# Lecture 8 - Hierarchical and k-means Clustering

Adnan Ferdous Ashrafi

Stamford University Bangladesh

# Table of Contents

# Clustering task- Definition

*Clustering* refers to the grouping of records, observations, or cases into classes of similar objects. A cluster is a collection of records that are similar to one another and dissimilar to records in other clusters.

Clustering differs from classification in that there is no target variable for clustering. The clustering task does not try to classify, estimate, or predict the value of a target variable.

Instead, clustering algorithms seek to segment the entire data set into relatively homogeneous subgroups or clusters, where the similarity of the records within the cluster is maximized, and the similarity to records outside this cluster is minimized.

# Clustering task- Example

For example, Claritas, Inc. is a clustering business that provides demographic profiles of each geographic area in the United States, as defined by zip code. One of the clustering mechanisms they use is the PRIZM segmentation system, which describes every U.S. zip code area in terms of distinct lifestyle types. Recall, for example, that the clusters identified for zip code 90210, Beverly Hills, California, were:

- Cluster 01: Blue Blood Estates
- Cluster 10: Bohemian Mix
- Cluster 02: Winner's Circle
- Cluster 07: Money and Brains
- Cluster 08: Young Literati

# Clustering task- Example

The description for cluster 01: Blue Blood Estates is “Established executives, professionals, and ‘old money’ heirs that live in America’s wealthiest suburbs. They are accustomed to privilege and live luxuriously—one-tenth of this group’s members are multimillionaires. The next affluence level is a sharp drop from this pinnacle.”

# Clustering task- Application of Clustering

Examples of clustering tasks in business and research include:

- Target marketing of a niche product for a small-capitalization business that does not have a large marketing budget
- For accounting auditing purposes, to segment financial behavior into benign and suspicious categories
- As a dimension-reduction tool when a data set has hundreds of attributes
- For gene expression clustering, where very large quantities of genes may exhibit similar behavior

Clustering is often performed as a preliminary step in a data mining process, with the resulting clusters being used as further inputs into a different technique downstream, such as neural networks. Due to the enormous size of many present-day databases, it is often helpful to apply clustering analysis first, to reduce the search space for the downstream algorithms.

# Issues in clustering

Cluster analysis encounters many of the same issues that we dealt with in the chapters on classification. For example, we shall need to determine:

- How to measure similarity
- How to recode categorical variables
- How to standardize or normalize numerical variables
- How many clusters we expect to uncover

# Distance Measure- Continuous Variables

For simplicity, in this course we concentrate on *Euclidean distance* between records:

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

where $x = x_1, x_2, \dots, x_m$, and $y = y_1, y_2, \dots, y_m$ represent the $m$ attribute values of two records. Of course, many other metrics exist, such as *city-block* distance:

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

or *Minkowski distance*, which represents the general case of the foregoing two metrics for a general exponent $q$:

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

# Distance Measure- Categorical Variables

For categorical variables, we may again define the “different from” function for comparing the ith attribute values of a pair of records:

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

where $x_i$ and $y_i$ are categorical values. We may then substitute $\text{different}(x_i, y_i)$ for the $i^{\text{th}}$ term in the Euclidean distance metric above.

# Distance Measure- Condition of clustering

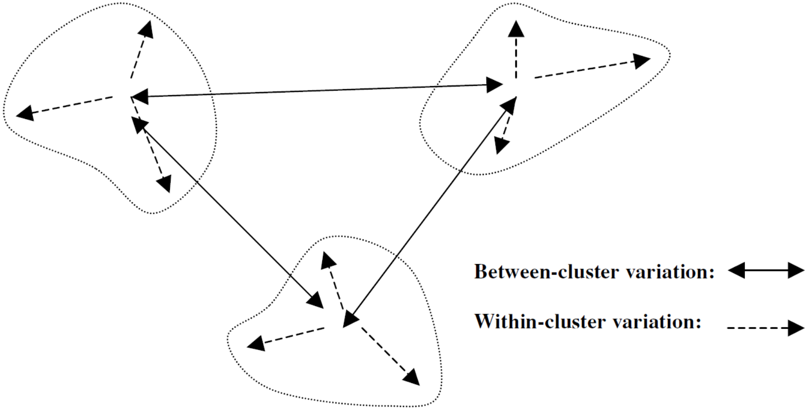


**Figure 1:** Clusters should have small within-cluster variation compared to the between cluster variation.

# Optimal performance in clustering

For optimal performance, clustering algorithms, just like algorithms for classification, require the data to be normalized so that no particular variable or subset of variables dominates the analysis. Analysts may use either the *min–max normalization* or *Z-score standardization*, discussed in earlier chapters:

$$\textit{Min-max Normalization:} \quad X^* = \frac{X - \min(x)}{\text{Range}(x)}$$

$$\textit{Z-score standardization:} \quad X^* = \frac{X - \text{mean}(x)}{\text{SD}(x)}$$

All clustering methods have as their goal the identification of groups of records such that similarity within a group is very high while the similarity to records in other groups is very low.

