Lecture Notes for Chapter 7

Introduction to Data Mining, 2nd Edition

by

Tan, Steinbach, Karpatne, Kumar
DBSCAN

DBSCAN is a density-based algorithm.

- Density = number of points within a specified radius (Eps)

- A point is a core point if it has at least a specified number of points (MinPts) within Eps
  - These are points that are at the interior of a cluster
  - Counts the point itself

- 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
DBSCAN: Core, Border, and Noise Points

MinPts = 7

noise point

border point

core point

Eps

C

A

B

Eps
DBSCAN Algorithm

- Eliminate noise points
- Perform clustering on the remaining points

```plaintext
current_cluster_label ← 1
for all core points do
    if the core point has no cluster label then
        current_cluster_label ← current_cluster_label + 1
        Label the current core point with cluster label current_cluster_label
    end if
    for all points in the Eps-neighborhood, except i\textsuperscript{th} the point itself do
        if the point does not have a cluster label then
            Label the point with cluster label current_cluster_label
        end if
    end for
end for
```
DBSCAN: Core, Border and Noise Points

Original Points

Point types: core, border and noise

Eps = 10, MinPts = 4
When DBSCAN Works Well

- Resistant to Noise
- Can handle clusters of different shapes and sizes
When DBSCAN Does NOT Work Well

Original Points

- Varying densities
- High-dimensional data

(MinPts=4, Eps=9.75).

(MinPts=4, Eps=9.92)
DBSCAN: Determining EPS and MinPts

- Idea is that for points in a cluster, their \( k^{th} \) nearest neighbors are at roughly the same distance.
- Noise points have the \( k^{th} \) nearest neighbor at farther distance.
- So, plot sorted distance of every point to its \( k^{th} \) nearest neighbor.

![Plot showing sorted distance of points to their \( k^{th} \) nearest neighbor](image)
CLUSTER VALIDITY
Cluster Validity

- For supervised classification we have a variety of measures to evaluate how good our model is
  - Accuracy, precision, recall

- For cluster analysis, the analogous question is how to evaluate the “goodness” of the resulting clusters?

- But “clusters are in the eye of the beholder”!

- Then why do we want to evaluate them?
  - To avoid finding patterns in noise
  - To compare clustering algorithms
  - To compare two sets of clusters
  - To compare two clusters
Clusters found in Random Data

Random Points

K-means

DBSCAN

Complete Link
Different Aspects of Cluster Validation

1. Determining the clustering tendency of a set of data, i.e., distinguishing whether non-random structure actually exists in the data.

2. Comparing the results of a cluster analysis to externally known results, e.g., to externally given class labels.

3. Evaluating how well the results of a cluster analysis fit the data without reference to external information.
   - Use only the data

4. Comparing the results of two different sets of cluster analyses to determine which is better.

5. Determining the ‘correct’ number of clusters.

For 2, 3, and 4, we can further distinguish whether we want to evaluate the entire clustering or just individual clusters.
Measures of Cluster Validity

- Numerical measures that are applied to judge various aspects of cluster validity, are classified into the following three types.
  - **External Index**: Used to measure the extent to which cluster labels match externally supplied class labels.
    - Entropy
  - **Internal Index**: Used to measure the goodness of a clustering structure *without* respect to external information.
    - Sum of Squared Error (SSE)
  - **Relative Index**: Used to compare two different clusterings or clusters.
    - Often an external or internal index is used for this function, e.g., SSE or entropy

- Sometimes these are referred to as *criteria* instead of *indices*
  - However, sometimes criterion is the general strategy and index is the numerical measure that implements the criterion.
Measuring Cluster Validity Via Correlation

- Two matrices
  - Proximity Matrix
  - Ideal Similarity Matrix
    - One row and one column for each data point
    - An entry is 1 if the associated pair of points belong to the same cluster
    - An entry is 0 if the associated pair of points belongs to different clusters

- Compute the correlation between the two matrices
  - Since the matrices are symmetric, only the correlation between \( n(n-1)/2 \) entries needs to be calculated.

