Evaluation Metrics for Clustering Models

Dr. Or Haim Anidjar

Based on https://towardsdatascience.com/evaluation-metrics-for-clustering-models-5dde821dd6cd

Introduction

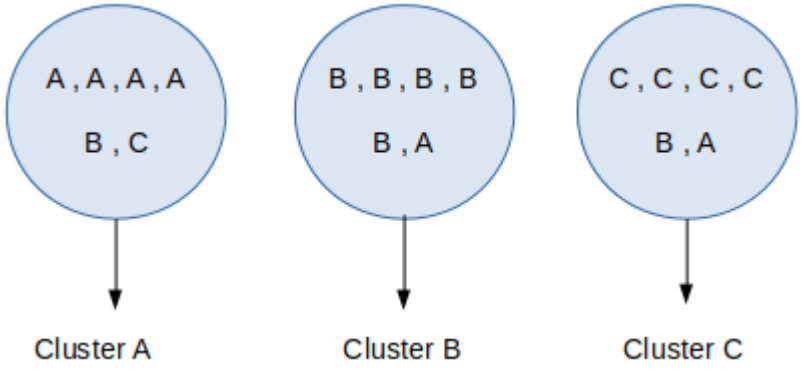
- Clustering is a fundamental task in machine learning. Clustering algorithms group data points in clusters in a way that similar data points are grouped together.
- The ultimate goal of a clustering algorithm is to achieve high intra-cluster similarity and low inter-cluster similarity.
- In other words, we want data points in the same cluster to be as close to each other as possible. The distance between different clusters needs to be as high as possible.

Introduction

- There are different metrics used to evaluate the performance of a clustering model or clustering quality. In this article, we will cover the following metrics:
  - Purity
  - Normalized mutual information (NMI)
  - Rand index

Purity

- Purity is quite simple to calculate. We assign a label to each cluster based on the most frequent class in it. Then the purity becomes the number of correctly matched class and cluster labels divided by the number of total data points.
- Consider a case where our clustering model groups the data points into 3 clusters as seen below:



Purity

- Each cluster is assigned with the most frequent class label. We sum the number of correct class labels in each cluster and divide it by the total number of data points.

$$purity = \frac{(clusterA + clusterB + clusterC)}{total} = \frac{(4+5+4)}{18} = 0.722$$

- In general, purity increases as the number of clusters increases. For instance, if we have a model that groups each observation in a separate cluster, the purity becomes one.

- For this very reason, purity cannot be used as a trade off between the number of clusters and clustering quality.

Normalized mutual information (NMI)

- NMI is related to the information theory. We need first to understand what ***Entropy*** is.
- ***Entropy*** is a measure that quantifies uncertainty.

$$H(p) = -\sum_{i} p_i \log_2 (p_i)$$

- p_i is the probability of the label i (that is, p_i).
- Let's calculate the entropy of the class labels in the previous examples.

Normalized mutual information (NMI)

- We can calculate the probability of a class label by dividing the number of data points belong to that class to the total number of data points. For instance, probability of class A is 6 / 18.

- The entropy in our case is calculated as below. If you run the calculation, you will see that the result is 1.089.

$$entropy = -\left(\left(\frac{6}{18}\right) \cdot \log\left(\frac{6}{18}\right)\right) - \left(\left(\frac{7}{18}\right) \cdot \log\left(\frac{7}{18}\right)\right) - \left(\left(\frac{5}{18}\right) \cdot \log\left(\frac{5}{18}\right)\right)$$

Normalized mutual information (NMI)

- The labels are approximately equally distributed among classes so we have relatively high entropy.

- Entropy decreases as the uncertainty decreases. Consider a case where we have two classes (9 data points in class A and 1 data point in class B). In that case, we are more certain than the previous case if we are to predict the class of a randomly selected data point.

- The entropy is this case is calculated as below which results in 0.325.