In other words, as shown in Figure 1, clustering algorithms seek to construct clusters of records such that the between-cluster variation (BCV) is large compared to the within-cluster variation (WCV) This is somewhat analogous to the concept behind analysis of variance.

# Hierarchical Clustering Methods- Definition

Clustering algorithms are either hierarchical or nonhierarchical.

## Definition

In hierarchical clustering, a treelike cluster structure (dendrogram) is created through recursive partitioning (divisive methods) or combining (agglomerative) of existing clusters.

# Hierarchical Clustering Methods- Agglomerative and Divisive clustering methods

*Agglomerative clustering methods* [1] initialize each observation to be a tiny cluster of its own. Then, in succeeding steps, the two closest clusters are aggregated into a new combined cluster. In this way, the number of clusters in the data set is reduced by one at each step. Eventually, all records are combined into a single huge cluster.

*Divisive clustering methods* begin with all the records in one big cluster, with the most dissimilar records being split off recursively, into a separate cluster, until each record represents its own cluster.

Because most computer programs that apply hierarchical clustering use agglomerative methods, we focus on those.

# Agglomerative methods- Distance between clusters

Distance between records is rather straightforward once appropriate re-coding and normalization has taken place.

**But how do we determine distance between clusters of records?**

**Should we consider two clusters to be close if their nearest neighbors are close or if their farthest neighbors are close?**

**How about criteria that average out these extremes?**

We examine several criteria for determining distance between arbitrary clusters A and B:

# Agglomerative methods- Single Linkage

*Single linkage*, sometimes termed the *nearest-neighbor approach*, is based on the minimum distance between any record in cluster A and any record in cluster B. **In other words, cluster similarity is based on the similarity of the most similar members from each cluster.**

Single linkage tends to form long, slender clusters, which may sometimes lead to heterogeneous records being clustered together.

# Agglomerative methods- Complete Linkage

*Complete linkage*, sometimes termed the *farthest-neighbor approach*, is based on the maximum distance between any record in cluster A and any record in cluster B.**In other words, cluster similarity is based on the similarity of the most dissimilar members from each cluster.**

Complete-linkage tends to form more compact, sphere-like clusters, with all records in a cluster within a given diameter of all other records.

# Agglomerative methods- Average Linkage

*Average linkage* is designed to reduce the dependence of the cluster-linkage criterion on extreme values, such as the most similar or dissimilar records. **In average linkage, the criterion is the average distance of all the records in cluster A from all the records in cluster B.**

The resulting clusters tend to have approximately equal within-cluster variability.

# Agglomerative methods- Example

Let's examine how these linkage methods work, using the following small, one-dimensional data set:

2 5 9 15 16 18 25 33 33 45

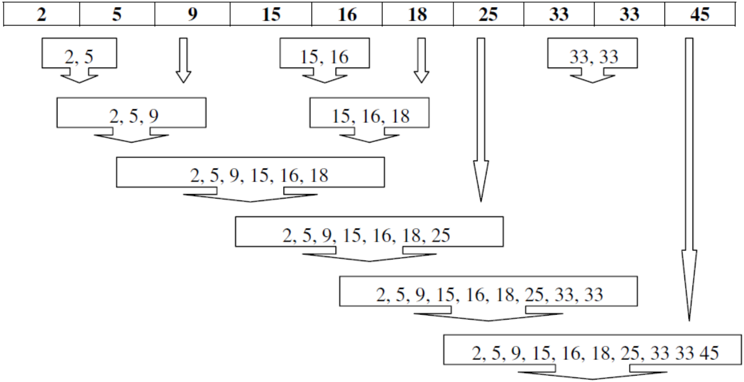
# Agglomerative methods- Example- Single Linkage

Suppose that we are interested in using single-linkage agglomerative clustering on this data set. Agglomerative methods start by assigning each record to its own cluster. Then, single linkage seeks the minimum distance between any records in two clusters.

