COMPSCI 762 Tutorial 9

Tutorial on Unsupervised Learning

Luke Chang

May 2021

The University of Auckland

Unsupervised Learning - Cluster Analysis

- Cluster analysis is an unsupervised learning task.
- The task of clustering is to partition a set of objects such that objects in the same group are more similar to each other than those in other groups.
- Evaluation metrics:
-
    - *Sum of Squared Error* (SSE)
    - *Sum of Squared Between*(SSB)
- Algorithms we will cover:
    - K-means
    - Hierarchical clustering
    - DBSCAN: Density-based clustering

K-means

- Partition the data into clusters
- Hyperparameter: the number of clusters, K
- Each cluster is associated with a centroid
- Each point is assigned to the cluster with the closest centroid
- Iterative method: Update centroids in each iteration, converge until the centroids don't change

Limitations:

- When clusters have:
  - Different sizes
  - Different densities
  - Non-globular shapes (hypersphere)
- When data contains outliers

K-means Pseudocode

Select K points as the initial centroids
repeat:
  For each point:
    Assign the point to the closest centroid
  For $i \in \{1, \ldots, K\}$:
    Update the i centroid
until The centroids (or points) don't change

Evaluation Metric: Sum of Squared Error (SSE)

- No true labels are available in the unsupervised learning
- Use sum of the squared errors to evaluate the performance

$$\mathrm{SSE} = \sum_{i=1}^{K} \sum_{x \in C_i} \mathrm{dist}^2(m_i, x)$$

- SSE highly depends on **K** and the **initial centroids**
- Depends on the initial K-centroids, the K-means clusters with same the K value may not have the same SSE

Solve the Initial Centroids Problem

Problem:

- Stuck in the local minimal

Example

Demonstration of k-means assumptions from *sklearn*

Solution:

- Multiple runs
- Use hierarchical clustering to determine the initial centroids
- Define more than K initial centroids and then select among these initial centroids
- Postprocessing:
  - Remove empty clusters and small cluster
  - Split "loose" clusters – High SSE
  - Merge clusters when they are "close" – Low SSE
  - Apply these steps multiple times and use the pruned centroids as new initial centroids
- Improved k-means algorithms: Bisecting K-means, Mini Batch K-Means

Hierarchical Clustering

- Produces a set of nested clusters organized as a hierarchical tree
- Can be visualized as a dendrogram
- No assumption on the number of clusters
- Two main types:
  - Agglomerative: Start from each data point, merge the closest pair of clusters
  - Divisive: Start with one big cluster which contains all data, split at each step

Agglomerative Clustering Algorithm Pseudocode

Compute the proximity matrix
Let each point be a cluster
repeat:
    Merge the two closest clusters
    Update the proximity matrix
until only a single cluster remains

Proximity Matrix

The linkage criteria determines the metric used for the merge strategy:

- **Min – Single-linkage:** uses the minimum of the distances between all observations of the two sets
  - Can handle non-elliptical shapes
  - Sensitive to noise and outliers
- **Max – Complete-linkage:** uses the maximum distances between all observations of the two sets
  - Less susceptible to noise and outliers
  - Tends to break large clusters
  - Biased towards globular clusters
- **Group average – Average-linkage:** The average of the distances between all observations of pairs of clusters
  - Less susceptible to noise and outliers
  - Biased towards globular clusters

$$\frac{1}{|C_i| \cdot |C_j|} \sum_{i \in C_i} \sum_{j \in C_j} d(i,j)$$

Proximity Matrix (continue)

- **Distance Between Centroids – Centroid-linkage:** Distance between the centroids of two clusters
- **Ward's minimum variance method – Ward's linkage:** Minimizes the sum of squared differences of the clusters being merged

$$\mathrm{SSE}(C_i, C_j) - [\mathrm{SSE}(C_i) + \mathrm{SSE}(C_j)]$$

  where $\mathrm{SSE}(C_i, C_j)$ is the SSE of the union of the cluster i and the cluster j.

  - Less susceptible to noise and outliers
  - Biased towards globular clusters
  - Hierarchical analogue of K-means; Can be used to initialize K-means

Example

Hierarchical clustering from *sklearn*

DBSCAN: Density-based Clustering

- DBSCAN: Density-Based Spatial Clustering of Applications with Noise
- **Density:** The number of points within a specified radius (ε)
- **MinPts (min_samples)):** A point is a **core point** if it has at least **MinPts** within ε.
- A **border point** is not a **core point**, but is in the neighborhood of a core point.
- A **noise point** is any point that is not a core point or a border point.

