Clustering

What is Clustering

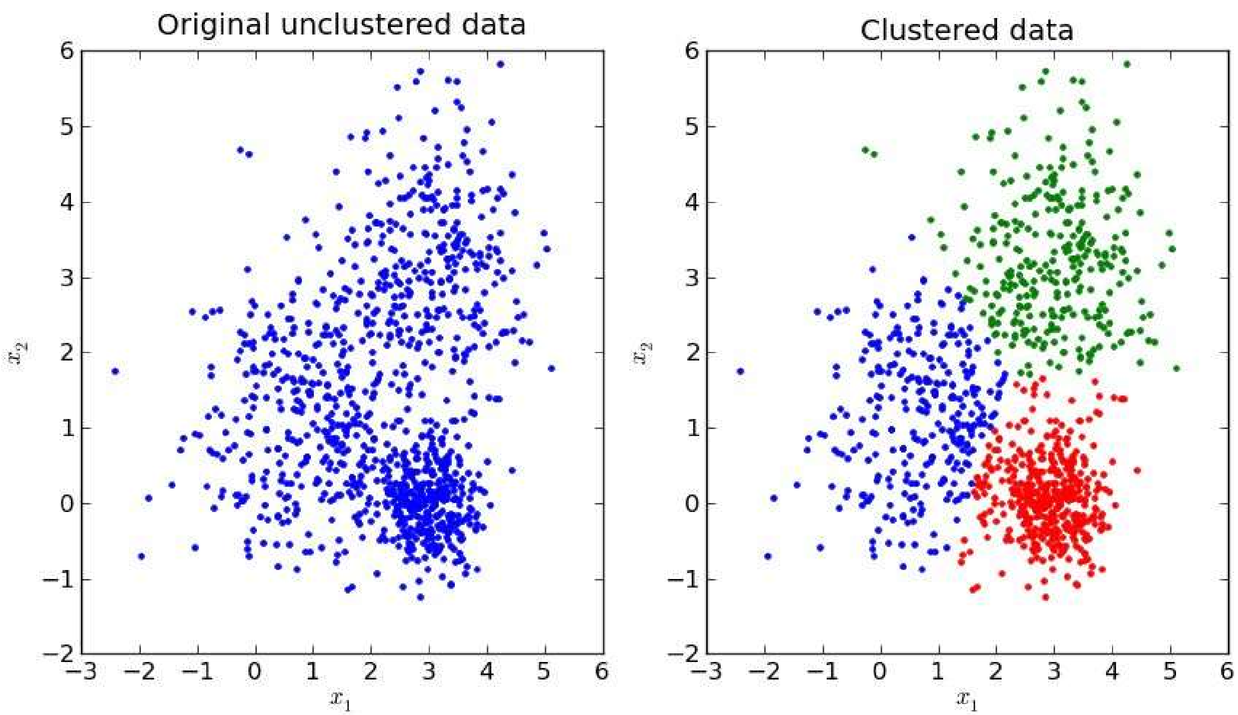
- Many applications require the partitioning of data points into intuitively similar groups.
- An informal and intuitive definition of clustering is as follows:
  - Given a set of data points, partition them into groups containing very similar data points.
- Applications include:
  - Data summarization: use cluster representitives as a summary.
  - Customer segmentation: group customers based on their attributes.
  - Social network analysis: detect communities.
  - Helper for other DM problems: e.g., clustering for outlier analysis.

Clustering Methods

- Representative-Based Algorithms
  - K-Means, K-Medians, K-Medoids.
  - Find a high quality set of representatives, then link data points to their closest representatives.
- Hierarchical Clustering Algorithms
  - Agglomerative: keep grouping similar objects/nodes.
  - Divisive: keep splitting dissimilar groups.
- Probabilistic Model-Based Algorithms:
  - Soft clustering.
- Density-Based Algorithms
  - DBSCAN, DENCLUE.

Representative-Based Algorithms

- Input: k, the number of clusters.



https://mubaris.com/posts/kmeans-clustering/

The sum of the distances of the different data points to their closest representatives needs to be minimized

The time complexity of each iteration is O(k · n · d) for a data set of size n and dimensionality d... If also consider iteration O(t * k * n * d) where t is the number of iterations

The space complexity of K-Means is generally considered to be O(n * d + k * d), where:
n is the number of data points,
d is the number of dimensions (features),
k is the number of clusters.
Notes:
The primary contributors to space complexity include the storage of data points (n×d), the cluster centers (k×d), and some additional variables for bookkeeping.