The minimum cluster distance is clearly between the single-record clusters which each contain the value 33, for which the distance must be zero for any valid metric. Thus, these two clusters are combined into a new cluster of two records, both of value 33, as shown in Figure 2.

Note that, after step 1, only nine (n - 1) clusters remain. Next, in step 2, the clusters containing values 15 and 16 are combined into a new cluster, since their distance of 1 is the minimum between any two clusters remaining.

# Agglomerative methods- Example- Single Linkage



| 2 | 5 | 9 | 15 | 16 | 18 | 25 | 33 | 33 | 45 |
|---|---|---|---|---|---|---|---|---|---|

2, 5

15, 16

33, 33

2, 5, 9

15, 16, 18

2, 5, 9, 15, 16, 18

2, 5, 9, 15, 16, 18, 25

2, 5, 9, 15, 16, 18, 25, 33, 33

2, 5, 9, 15, 16, 18, 25, 33 33 45

**Figure 2:** Single-linkage agglomerative clustering on the sample data set.

# Agglomerative methods- Example- Single Linkage

Here are the remaining steps:

- Step 3: The cluster containing values 15 and 16 (cluster {15,16}) is combined with cluster {18}, since the distance between 16 and 18 (the closest records in each cluster) is two, the minimum among remaining clusters.
- Step 4: Clusters {2} and {5} are combined.
- Step 5: Cluster {2,5} is combined with cluster {9}, since the distance between 5 and 9 (the closest records in each cluster) is four, the minimum among remaining clusters.

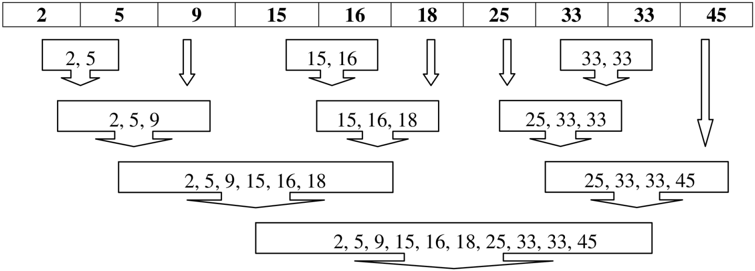
# Agglomerative methods- Example- Single Linkage

- Step 6: Cluster {2,5,9} is combined with cluster {15,16,18}, since the distance between 9 and 15 is six, the minimum among remaining clusters.
- Step 7: Cluster {2,5,9,15,16,18} is combined with cluster {25}, since the distance between 18 and 25 is seven, the minimum among remaining clusters.
- Step 8: Cluster {2,5,9,15,16,18,25} is combined with cluster {33,33}, since the distance between 25 and 33 is eight, the minimum among remaining clusters.
- Step 9: Cluster {2,5,9,15,16,18,25,33,33} is combined with cluster {45}. This last cluster now contains all the records in the data set.

# Agglomerative methods- Example- Complete-Linkage

Next, let's examine whether using the complete-linkage criterion would result in a different clustering of this sample data set. Complete linkage seeks to minimize the distance among the records in two clusters that are farthest from each other. Figure 3 illustrates complete-linkage clustering for this data set.

# Agglomerative methods- Example- Complete-Linkage



| 2 | 5 | 9 | 15 | 16 | 18 | 25 | 33 | 33 | 45 |
|---|---|---|---|---|---|---|---|---|---|

2, 5

15, 16

33, 33

2, 5, 9

15, 16, 18

25, 33, 33

2, 5, 9, 15, 16, 18

25, 33, 33, 45

2, 5, 9, 15, 16, 18, 25, 33, 33, 45

**Figure 3:** Complete-linkage agglomerative clustering on the sample data set.

# Agglomerative methods- Example- Complete-Linkage

The steps are as follows:

- Step 1: Since each cluster contains a single record only, there is no difference between single linkage and complete linkage at step 1. The two clusters each containing 33 are again combined.
- Step 2: Just as for single linkage, the clusters containing values 15 and 16 are combined into a new cluster. Again, this is because there is no difference in the two criteria for single-record clusters.
- Step 3: At this point, complete linkage begins to diverge from its predecessor. In single linkage, cluster {15,16} was at this point combined with cluster {18}. But complete linkage looks at the farthest neighbors, not the nearest neighbors. The farthest neighbors for these two clusters are 15 and 18, for a distance of 3. This is the same distance separating clusters {2} and {5}. The complete-linkage criterion is silent regarding ties, so we arbitrarily select the first such combination found, therefore combining the clusters {2} and {5} into a new cluster.