- High correlation indicates that points that belong to the same cluster are close to each other.

- Not a good measure for some density or contiguity based clusters.
Measuring Cluster Validity Via Correlation

- Correlation of ideal similarity and proximity matrices for the K-means clusterings of the following two data sets.

\[
\text{Corr} = -0.9235
\]

\[
\text{Corr} = -0.5810
\]
Using Similarity Matrix for Cluster Validation

- Order the similarity matrix with respect to cluster labels and inspect visually.
Using Similarity Matrix for Cluster Validation

- Clusters in random data are not so crisp

DBSCAN
Using Similarity Matrix for Cluster Validation

- Clusters in random data are not so crisp

K-means
Using Similarity Matrix for Cluster Validation

- Clusters in random data are not so crisp

Complete Link
Using Similarity Matrix for Cluster Validation

DBSCAN
Clusters in more complicated figures aren’t well separated

Internal Index: Used to measure the goodness of a clustering structure without respect to external information
  – SSE

SSE is good for comparing two clusterings or two clusters (average SSE).

Can also be used to estimate the number of clusters
Internal Measures: SSE

- SSE curve for a more complicated data set

SSE of clusters found using K-means
Framework for Cluster Validity

- Need a framework to interpret any measure.
  - For example, if our measure of evaluation has the value, 10, is that good, fair, or poor?

- Statistics provide a framework for cluster validity
  - The more “atypical” a clustering result is, the more likely it represents valid structure in the data
  - Can compare the values of an index that result from random data or clusterings to those of a clustering result.
    - If the value of the index is unlikely, then the cluster results are valid
  - These approaches are more complicated and harder to understand.

- For comparing the results of two different sets of cluster analyses, a framework is less necessary.
  - However, there is the question of whether the difference between two index values is significant
### Statistical Framework for SSE

#### Example

- Compare SSE of 0.005 against three clusters in random data
- Histogram shows SSE of three clusters in 500 sets of random data points of size 100 distributed over the range 0.2 – 0.8 for x and y values
Statistical Framework for Correlation

- Correlation of ideal similarity and proximity matrices for the K-means clusterings of the following two data sets.

\[ \text{Corr} = -0.9235 \]

\[ \text{Corr} = -0.5810 \]
Internal Measures: Cohesion and Separation

- **Cluster Cohesion**: Measures how closely related are objects in a cluster
  - Example: SSE

- **Cluster Separation**: Measure how distinct or well-separated a cluster is from other clusters

- Example: Squared Error
  - Cohesion is measured by the within cluster sum of squares (SSE)
    \[
    SSE = WSS = \sum_i \sum_{x \in C_i} (x - m_i)^2
    \]
  - Separation is measured by the between cluster sum of squares
    \[
    BSS = \sum_i |C_i| (m - m_i)^2
    \]
  - Where \(|C_i|\) is the size of cluster \(i\)
Internal Measures: Cohesion and Separation

- **Example: SSE**
  - \( \text{BSS} + \text{WSS} = \text{constant} \)

\[
\begin{align*}
\text{K=1 cluster:} & \quad SSE = \text{WSS} = (1 - 3)^2 + (2 - 3)^2 + (4 - 3)^2 + (5 - 3)^2 = 10 \\
& \quad BSS = 4 \times (3 - 3)^2 = 0 \\
& \quad Total = 10 + 0 = 10
\end{align*}
\]

\[
\begin{align*}
\text{K=2 clusters:} & \quad SSE = \text{WSS} = (1 - 1.5)^2 + (2 - 1.5)^2 + (4 - 4.5)^2 + (5 - 4.5)^2 = 1 \\
& \quad BSS = 2 \times (3 - 1.5)^2 + 2 \times (4.5 - 3)^2 = 9 \\
& \quad Total = 1 + 9 = 10
\end{align*}
\]
A proximity graph based approach can also be used for cohesion and separation.

