Discovering Knowledge in Data

Daniel T. Larose, Ph.D.

Chapter 10
Hierarchical and *k*-Means Clustering

Prepared by James Steck and Eric Flores

Clustering Task

- Clustering refers to grouping records, observations, or cases into classes of similar objects
- Cluster is a collection of records similar to one another
- Records in one cluster are dissimilar to records in other clusters
- Clustering is an unsupervised data mining task. Therefore, no target variable is specified
- Clustering algorithms segment records and maximize homogeneity in subgroups (or segments)
- Similarity to records outside of a cluster is minimized

Clustering Task *(cont'd)*

- For example, Nielsen PRIZM, developed by Claritas, Inc. provides demographic profiles of geographic areas, according to zip code
- PRIZM segmentation system clusters zip codes in terms of lifestyle types. One zip code might belong to more than one cluster.
- Example: Clusters identified for 90210 Beverly Hills, CA
  - Cluster 01: *Upper Crust Estates*

    "The nation's most exclusive address, Upper Crust is the wealthiest lifestyle in America, a Haven for empty-nesting couples between the ages of 45 and 64. No segment has a higher concentration of residents earning over $100,000 a year and possessing a postgraduate degree. And none has a more opulent standard of living."
  - Cluster 03: Movers and Shakers
  - Cluster 04: *Young Digerati*
  - Cluster 07: *Money and Brains*
  - Cluster 16: *Bohemian Mix*

Clustering Task *(cont'd)*

- Clustering Tasks in Business and Research
  - Target marketing of a niche product for a small business without large marketing budget
  - Accounting auditing: Segment financial behavior into benign and suspicious categories
  - As a dimension-reduction tool when data set has hundreds of attributes
  - Gene expression clustering, where very large quantities of genes exhibit similar characteristics
- Clustering is often performed as a preliminary step in a data mining process
- Resulting clusters are used as inputs into other data mining techniques such as Neural Networks

Clustering Task *(cont'd)*

- Applying cluster analysis to enormous databases is helpful
- This Reduces the search space for downstream algorithms

- Cluster analysis encounters similar issues faced in classification
  - How to measure similarity
  - How to recode categorical variables
  - How to standardize or normalize numerical variables
  - How many clusters we expect to uncover

Clustering Task *(cont'd)*

- Measuring Similarity
  - In this book, we use Euclidean Distance that measures distance between records

$$d_{\text{Euclidean}}(\mathbf{x},\mathbf{y}) = \sqrt{\sum_i (x_i - y_i)^2}$$

where $\mathbf{x} = x_1, x_2, \ldots, x_m$ and $\mathbf{y} = y_1, y_2, \ldots, y_m$

represent the m attribute values of two records

  - Other distance measurements include City-Block Distance and Minkowski Distance (q is a general exponent)

$$d_{\text{City-Block}}(\mathbf{x},\mathbf{y}) = \sum_i |x_i - y_i|$$

$$d_{\text{Minkowski}}(\mathbf{x},\mathbf{y}) = \sum_i |x_i - y_i|^q$$

Clustering Task *(cont'd)*

- For categorical variables, we use "Different From" function for comparing the i-th attribute values of a pair of records:

$$\text{different}(x_i, y_i) = \begin{cases} 0 & \text{if } x_i = y_i \\ 1 & \text{otherwise} \end{cases}$$

- We may substitute *different(x,y)* for each categorical attribute in Euclidean Distance function
- Normalizing data enhances performance of clustering algorithms
- Use Min-max Normalization or Z-Score Standardization

$$\text{Z - Score Standardization} = \frac{X - \text{mean}(X)}{\text{standard deviation}(X)}$$