# Agglomerative methods- Example- Complete-Linkage

- Step 4: Now cluster {15,16} is combined with cluster {18}.
- Step 5: Cluster {2,5} is combined with cluster {9}, since the complete-linkage distance is 7, the smallest among remaining clusters.
- Step 6: Cluster {25} is combined with cluster {33,33}, with a complete-linkage distance of 8.
- Step 7: Cluster {2,5,9} is combined with cluster {15,16,18}, with a complete-linkage distance of 16.
- Step 8: Cluster {25,33,33} is combined with cluster {45}, with a complete-linkage distance of 20.
- Step 9: Cluster {2,5,9,15,16,18} is combined with cluster {25,33,33,45}. All records are now contained in this last large cluster.

# Agglomerative methods- Example- Average-Linkage

Finally, with average linkage, the criterion is the average distance of all the records in cluster A from all the records in cluster B. Since the average of a single record is the record's value itself, this method does not differ from the earlier methods in the early stages, where single-record clusters are being combined.

# Agglomerative methods- Example- Average-Linkage

At step 3, average linkage would be faced with the choice of combining clusters {2} and {5}, or combining the {15, 16} cluster with the single-record {18} cluster. The average distance between the {15, 16} cluster and the {18} cluster is the average of $|18 - 15|$ and $|18 - 16|$, which is 2.5, while the average distance between clusters {2} and {5} is of course 3. Therefore, average linkage would combine the {15, 16} cluster with cluster {18} at this step, followed by combining cluster {2} with cluster {5}.

# Agglomerative methods- Example- Average-Linkage

The reader may verify that the average-linkage criterion leads to the same hierarchical structure for this example as the complete-linkage criterion. In general, average linkage leads to clusters more similar in shape to complete linkage than does single linkage.

# k-MEANS CLUSTERING- Definition

*K-means* [2] defines a prototype in terms of a centroid, which is usually the mean of a group of points, and is typically applied to objects in a continuous n-dimensional space. Interestingly, a centroid almost never corresponds to an actual data point. In this section, we will focus on K-means, which is one of the oldest and most widely used clustering algorithms.

# k-MEANS CLUSTERING- Algorithm

The k-means clustering algorithm [2] is a straightforward and effective algorithm for finding clusters in data. The algorithm proceeds as follows.

- *Step 1*: Ask the user how many clusters k the data set should be partitioned into.
- *Step 2*: Randomly assign $k$ records to be the initial cluster center locations.
- *Step 3*: For each record, find the nearest cluster center. Thus, in a sense, each cluster center "owns" a subset of the records, thereby representing a partition of the data set. We therefore have $k$ clusters, $C_1, C_2, \ldots, C_k$.
- *Step 4*: For each of the $k$ clusters, find the cluster centroid, and update the location of each cluster center to the new value of the centroid.
- *Step 5*: Repeat steps 3 to 5 until convergence or termination.

# k-MEANS CLUSTERING- Algorithm

The “nearest” criterion in step 3 is usually Euclidean distance, although other criteria may be applied as well. The cluster centroid in step 4 is found as follows. Suppose that we have $n$ data points $(a_1, b_1, c_1), (a_2, b_2, c_2), \ldots, (a_n, b_n, c_n)$, the centroid of these points is the center of gravity of these points and is located at point $(\sum a_i/n, \sum b_i/n, \sum c_i/n)$.

For example, the points (1,1,1), (1,2,1), (1,3,1), and (2,1,1) would 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- Algorithm

The algorithm terminates when the centroids no longer change. In other words, the algorithm terminates when for all clusters $C_1, C_2, \ldots, C_k$, all the records "owned" by each cluster center remain in that cluster. Alternatively, the algorithm may terminate when some convergence criterion is met, such as no significant shrinkage in the sum of squared errors:

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

where $p \in C_i$ represents each data point in cluster $i$ and $m_i$ represents the centroid of cluster $i$.

# Example of k-means clustering at work- Problem statement

Let’s examine an example of how the k-means algorithm works. Suppose that we have the eight data points in two-dimensional space shown in Figure 4 and plotted in Figure 5 and are interested in uncovering k = 2 clusters.

