Clustering Accuracy

Moo K. Chung

University of Wisconsin-Madison, USA

mkchung@wisc.edu

Abstract. In this short lecture, we explain how to compute the clustering accuracy in general k cluster settings in Matlab.

1 Basics on Clustering

Clustering is the process of partitioning a dataset into groups such that data points within a cluster share more similarity with each other than with those in other clusters. The goal of clustering is to uncover underlying structures in data. Various clustering algorithms exist, including k-means clustering, hierarchical clustering, and density-based clustering, each suited for different types of data and applications.

A commonly used method is k-means clustering, where the data is partitioned into k clusters by minimizing the sum of squared Euclidean distances between points and their respective cluster centroids. The algorithm iteratively updates cluster assignments and centroids until convergence. Another approach is hierarchical clustering, which builds a tree of nested clusters through either agglomerative (bottom-up) or divisive (top-down) strategies. Density-based clustering, such as DBSCAN, identifies clusters as dense regions separated by lower-density areas, making it robust to noise and capable of finding arbitrarily shaped clusters.

Given a dataset $X = \{x_1, x_2, \ldots, x_n\}$, clustering algorithms aim to assign each data point x_i to a cluster label $y_i \in \{1, \ldots, k\}$. The effectiveness of clustering is most often evaluated using the clustering accuracy when true labels are available.

2 Clustering Accuracy

Let y_i be the true classification label for the i-th data. Let $\widehat{y}_i$ be the estimate of y_i we determined from classification algorithms. Let $y = (y_1, \cdots, y_n)$ and $\widehat{y} = (\widehat{y}_1, \cdots, \widehat{y}_n)$. The classification accuracy $A(y, \widehat{y})$ is given by

$$A(\widehat{y}, y) = \frac{1}{n} \sum_{i=1}^{n} \mathbf{1}(\widehat{y} = y),$$

where $\mathbf{1}$ is the indicator function.

In clustering, there is no direct association between true clustering labels and predicted cluster labels. Given k clusters $C_1, \cdots, C_k$, its permutation $\pi(C_1), \cdots, \pi(C_k)$ is also a valid cluster for $\pi \in \mathbb{S}_k$, the permutation group of order k. Suppose [1 1 2 1 1 3 3] is the estimated cluster labels when the true labels are [1 1 1 2 2 3 3]. Then any permutation of estimated cluster labels such as [2 2 1 2 2 3 3] and [3 3 1 3 3 2 2] are other valid cluster labels. There are $k!$ possible permutations in $\mathbb{S}_k$ (Chung et al. 2019). Thus the clustering accuracy is modified as

$$A(\widehat{y}, y) = \frac{1}{n} \max_{\pi \in \mathbb{S}_k} \sum_{i=1}^{n} \mathbf{1}(\pi(\widehat{y}) = y).$$

This a modification to an assignment problem and can be solved using the Hungarian algorithm in $\mathcal{O}(k^3)$ run time (Edmonds & Karp 1972). In Matlab, it can be solved using `confusionmat.m`, which tabulates misclustering errors between the true cluster labels and predicted cluster labels. The confusion matrix $C(\widehat{y}, y)$ is a matrix of size $k \times k$ tabulating the correct number of clustering in each cluster. The diagonal entries show the correct number of clustering while the off-diagonal entries show the incorrect number of clusters. In Matlab, it can be computed using `confusionmat.m`:

```
ytrue = [ 1 1 1  2 2 3 3]
ypred = [ 1 1 2  1 1 3 3]
C = confusionmat(ypred, ytrue)

C =
     2     2     0
     1     0     0
     0     0     2
```

Alternately, we can compute the confusion matrix by simply counting the number of correct clustering:

```
C=zeros(k);
n=length(ytrue);
for i=1:n
    C(ypred(i),ytrue(i))=C(ypred(i),ytrue(i))+1;
end
```

To compute the clustering accuracy, we need to sum the diagonal entries. But the above matrix `C` is one possible confusion matrix. Under the permutation of cluster labels, we can get different confusion matrices. For large k, it is prohibitive expensive to search for all permutations. Thus we need to maximize the sum of diagonals of the confusion matrix under permutation with weight $C = (c_{ij})$:

$$\frac{1}{n} \max_{Q \in \mathbb{S}_k} \operatorname{tr}(QC) = \frac{1}{n} \max_{Q \in \mathbb{S}_k} \sum_{i,j} q_{ij} c_{ij}, \tag{1}$$

where $Q = (q_{ij})$ is the permutation matrix consisting of entries 0 and 1 such that there is exactly single 1 in each row and each column. This is a linear sum assignment problem (LSAP), a special case of linear assignment problem (Bougleux & Brun 2016). LSAP is solved using `matchpairs.m` in Matlab (Duff & Koster 2001):

```
M=matchpairs(C, 0, 'max');

M =
     2     1
     1     2
     3     3

accuracy = sum(C(sub2ind(size(C), M(:,1), M(:,2))))/n

accuracy=
   0.7143
```

The whole procedure is packaged into a single Matlab function `clustering_accuracy.m`, which can be downloaded from http://pages.stat.wisc.edu/~mchung/dynamicTDA/matlab/clustering_accuracy.m.

Bibliography

Bougleux, S. & Brun, L. (2016), ‘Linear sum assignment with edition’, *arXiv preprint arXiv:1603.04380* .

Chung, M., Xie, L., Huang, S.-G., Wang, Y., Yan, J. & Shen, L. (2019), Rapid acceleration of the permutation test via transpositions, *in* ‘International Workshop on Connectomics in Neuroimaging’, Vol. 11848, Springer, pp. 42–53.

Duff, I. & Koster, J. (2001), ‘On algorithms for permuting large entries to the diagonal of a sparse matrix’, *SIAM Journal on Matrix Analysis and Applications* **22**(4), 973–996.

Edmonds, J. & Karp, R. (1972), ‘Theoretical improvements in algorithmic efficiency for network flow problems’, *Journal of the ACM (JACM)* **19**, 248–264.