Representative-Based Algorithms

- Consider a data set D containing n data points denoted by $X_1 \ldots X_n$ in d-dimensional space.
- The goal is to determine k representatives $Y_1 \ldots Y_k$ that minimize the following objective function O:

$$O = \sum_{i=1}^{n} \left[\min_j Dist(\overline{X_i}, \overline{Y_j}) \right].$$

Representative-Based Algorithms

- Note that the optimal assignment of data points to representatives (Y1...Yn) are unknown a priori.
- The assignment of points to representatives requires first finding the representatives, and finding the representatives requires first knowing the groups. Chicken and egg !
- Such optimization problems are solved with the use of an iterative approach where candidate representatives and candidate assignments are used to improve each other.

Representative-Based Algorithms

- Generic *k*-representatives approach starts by initializing the *k* representatives *S* with the use of a straightforward heuristic (such as random sampling from the original data), and then refines the representatives and the clustering assignment, iteratively, as follows:
  - (Assign step) Assign each data point to its closest representative in *S* using distance function *Dist*(·, ·), and denote the corresponding clusters by $C1 \ldots Ck$.
  - (Optimize step) Determine the optimal representative *Yj* for each cluster *Cj* that minimizes its *local* objective function

$$\sum_{\overline{X_i} \in \mathcal{C}_j} \left[Dist(\overline{X_i}, \overline{Y_j}) \right]$$

Representative-Based Algorithms

Algorithm *GenericRepresentative*(Database: $\mathcal{D}$, Number of Representatives: k

begin

Initialize representative set S;

repeat

Create clusters $(\mathcal{C}_1 \dots \mathcal{C}_k)$ by assigning each point in $\mathcal{D}$ to closest representative in S using the distance function $Dist(\cdot,\cdot)$;

Recreate set S by determining one representative $\overline{Y_j}$ for each $\mathcal{C}_j$ that minimizes $\sum_{\overline{X_i} \in \mathcal{C}_j} Dist(\overline{X_i}, \overline{Y_j})$;

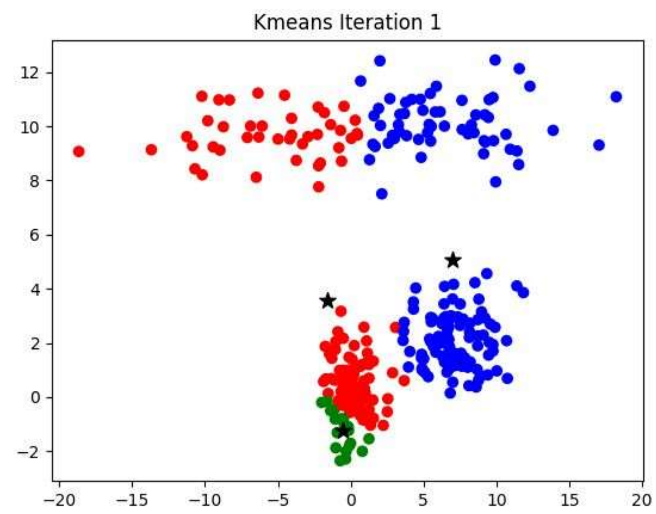
until convergence;

return $(\mathcal{C}_1 \dots \mathcal{C}_k)$;

end

Representative-Based Algorithms

- The idea is to improve the objective function over multiple iterations.
- Typically, the increase is significant in early iterations, but it slows down in later iterations. The primary computational bottleneck of the
- approach is the assignment step where the distances need to be computed between all point representative pairs.



The k-Means Algorithm

- A variant of the Representative-Based Algorithms, where the objective function is the sum of the squares of the Euclidean distances (L_2-norm) of data points to their closest representatives (SSE).

$$Dist(\overline{X_i}, \overline{Y_j}) = ||\overline{X_i} - \overline{Y_j}||_2^2.$$

The K-Means Algorithm

K=3, D={A1=(2,10), A2=(2,5), A3=(8,4), A4=(5,8), A5=(7,5), A6=(6,4), A7=(1,2), A8=(4,9)}. Choose your seeds.

	A1	A2	A3	A4	A5	A6	A7	A8
A1	0	$\sqrt{25}$	$\sqrt{36}$	$\sqrt{13}$	$\sqrt{50}$	$\sqrt{52}$	$\sqrt{65}$	$\sqrt{5}$
A2		0	$\sqrt{37}$	$\sqrt{18}$	$\sqrt{25}$	$\sqrt{17}$	$\sqrt{10}$	$\sqrt{20}$
A3			0	$\sqrt{25}$	$\sqrt{2}$	$\sqrt{2}$	$\sqrt{53}$	$\sqrt{41}$
A4				0	$\sqrt{13}$	$\sqrt{17}$	$\sqrt{52}$	$\sqrt{2}$
A5					0	$\sqrt{2}$	$\sqrt{45}$	$\sqrt{25}$
A6						0	$\sqrt{29}$	$\sqrt{29}$
A7							0	$\sqrt{58}$
A8								0

https://webdocs.cs.ualberta.ca/~zaiane/courses/cmput695/F07/exercises/Exercises695Clus-solution.pdf