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

**Figure 4:** Data Points for k-Means Example

# Example of k-means clustering at work- Problem statement

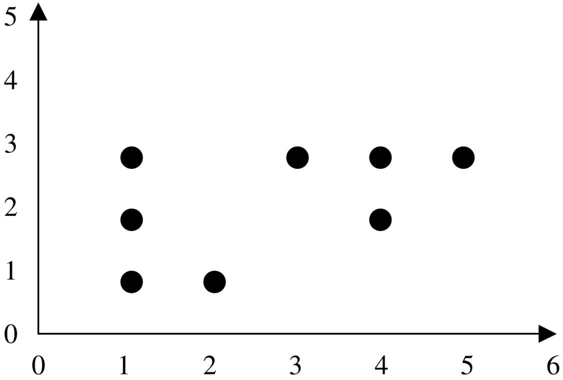


**Figure 5:** How will k-means partition this data into k = 2 clusters?

# Example of k-means clustering at work- Applying k-means

Let's apply the k-means algorithm step by step.

- Step 1: Ask the user how many clusters k the data set should be partitioned into. We have already indicated that we are interested in k = 2 clusters.
- Step 2: Randomly assign k records to be the initial cluster center locations. For this example, we assign the cluster centers to be $m_1 = (1,1)$ and $m_2 = (2,1)$.
- Step 3 (first pass): For each record, find the nearest cluster center. Table 8.2 contains the (rounded) Euclidean distances between each point and each cluster center $m_1 = (1,1)$ and $m_2 = (2,1)$, along with an indication of which cluster center the point is nearest to. Therefore, cluster 1 contains points {a,e,g}, and cluster 2 contains points {b,c,d,f,h}. Once cluster membership is assigned, the sum of squared errors may be found:

$$\begin{aligned}\text{SSE} &= \sum_{i=1}^{k} \sum_{p \in C_i} d(p, m_i)^2 \\ &= 2^2 + 2.24^2 + 2.83^2 + 3.61^2 + 1^2 + 2.24^2 + 0^2 + 0^2 = 36\end{aligned}$$

# Example of k-means clustering at work- Applying k-means

- As remarked earlier, we would like our clustering methodology to maximize the between-cluster variation with respect to the within-cluster variation. Using $d(m_1, m_2)$ as a surrogate for BCV and SSE as a surrogate for WCV, we have:

$$\frac{\text{BCV}}{\text{WCV}} = \frac{d(m_1, m_2)}{\text{SSE}} = \frac{1}{36} = 0.0278$$

  We expect this ratio to increase with successive passes.

# Example of k-means clustering at work- Applying k-means

- Step 4 (first pass): For each of the k clusters find the cluster centroid and update the location of each cluster center to the new value of the centroid. The centroid for cluster 1 is [(1 + 1 + 1) /3, (3 + 2 + 1) /3] = (1,2). The centroid for cluster 2 is [(3 + 4 + 5 + 4 + 2) /5, (3 + 3 + 3 + 2 + 1) /5] = (3.6, 2.4). The clusters and centroids (triangles) at the end of the first pass are shown in Figure 7. Note that m 1 has moved up to the center of the three points in cluster 1, while m 2 has moved up and to the right a considerable distance, to the center of the five points in cluster 2.
- Step 5: Repeat steps 3 and 4 until convergence or termination. The centroids have moved, so we go back to step 3 for our second pass through the algorithm.