$$\text{Min - Max Normalization} = \frac{X - \min(X)}{\max(X) - \min(X)}$$

Clustering Task *(cont'd)*

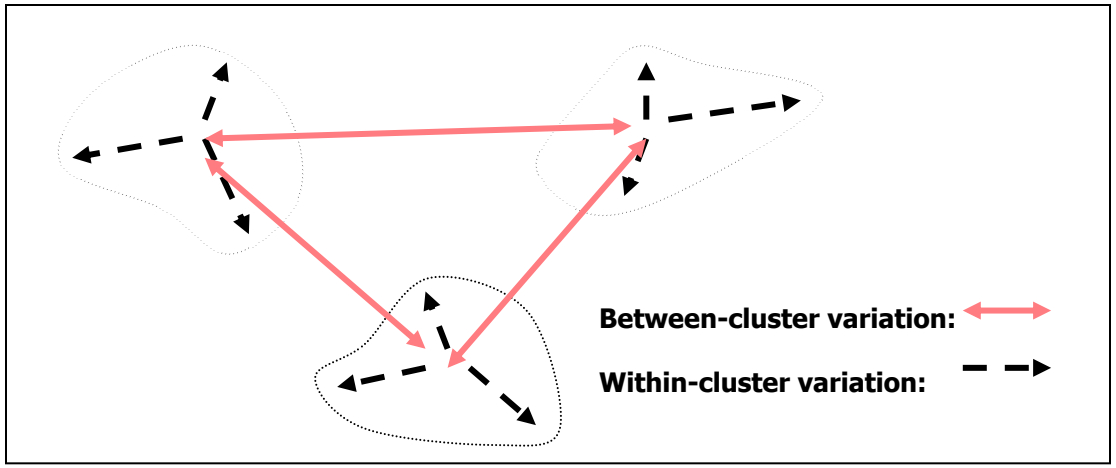


Figure 10.1

- Clustering identifies groups of highly-similar records
- Algorithms construct clusters where between-cluster variation (BCV) is large compared to within-cluster variation (WCV)

Hierarchical Clustering Methods

- Clustering algorithms either Hierarchical or Non-Hierarchical

- Hierarchical
  - Treelike cluster structure (dendogram) is created through recursive partitioning (Divisive Methods) or combining (Agglomerative Methods) existing clusters
  - Divisive Methods:
  - All records initialized into single cluster
  - At each iteration, most dissimilar records are split off into a separate cluster (one of the clusters is split into two clusters)
  - Eventually, each record represents its own cluster

Hierarchical Clustering Methods *(cont'd)*

- Agglomerative Methods
- Each observation initialized to become its own cluster
- At each iteration, two closest clusters are aggregated together
- Number of clusters reduced by one after each step
- Eventually, all records are combined into a single huge cluster

- Agglomerative methods are more popular hierarchical methods, therefore, we focus on this approach.

- Measuring distance between records is straightforward once recoding and normalization are applied
- However, how is the distance between clusters determined?

Hierarchical Clustering Methods *(cont'd)*

- Distance Between Clusters
  - Several criteria examined to determine distance between clusters A and B
  - Single Linkage
  - Known as Nearest-Neighbor Approach
  - Minimum distance between any record in cluster A and any record in cluster B
  - Cluster similarity is based on the most similar records from each cluster
  - Tends to form long, slender clusters
  - Sometime heterogeneous records are clustered together

Hierarchical Clustering Methods *(cont'd)*

- Complete Linkage
- Known as Farthest-Neighbor Approach
- Maximum distance between any record in cluster A and any record in cluster B
- Cluster similarity based on the most dissimilar records from each cluster
- Tends to form compact, sphere-like clusters
- All records in cluster are within given diameter of other records

- Average Linkage
- Designed to reduce dependence of cluster-linkage to extreme values, such as most similar or dissimilar records

Hierarchical Clustering Methods *(cont'd)*

- Measure is the average distance of records in cluster A from records in cluster B
- Resulting clusters have approximately equal within-cluster variability

k-Means Clustering

- *k*-Means clustering algorithm is effective at finding clusters in data

- *k*-Means Algorithm
  - Step 1: Analyst specifies *k* = number of clusters to partition data into.
  - Step 2: Randomly assign k records to be the initial cluster center locations.
  - Step 3: Each record is assigned to its nearest cluster center. Each cluster center "owns" a subset of the records resulting in *k* clusters, C_1, C_2, ..., C_k
  - Step 4: For each of the *k* clusters, find the cluster centroid.

