second partition.

Entropy assesses uncertainty about the cluster of a random data point in a particular partition, without any other information. Mutual information in (11) measures to which extent this uncertainty reduces given the cluster of that data point in another partition.

VI utilizes entropy and mutual information in the following way.

$$\begin{aligned} VI(U^{(1)}, U^{(2)}) =& [H(U^{(1)}) - I(U^{(1)}, U^{(2)})] + \\ & [H(U^{(2)}) - I(U^{(1)}, U^{(2)})] \\ =& H(U^{(1)}) + H(U^{(2)}) - 2I(U^{(1)}, U^{(2)}) \end{aligned} \tag{12}$$

VI in (12) is a metric driven out of information theory that does not regard any assumptions about how clusterings are generated. In this sense, it can be applied to both crisp and fuzzy clusterings. VI points out the information existing in both partitions that cannot be reduced further via mutual information content.

Clustering validation is also conducted when no labeled information is available, assessing how well-separated and compact generated clusters are. This kind of internal validation requires formulation of particular criteria, out of many **Davies-Bouldin Index** is going to be utilized in this assignment.

Davies-Bouldin index (DBI) requires the computation of within-cluster dispersion, as indicated in the following.

$$S_j = \left(\frac{1}{M_j} \sum_{\mathbf{x}_i \in C_j} \|\mathbf{x}_i - \mathbf{c}_j\|_p^p \right)^{1/p} \tag{13}$$

where p is the coefficient specifying the moment of dispersion. When $p = 1$, average distance of objects within the cluster is computed, and when $p = 2$, standard deviation of within-cluster point distances is

calculated, in accordance with the first and the second statistical moments.

Utilization of within-cluster dispersion in (13) within the DBI metric is as follow.

$$DBI(U) = \frac{1}{k} \sum_{i=1}^{k} \max_{j \neq i} \frac{S_i + S_j}{\Theta_{ij}} \tag{14}$$

where Θ_{ij} is the distance between centroids $\mathbf{c_i}$ and $\mathbf{c_j}$.

DBI metric of a partition averages the cluster-wise maximum dispersion to minimum separation ratio, as in (14). In other words, higher values of this metric indicates that generated clusters are relatively less compact and badly-separated on the average case. Therefore, lower values of the metric are more favorable in deducing clusters of higher qualities.

3 Programming Tasks

This homework invites you to conduct cluster analysis on the given data set, which will be described in detail subsequently.

Training data for this task is included in the file named "sat.data" as Landsat image data dated back to 1993. There are 6435 instances each of which is represented using 36 integer attributes corresponding to pixel readings of the multispectral satellite image patches, followed by another integer variable corresponding to one of the six different soil types as supervised information. Data set is publicly available on UCI Machine Learning Repository, under this link to which you may refer to regarding details.

In this assignment, your aim will be discovering clusters on the data set which are to be evaluated with respect to external and internal validity criteria.

Change the default format of the MATLAB workspace into long fixed-decimal format to avoid possible numerical errors.

First, you need to input the given file by proper MATLAB functions such as **load** command into your own choice of variables to initiate the computation. Use of data design matrix $\mathbf{X}$ together with a vector of class labels (not be used until external validation) is highly recommended.

You may **pre-process** your data set.

Write your own custom MATLAB function that implements k-means clustering algorithm.

Write another function that implements hierarchical clustering under complete linkage and Ward's method. These two functions should be implemented by yourselves. Do **not** use toolbox functions.

Run hierarchical clustering algorithm under the two mentioned linkage schemes. Visualize the clustering you obtained using MATLAB's built-in **dendrogram** command or some publicly-available plotting facility.

Run k-means algorithm for six clusters under different choices of initial clusters. Did the algorithm converge on every case to the same clusters? Discuss the results you obtained.

Using available class labels, conduct external validation with respect to class purity index and variation of information index regarding all three clustering methods. Comment on the obtained results in your report.

Decide at which level the dendrogram may be cut off using Davies-Bouldin Index. For some cluster-initialization methodology, decide the number of clusters for which the most compact and well-separated clusters are generated by k-means algorithm, again using DBI as the internal validity criterion.

4 Restrictions and Tips

- Do not use any available MATLAB repository files without referring to them in your report.
- Toolbox function use is restricted in this homework. You may only use toolbox functions for plotting utilities.
- Vectorization is very important in this task.
- Implementation should be of your own. Readily-used codes should not exceed a reasonable threshold within your total work.
- Don't forget that the code you are going to submit will also be subject to manual inspection.

5 Submission

- **Late Submission:** No late submission is accepted.
- Your scripts and function files together with a 3-to-4 pages long report focusing on theoretical and practical aspects you observed regarding this task should be uploaded on COW before the specified deadline as a compressed archive file whose name should be <<student_id>_hw3> preceding the file extension.
- The archive must contain **no directories** on top of MATLAB/Octave scripts and function files.

6 Regulations

1. **Cheating: We have zero tolerance policy for cheating**. People involved in cheating will be punished according to the university regulations.
2. **Newsgroup:** You must follow the newsgroup (news.ceng.metu.edu.tr) for discussions and possible updates on a daily basis.