# Example of k-means clustering at work- Applying k-means

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

**Figure 6:** Finding the Nearest Cluster Center for Each Record (First Pass)

# Example of k-means clustering at work- Applying k-means

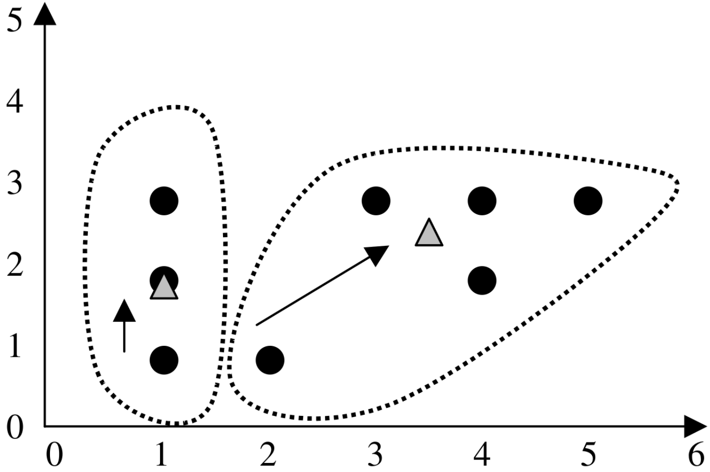


**Figure 7:** Clusters and centroids Δ after first pass through k-means algorithm.

# Example of k-means clustering at work- Applying k-means

- Step 3 (second pass): For each record, find the nearest cluster center. Table 8.3 shows the distances between each point and each updated cluster center $m_1 = (1,2)$ and $m_2 = (3.6,2.4)$, together with the resulting cluster membership. There has been a shift of a single record (h) from cluster 2 to cluster 1. The relatively large change in m 2 has left record *h* now closer to $m_1$ than to $m_2$ , so that record *h* now belongs to cluster 1. All other records remain in the same clusters as previously. Therefore, cluster 1 is {a,e,g,h}, and cluster 2 is {b,c,d,f}. The new sum of squared errors is

$$\begin{aligned} \text{SSE} &= \sum_{i=1}^{k} \sum_{p \in C_i} d(p, m_i)^2 \\ &= 1^2 + 0.85^2 + 0.72^2 + 1.52^2 + 0^2 + 0.57^2 + 1^2 + 1.41^2 \\ &= 7.88 \end{aligned}$$

which is much reduced from the previous SSE of 36, indicating a better clustering solution.

# Example of k-means clustering at work- Applying k-means

We also have:

$$\frac{\text{BCV}}{\text{WCV}} = \frac{d(m_1, m_2)}{\text{SSE}} = \frac{2.63}{7.88} = 0.3338$$

which is larger than the previous 0.0278, indicating that we are increasing the between-cluster variation with respect to the within-cluster variation.

- Step 4 (second pass): For each of the k clusters, find the cluster centroid and update the location of each cluster center to the new value of the centroid. The new centroid for cluster 1 is [(1 + 1 + 1 + 2)/4, (3 + 2 + 1 + 1)/4] = (1.25, 1.75). The new centroid for cluster 2 is [(3 + 4 + 5 + 4)/4, (3 + 3 + 3 + 2)/4] = (4, 2.75). The clusters and centroids at the end of the second pass are shown in Figure 8.6. Centroids $m_1$ and $m_2$ have both moved slightly.

# Example of k-means clustering at work- Applying k-means

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

**Figure 8:** Finding the Nearest Cluster Center for Each Record (second Pass)

# Example of k-means clustering at work- Applying k-means

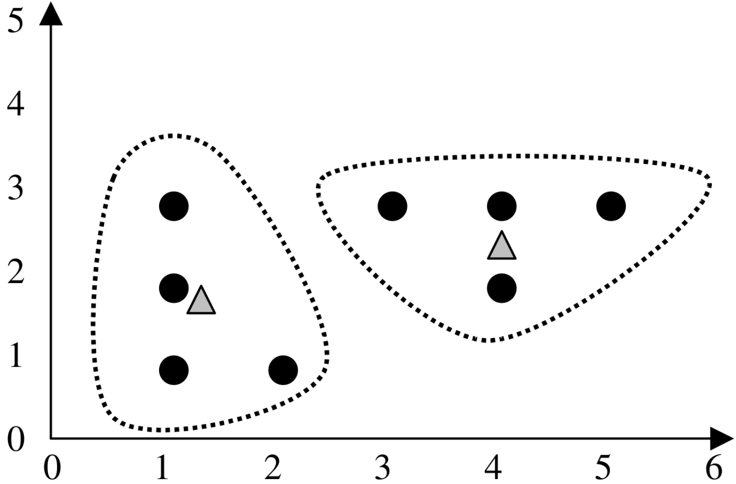


**Figure 9:** Clusters and centroids Δ after second pass through k-means algorithm.

# Example of k-means clustering at work- Applying k-means

- Step 5: Repeat steps 3 and 4 until convergence or termination. Since the centroids have moved, we once again return to step 3 for our third (and as it turns out, final) pass through the algorithm.
- Step 3 (third pass): For each record, find the nearest cluster center. Table 8.4 shows the distances between each point and each newly updated cluster center $m_1 = (1.25, 1.75)$ and $m_2 = (4, 2.75)$, together with the resulting cluster membership. Note that no records have shifted cluster membership from the preceding pass.