- Cluster cohesion is the sum of the weight of all links within a cluster.
- Cluster separation is the sum of the weights between nodes in the cluster and nodes outside the cluster.
Silhouette coefficient combines ideas of both cohesion and separation, but for individual points, as well as clusters and clusterings.

For an individual point, $i$
- Calculate $a =$ average distance of $i$ to the points in its cluster
- Calculate $b =$ min (average distance of $i$ to points in another cluster)
- The silhouette coefficient for a point is then given by

$$s = \frac{(b - a)}{\max(a,b)}$$

- Typically between 0 and 1.
- The closer to 1 the better.

Can calculate the average silhouette coefficient for a cluster or a clustering.
Table 5.9. K-means Clustering Results for LA Document Data Set

<table>
<thead>
<tr>
<th>Cluster</th>
<th>Entertainment</th>
<th>Financial</th>
<th>Foreign</th>
<th>Metro</th>
<th>National</th>
<th>Sports</th>
<th>Entropy</th>
<th>Purity</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>3</td>
<td>5</td>
<td>40</td>
<td>506</td>
<td>96</td>
<td>27</td>
<td>1.2270</td>
<td>0.7474</td>
</tr>
<tr>
<td>2</td>
<td>4</td>
<td>7</td>
<td>280</td>
<td>29</td>
<td>39</td>
<td>2</td>
<td>1.1472</td>
<td>0.7756</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>7</td>
<td>4</td>
<td>671</td>
<td>0.1813</td>
<td>0.9796</td>
</tr>
<tr>
<td>4</td>
<td>10</td>
<td>162</td>
<td>3</td>
<td>119</td>
<td>73</td>
<td>2</td>
<td>1.7487</td>
<td>0.4390</td>
</tr>
<tr>
<td>5</td>
<td>331</td>
<td>22</td>
<td>5</td>
<td>70</td>
<td>13</td>
<td>23</td>
<td>1.3976</td>
<td>0.7134</td>
</tr>
<tr>
<td>6</td>
<td>5</td>
<td>358</td>
<td>12</td>
<td>212</td>
<td>48</td>
<td>13</td>
<td>1.5523</td>
<td>0.5525</td>
</tr>
<tr>
<td>Total</td>
<td>354</td>
<td>555</td>
<td>341</td>
<td>943</td>
<td>273</td>
<td>738</td>
<td>1.1450</td>
<td>0.7203</td>
</tr>
</tbody>
</table>

**Entropy** For each cluster, the class distribution of the data is calculated first, i.e., for cluster $j$ we compute $p_{ij}$, the ‘probability’ that a member of cluster $j$ belongs to class $i$ as follows: $p_{ij} = m_{ij}/m_j$, where $m_j$ is the number of values in cluster $j$ and $m_{ij}$ is the number of values of class $i$ in cluster $j$. Then using this class distribution, the entropy of each cluster $j$ is calculated using the standard formula $e_j = \sum_{i=1}^{L} p_{ij} \log_2 p_{ij}$, where the $L$ is the number of classes. The total entropy for a set of clusters is calculated as the sum of the entropies of each cluster weighted by the size of each cluster, i.e., $e = \sum_{j=1}^{K} \frac{m_j}{m} e_j$, where $m_j$ is the size of cluster $j$, $K$ is the number of clusters, and $m$ is the total number of data points.

**Purity** Using the terminology derived for entropy, the purity of cluster $j$, is given by $purity_j = \max p_{ij}$ and the overall purity of a clustering by $purity = \sum_{j=1}^{K} \frac{m_j}{m} purity_j$. 
Final Comment on Cluster Validity

“The validation of clustering structures is the most difficult and frustrating part of cluster analysis.
Without a strong effort in this direction, cluster analysis will remain a black art accessible only to those true believers who have experience and great courage.”

*Algorithm for Clustering Data*, Jain and Dubes