

Clustering Accuracy

Moo K. Chung

University of Wisconsin-Madison, USA

mkchung@wisc.edu

Abstract. We explain how to compute clustering accuracy in general k clusters in Matlab.

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

$$A(\hat{y}, y) = \frac{1}{n} \sum_{i=1}^n \mathbf{1}(\hat{y}_i \neq y_i),$$

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, \dots, C_k , its permutation $\pi(C_1), \dots, \pi(C_k)$ 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

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

This a modification to assignment problem can be solved using 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 $C(\hat{y}, y)$ 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 $C = (c_{ij})$:

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

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

```

Acknowledgements

We would like to thank Botao Wang of Xi'an Jiaotong University, China for discussion of the problem. This study is funded by NIHR01 EB02875 and NSF MDS-2010778.

Bibliography

- Bogleux, 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.