Lecture 24 - Clustering Validity Measures; Non-parametric Generative Classification

Cluster Validity Metrics

How would you evaluate clustering results? - **Cluster Validity Indices**

- Cluster validity indices are used for a number of different goals. For example, cluster validity metrics can be used to compare clustering results, try to determine the *correct* number of clusters, try to select the *correct* parameter settings, try to evaluate the approppriateness of the clustering result based on the data only (and not using another result or "ground truth" data).

In general, there are three types of **index criteria** to perform cluster validity:

1. **Internal criteria.** We evaluate the results of a clustering algorithm in terms of quantities that involve the vectors of the data set themselves.
2. **External criteria.** We evaluate the results of a clustering algorithm based on a pre-specified structure, which is imposed on a data set and reflects our intuition about the clustering structure of the data set.
3. **Relative criteria.** We evaluate the results of a clustering structure by comparing it to other clustering schemes, resulting by the same algorithm but with different parameter values. In practice, relative criteria are a combination on internal and external criteria.

Internal Criteria

As the goal of clustering is to make objects within the same cluster similar and objects in different clusters distinct, internal cluster validity measures are defined by combining compactness and separability.

The optimal clustering scheme under the internal criteria index includes:

- Compactness (or intra-distance or within-cluster scatter): The members of each cluster should be as close to each other as possible. A common measure of compactness is the variance, which should be minimized.
- Separation (or inter-distance or between-cluster scatter): This indicates how distinct two clusters are. It computes the distance between two different clusters. There are three common approaches measuring the distance between two different clusters:
  - Single linkage: It measures the distance between the closest members of the clusters.
  - Complete linkage: It measures the distance between the most distant members.
  - Comparison of centroids: It measures the distance between the centers of the clusters.

Example: Silhouette Index

The Silhoute Index is an internal cluster validity index that is used to judge the quality of any clustering solution.

Given a set of data points $X = \{x_1, \ldots, x_N\}$ and a partition of X (i.e. clustering result). Let's define the following:

- a_i is the average distance of the point x_i to all the other points of the cluster in which x_i is assigned to
- b_i is the average distance of the point x_i to all the other points of in the other clusters.

For every data point $x_i \in X$, the Silhouette Index is defined as:

$$s = \frac{1}{N} \sum_{i=1}^{N} \frac{b_i - a_i}{\max(a_i, b_i)}$$

- Silhouette index is the average silhouette of all data points and it reflects the compactness and separation of clusters.
- The value of silhouette index varies from -1 and 1 and higher indicates better clustering results.

There are many other internal cluster validity indices!

External Criteria

External cluster validity indices are used to measure how well a clustering result matches a set of *give* labels. External cluster validity indices can be used to:

- compare the clustering results with the *ground truth* (true labels),
- compare clustering results between different clustering algorithms to measure how different they are and how stable a particular clustering is on a data set across parameter settings and/or algorithms.

Example: Rand Index

The Rand Index is an external cluster validity index that is used to compare clustering results obtained from different parameter settings or algorithms.

Given a set of data points X and two partitions (i.e. clustering results) of X to compare. One partition $C = \{C_1, \ldots, C_k\}$, that partitions X into k clusters, and another partition $D = \{D_1, \ldots, D_s\}$, that partitions X into s clusters. Let's define the following:

- a is the number of pairs of elements in X that are in the same subset in C and in the same subset in D.
- b is the number of pairs of elements in X that are in different subset in C and in different subset in D.
- c is the number of pairs of elements in X that are in the same subset in C and in different subset in D.

- d is the number of pairs of elements in X that are in different subset in C and in the same subset in D.

The Rand Index is defined as:

$$r = \frac{a+b}{a+b+c+d}$$

- Intuitively, $a+b$ can be considered as the number of *agreements* between C and D, and $c+d$ as the number of *disagreements* between C and D.
- The value of rand index varies from 0 and 1 and higher indicates higher consistency between partitions C and D.

There are many other external cluster validity indices!

Non-parametric Generative Classification

We have introduced the probabilistic generative classifier, and, as we discussed, the probabilistic generative classifier requires us to assume a parametric form for each class (e.g., each class is represented by a multivarite Gaussian distribution, etc.). Because of this, the probabilistic generative classifier is a *parametric* approach.

- Parametric approaches have the drawback that the functional parametric form needs to be decided/assumed in advance and, if chosen poorly, might a poor model of the distribution that generates the data resulting in poor performance.

Non-parametric methods are those that do not assume a particular generating distribution for the data. The **K-nearest nerighbors (K-NN)** algorithm is one example of a non-parametric classifier.

K-Nearest Neighbors Classifier

Nearest neighbors methods compare a test point to the k nearest training data points and then estimate an output value based on the desired/true output values of the k nearest training points.

- Essentially, there is no "training" other than storing the training data points and their desired outputs
- In test, you need to:
    1. Determine which k neighbors in the training data are closest to the test point; and,
    2. Determine the output value for the test point.

In order to find the k *nearest-neighbors* in the training data, you need to define a **similarity measure** or a **dissimilarity measure**. The most common dissimilarity measure is Euclidean distrance:

- Euclidean distance: $d_E(\mathbf{x}_1, \mathbf{x}_2) = \sqrt{(\mathbf{x}_1 - \mathbf{x}_2)^T(\mathbf{x}_1 - \mathbf{x}_2)}$

- City-block distance: $d_{CB}(\mathbf{x}_1, \mathbf{x}_2) = \sum_{i=1}^{n} |\mathbf{x}_{1i} - \mathbf{x}_{2i}|$
- Mahalanobis distance: $d_M(\mathbf{x}_1, \mathbf{x}_2) = \sqrt{(\mathbf{x}_1 - \mathbf{x}_2)^T \Sigma^{-1} (\mathbf{x}_1 - \mathbf{x}_2)}$
- Cosine angle similarity: $\cos(\theta) = \frac{\mathbf{x}_1^T \mathbf{x}_2}{\|\mathbf{x}_1\|_2^2 \|\mathbf{x}_2\|_2^2}$
- and many more.

If you are doing classification, once you find the k nearest neighbors to your test point in the training data, then you can determine the class label of your test point using (most commonly) **majority vote**.

- If there are ties, they can be broken randomly or using schemes like applying the label to the closest data point in the neighborhood.

Discussions

1. What happens when there are imbalanced classes?
2. Is k-NN sensitive to data scaling?