    Then, update each cluster center location to the new value of the centroid
  - Step 5: Repeats Steps 3 – 5 until convergence or termination

k-Means Clustering *(cont'd)*

- Nearest criterion in Step 3 is typically Euclidean Distance

- Determining Cluster Centroid
  - Assume *n* data points (a_1, b_1, c_1), (a_2, b_2, c_2), ..., (a_n, b_n, c_n)
  - Centroid of these points is center of gravity of these points located at point $(\Sigma\, a_i/n, \Sigma\, b_i/n, \Sigma\, c_i/n)$
  - For example, points (1, 1, 1), (1, 2, 1), (1, 3, 1), and (2, 1, 1) have centroid

$$\left(\frac{1+1+1+2}{4}, \frac{1+2+3+1}{4}, \frac{1+1+1+1}{4}\right) = (1.25, 1.75, 1.00)$$

k-Means Clustering *(cont'd)*

- *Termination*
  - *k*-Means algorithm terminates when centroids no longer change. In other words, for *k* clusters, C_1, C_2, ..., C_k, all records "owned" by each cluster center remain in that cluster.
  - Convergence criterion may also cause termination
  - For example, no significant reduction in mean squared error (MSE):

$$MSE = \frac{SSE}{N-k} = \frac{\sum_{i=1}^{k} \sum_{p \in C_i} d(p, m_i)^2}{N-k}$$

where *SSE* represents the *sum of squares error*, $p \in C_i$ represents each data point in cluster *i*, m_i represents the centroid (cluster center) of cluster *i*, *N* is the total sample size and *k* is the number of clusters

k-Means Clustering *(cont'd)*

- Clustering algorithms seek to construct clusters of records such that the between-cluster variation is large compared to the within-cluster variation
- Analogous to Analysis of Variance (ANOVA), we define a *pseudo-F statistic*:

$$F_{k-1,N-k} = \frac{MSB}{MSE} = \frac{SSB/k-1}{SSE/N-k}$$

where *MSE* is defined as above, *MSB* is the *mean square between*, and *SSB* is the *sum of squares between* clusters, defined as

$$SSB = \sum_{i=1}^{k} n_i \cdot d(m_i, M)^2$$

where n_i is the number of records in cluster *i*, m_i is the centroid of cluster *i*, and *M* is the grand mean of all the data

k-Means Clustering *(cont'd)*

- MSB represents the between-cluster variation and MSE represents the within-cluster variation
- "Good" cluster has large pseudo-F statistic – where the between cluster variation is large, compared to within-cluster variation
- As K-means algorithm proceeds, MSB increases, MSE decreases, and F increases

Example of *k*-Means Clustering at Work

- Assume $k = 2$ to cluster the following data points

a	b	c	d	e	f	g	h
(1, 3)	(3, 3)	(4, 3)	(5, 3)	(1, 2)	(4, 2)	(1, 1)	(2, 1)

- Step 1: $k = 2$ specifies number of clusters to partition
- Step 2: Randomly assign $k = 2$ records to be the initial cluster centers

  For example, $m_1 = (1, 1)$ and $m_2 = (2, 1)$

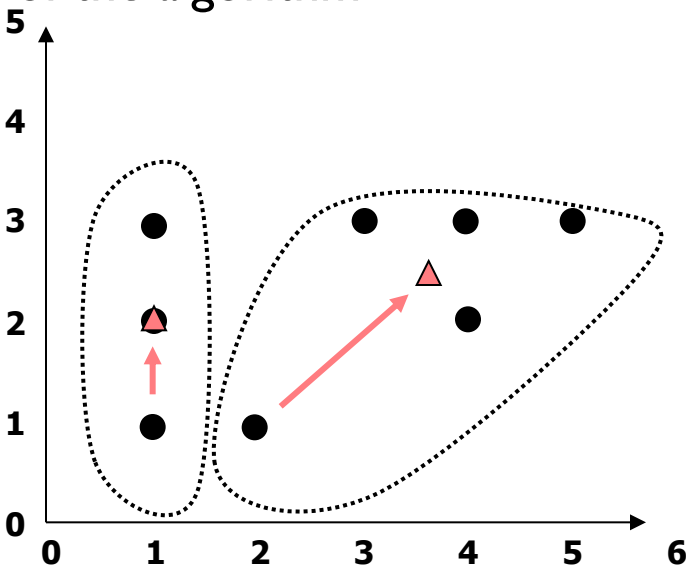
- First Iteration
  - Step 3: For each record, find nearest cluster center