# Example of k-means clustering at work- Applying k-means

| Point | Distance from $m_1$ | Distance from $m_2$ | Cluster Membership |
|---|---|---|---|
| $a$ | 1.27 | 3.01 | $C_1$ |
| $b$ | 2.15 | 1.03 | $C_2$ |
| $c$ | 3.02 | 0.25 | $C_2$ |
| $d$ | 3.95 | 1.03 | $C_2$ |
| $e$ | 0.35 | 3.09 | $C_1$ |
| $f$ | 2.76 | 0.75 | $C_2$ |
| $g$ | 0.79 | 3.47 | $C_1$ |
| $h$ | 1.06 | 2.66 | $C_2$ |

**Figure 10:** Clusters and centroids $\Delta$ after third pass through k-means algorithm.

# Example of k-means clustering at work- Applying k-means

The new sum of squared errors is

$$\begin{aligned}
\text{SSE} &= \sum_{i=1}^{k} \sum_{p \in C_i} d(p, m_i)^2 \\
&= 1.27^2 + 1.03^2 + 0.25^2 + 1.03^2 + 0.35^2 + 0.75^2 + 0.79^2 + 1.06^2 \\
&= 6.25
\end{aligned}$$

which is slightly smaller than the previous SSE of 7.88 and indicates that we have our best clustering solution yet. We also have:

$$\frac{\text{BCV}}{\text{WCV}} = \frac{d(m_1, m_2)}{\text{SSE}} = \frac{2.93}{6.25} = 0.4688$$

which is larger than the previous 0.3338, indicating that we have again increased the between-cluster variation with respect to the within-cluster variation. To do so is the goal of every clustering algorithm, in order to produce well-defined clusters such that the similarity within the cluster is high while the similarity to records in other clusters is low.

# Example of k-means clustering at work- Applying k-means

- Step 4 (third pass): For each of the k clusters, find the cluster centroid and update the location of each cluster center to the new value of the centroid. Since no records have shifted cluster membership, the cluster centroids therefore also remain unchanged.
- Step 5: Repeat steps 3 and 4 until convergence or termination. Since the centroids remain unchanged, the algorithm terminates.

# Example of k-means clustering at work- Applying k-means

Note that the k-means algorithm cannot guarantee finding the the global minimum SSE, instead often settling at a local minimum. To improve the probability of achieving a global minimum, the analyst should rerun the algorithm using a variety of initial cluster centers. Moore[3] suggests (1) placing the first cluster center on a random data point, and (2) placing the subsequent cluster centers on points as far away from previous centers as possible.

# Example of k-means clustering at work- Applying k-means

One potential problem for applying the k-means algorithm is: Who decides how many clusters to search for? That is, who decides $k$? Unless the analyst has a priori knowledge of the number of underlying clusters, therefore, an "outer loop" should be added to the algorithm, which cycles through various promising values of $k$. Clustering solutions for each value of $k$ can therefore be compared, with the value of $k$ resulting in the smallest SSE being selected.

What if some attributes are more relevant than others to the problem formulation? Since cluster membership is determined by distance, we may apply the same axis-stretching methods for quantifying attribute relevance that we discussed in Lecture 5.

# Further Reading

- Chapter 4.8 of Data Mining - Practical Machine Learning Tools and Techniques, Second Edition - Ian H. Witten, Eibe Frank
- Chapter 8 of DISCOVERING KNOWLEDGE IN DATA - An Introduction to Data Mining - DANIEL T. LAROSE
- Chapter 8 of Introduction to Data Mining (Second Edition) - Pang-Ning Tan, Michael Steinbach, Anuj Karpatne, Vipin Kumar

# References

[1] P. H. A. Sneath and R. R. Sokal, *Numerical Taxonomy: The Principles and Practice of Numerical Classification*. W. H. Freeman and Co., 1973.

[2] J. B. MacQueen, "Some methods for classification and analysis of multivariate observations," in *Proc. of the fifth Berkeley Symposium on Mathematical Statistics and Probability*, L. M. L. Cam and J. Neyman, Eds., vol. 1. University of California Press, 1967, pp. 281–297.

[3] A. W. Moore, *K-means and Hierarchical Clustering*. School of Computer Science, Carnegie Mellon University, 2011. [Online]. Available: https://www.cs.cmu.edu/~./awm/tutorials/kmeans11.pdf

*Thank you.*

*Any Questions?*