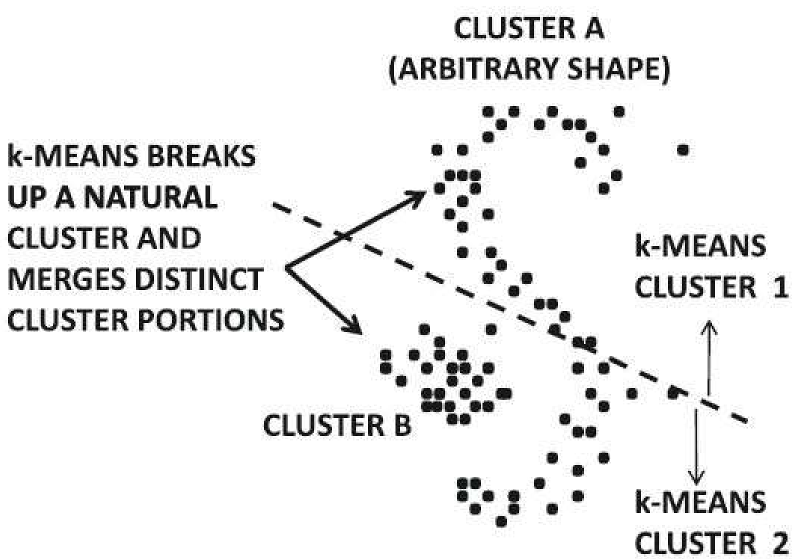
The K-Means Algorithm

- What is the run time complexity of K-Means ?
- Does k-means select the representatives among the original points ?
- Can you think of disadvantages ?

Mean is more sensitive to outlier
Distance needs to be computed between each data point and representative point
Only suitable for spherical shaped cluster

The K-Means Algorithm

- K-Means does not work well for clusters of arbitrary shape.
- It is biased towards finding spherical clusters.



The K-Medians Algorithm

- *Dist*(*Xi, Yj*) is the Manhattan distance:

$$Dist(\overline{X_i}, \overline{Y_j}) = ||X_i - Y_j||_1.$$

- Accordingly representative of a group is the median point. This is because the point that has the minimum sum of *L*1-distances to a set of points distributed on a line is the median of that set.
- The median is chosen independently at each dimension.
- As the median is less sensitive to outliers, K-Medians is more robust than K-Means.

The K-Medoids Algorithm

Algorithm $GenericMedoids$(Database: $\mathcal{D}$, Number of Representatives: k)
begin
  Initialize representative set S by selecting from $\mathcal{D}$;
  repeat
    Create clusters $(\mathcal{C}_1 \dots \mathcal{C}_k)$ by assigning
      each point in $\mathcal{D}$ to closest representative in S
      using the distance function $Dist(\cdot, \cdot)$;
    Determine a pair $\overline{X_i} \in \mathcal{D}$ and $\overline{Y_j} \in S$ such that
      replacing $\overline{Y_j} \in S$ with $\overline{X_i}$ leads to the
      greatest possible improvement in objective function;
    Perform the exchange between $\overline{X_i}$ and $\overline{Y_j}$ only
      if improvement is positive;
  until no improvement in current iteration;
  return $(\mathcal{C}_1 \dots \mathcal{C}_k)$;
end

The K-Medoids Algorithm

- Cluster representatives are always chosen from the dataset D.
- Requires only a distance function. No mean/median is required. So it can be used for complex types.
- Uses hill-climbing strategy, in which the representative set *S* is initialized to a set of points from the original database *D*.
- Subsequently, this set *S* is iteratively improved by exchanging a single point from set *S* with a data point selected from the database *D*.
- This iterative exchange can be viewed as a hill-climbing strategy, because each exchange can be viewed as a hill-climbing step.
- At every iteration, try multiple exchanges, and choose the best.

Practical Issues

- How to initialize the cluster representatives ?
  - Random.
  - Sample, then use another clustering method.
  - Sample k times, and use the centroid of each.
- K-Means can get stuck with a singleton cluster if an outlier is used in initialization. What to do ?
- It is difficult to determine a good value k. One may chose a bigger value than the analyst's k, then perform cluster merging as a post processing.

Credits and Readings

- These slides, except when explicitly stated, use material from:
  - Charu C.Aggarwal. Data Mining The Textbook, Springer.