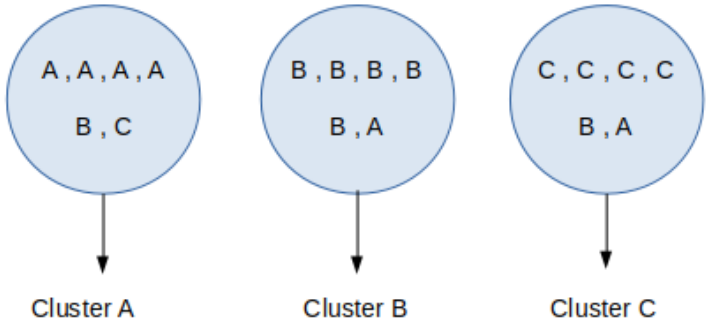
$$entropy = -\left(\left(\frac{9}{10}\right) \cdot \log\left(\frac{9}{10}\right)\right) - \left(\left(\frac{1}{10}\right) \cdot \log\left(\frac{1}{10}\right)\right)$$

Normalized mutual information (NMI)

- We now have a basic understanding of entropy. Normalized mutual information (NMI) gives us the reduction in entropy of class labels when we are given the cluster labels.
- In a sense, NMI tells us how much the uncertainty about class labels decreases when we know the cluster labels.
- It is similar to the information gain in decision trees. In the process of building a decision tree, each split adds information gain to the model. In fact, the split that results in the highest information gain is selected.

Normalized mutual information (NMI)

- Recall the case with three clusters. Since there are approximately equal number of data points in each class, we are uncertain about the class of a randomly picked data point.



- However, if we know a data point belongs to cluster A, it is highly likely that the point belongs to class A. Thus, our uncertainty has decreased. NMI measures this reduction in uncertainty. Thus, it is measure of clustering quality.

- One advantage of NMI is that we can use it to compare different clustering models that have different number of clusters because NMI is normalized.

Rand index

- Rand index is a measure of similarity between two clusters. We can use it to compare actual class labels and predicted cluster labels to evaluate the performance of a clustering algorithm.

- The first step is to create a set of unordered pairs of data points. For instance, if we have 6 data points, the set contains 15 unordered pairs which are also called binomial coefficients.

- The number of binomial coefficients can easily be calculated using the scipy package for Python.

Rand index

- Consider we have the following data points.

	name	actual	predicted
0	a	1	1
1	b	1	1
2	c	1	2
3	d	2	2
4	e	2	3
5	f	2	3

- The unordered pairs of data points are:
  - **{a,b}, {a,c}, {a,d}, {a,e}, {a,f}, {b,c}, {b,d}, {b,e}, {b,f}, {c,d}, {c,e}, {c,f}, {d,e}, {d,f}, {e,f}.**
- To calculate the rand index, we are interested in two values:
  - Number of pairs of elements **are** in the same cluster for both actual and predicted clustering.
  - Number of pairs of elements **are not** in the same cluster for both actual and predicted clustering.

Rand index

- The elements in pair **{a, b}** are in the same cluster for both actual and predicted. The other pair that fits this description is **{e, f}** (Total of 2 pairs).

- The elements in pair {a, d} are in different clusters for both actual and predicted clustering. The other pairs that fit this description are **{a,e}, {a,f}, {b,d}, {b,e}, {b,f}, {c,e}, {c,f}** (Total of 8 pairs)

- We can now introduce the formula for rand index: $$R = \frac{a+b}{\binom{n}{2}}$$

Rand index

- ***a*** is the number of times a pair of elements are in the same cluster for both actual and predicted clustering which we calculate as 2.
- ***b*** is the number of times a pair of elements are not in the same cluster for both actual and predicted clustering which we calculate as 8.
- The expression in the denominator is the total number of binomial coefficients - 15.
- Thus, rand index in this case is 10 / 15 = 0.67

Conclusion

- We have covered 3 commonly used evaluation metrics for clustering models.
- Evaluating a model is just as important as creating it. Without a robust and thorough evaluation, we might get unexpected results after the model is deployed.
- A comprehensive understanding of the evaluation metrics is essential to efficiently and appropriately use them.