    Euclidean distance from points to m_1 and m_2 shown

Point	a	b	c	d	e	f	g	h
Distance from m_1	2.00	2.83	3.61	4.47	1.00	3.16	0.00	1.00
Distance from m_2	2.24	2.24	2.83	3.61	1.41	2.24	1.00	0.00
Cluster Membership	C_1	C_2	C_2	C_2	C_1	C_2	C_1	C_2

Example of *k*-Means Clustering at Work *(cont'd)*

- Step 4: For each of the *k* clusters, find the cluster centroid, then update the location of each cluster center to the new centroid
- Cluster 1 = [(1 + 1 + 1)/3, (3 + 2 + 1)/3] = (1, 2), Cluster 2 = [(3 + 4 + 5 + 4 + 2)/5, (3 + 3 + 3 + 2 + 1)/5] = (3.6, 2.4)
- Figure shows the movement of cluster centers m*1* and m*2* (triangles) after first iteration of the algorithm



Example of *k*-Means Clustering at Work *(cont'd)*

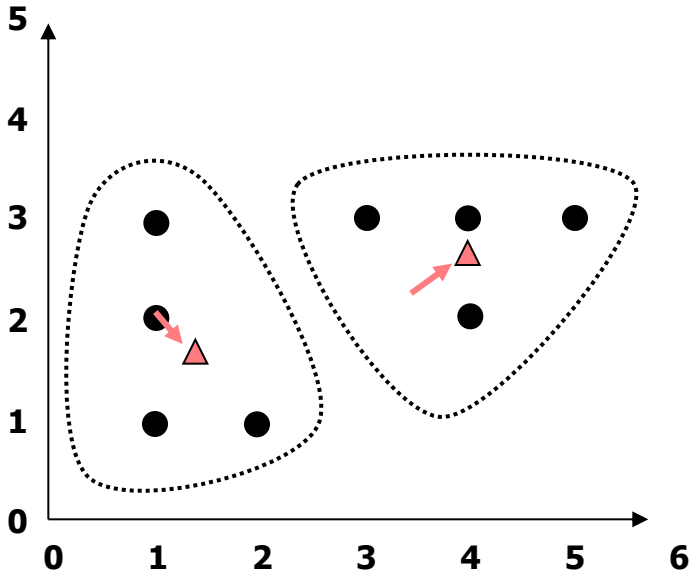
- Step 5: Repeats Steps 3 – 4 until convergence or termination

- Second Iteration
  - Repeat procedure for Steps 3 – 4
  - Again, for each record find nearest cluster center m_1 = (1, 2) or m_2 = (3.6, 2.4)
  - Table below shows how *h* moved to cluster 1

Point	a	b	c	d	e	f	g	h
Distance from m_1	1.00	2.24	3.16	4.12	0.00	3.00	1.00	1.41
Distance from m_2	2.67	0.85	0.72	1.52	2.63	0.57	2.95	2.13
Cluster Membership	C_1	C_2	C_2	C_2	C_1	C_2	C_1	C_1

Example of *k*-Means Clustering at Work *(cont'd)*

- Cluster centroids updated to m_1 = (1.25, 1.75) or m_2 = (4, 2.75)
- After Second Iteration, cluster centroids shown to move slightly



Example of *k*-Means Clustering at Work *(cont'd)*

- Third (Final) Iteration
  - Repeat procedure for Steps 3 – 4
  - Now, for each record find nearest cluster center m_1 = (1.25, 1.75) or m_2 = (4, 2.75)
  - This time, no records shift cluster membership
  - Centroids remain unchanged, therefore algorithm terminates