Limitations:

- Clusters with varied densities
- High dimensional data

DBSCAN Pseudocode

```
DBSCAN(DB, distFunc, eps, minPts) {
    C := 0                                                  /* Cluster counter */
    for each point P in database DB {
        if label(P) ≠ undefined then continue               /* Previously processed in inner loop */
        Neighbors N := RangeQuery(DB, distFunc, P, eps)     /* Find neighbors */
        if |N| < minPts then {                              /* Density check */
            label(P) := Noise                               /* Label as Noise */
            continue
        }
        C := C + 1                                          /* next cluster label */
        label(P) := C                                       /* Label initial point */
        SeedSet S := N \ {P}                                /* Neighbors to expand */
        for each point Q in S {                             /* Process every seed point Q */
            if label(Q) = Noise then label(Q) := C          /* Change Noise to border point */
            if label(Q) ≠ undefined then continue           /* Previously processed (e.g., border point) */
            label(Q) := C                                   /* Label neighbor */
            Neighbors N := RangeQuery(DB, distFunc, Q, eps) /* Find neighbors */
            if |N| ≥ minPts then {                          /* Density check (if Q is a core point) */
                S := S ∪ N                                  /* Add new neighbors to seed set */
            }
        }
    }
}
```

Cluster Cohesion and Separation

Cluster Cohesion

- Measures how closely related are data points in a cluster
- *Within Cluster Sum of Squares* (WCSS) = *Sum of Squared Error* (SSE)

$$\text{SSE} = \sum_{i=1}^{K} \sum_{x \in C_i} ||x - m_i||^2$$

Cluster Separation

- Measure how distinct or well-separated a cluster is from other clusters
- *Between cluster Sum of Squares* (BSS) a.k.a. *Sum of Squared Between* (SSB)

$$\text{SSB} = \sum_{i=1}^{K} |C_i|(m - m_i)^2$$

where $|C_i|$ is the size of cluster i, m is the grand mean, m_i is the mean for the cluster i.

Cluster Cohesion and Separation

- *Sum of Squares Total* (SST): The sum of squares between the n data points and the grand mean

$$\text{SST} = \text{SSB} + \text{SSE}$$

- SST is a constant based on the observed data points.
- The same terminologies are used in the one-way analysis of variance (**ANOVA**) test.

Sums of Squares (SS) Example

Example

Divide 1D data points $\{1,2,3,6,7\}$ into 2 clusters: $\{1,2,3\}$ and $\{6,7\}$.

- $n = 5$, $|C_1| = 3$, and $|C_2| = 2$
- $m = (1+2+3+6+7)/5 = 3.8$
- $m_1 = (1+2+3)/3 = 2$
- $m_2 = (6+7)/2 = 6.5$

$$
\begin{aligned}
\text{SSB} &= 3(3.8-2)^2 + 2(3.8-6.5)^2 = 24.3 \\
\text{SSE} &= (1-2)^2 + (2-2)^2 + (3-2)^2 + (6-6.5)^2 + (7-6.5)^2 = 2.5 \\
\text{SST} &= \text{SSB} + \text{SSE} = 24.3 + 2.5 = 26.8
\end{aligned}
$$

The alternative way to compute SST:

$$
\text{SST} = (1-3.8)^2 + (2-3.8)^2 + (3-3.8)^2 + (6-3.8)^2 + (7-3.8)^2 = 26.8
$$

Silhouette Coefficient

Silhouette coefficient combines ideas of both cohesion and separation, but for individual points.

For each data point, i:
a: The mean distance between a sample and all other points in the same class

$$a(i) = \frac{1}{|C_i| - 1} \sum_{j \in C_i, y \neq i} d(i,j)$$

b: The mean distance between a sample and all other points in the **next nearest cluster**

$$b(i) = \min_{k \neq i} \frac{1}{|C_k|} \sum_{j \in C_k} d(i,j)$$

Silhouette Coefficient for the Entire Dataset

The Silhouette Coefficient for a single sample $s(i)$ is then given as:

$$s(i) = \begin{cases} 1 - a(i)/b(i) & \text{, if } a(i) < b(i) \\ b(i)/a(i) - 1 & \text{, if } a(i) \leq b(i) \end{cases}$$

- The Silhouette Coefficient for a set of samples is given as the mean of the Silhouette Coefficient for each sample.
- The range is in $[0, 1]$, the **larger** the better.

Example

An example from *sklearn*

Overview of Clustering Methods From *sklearn*

Example

The *sklearn* Documentation for Clustering