DS 5230
Unsupervised Machine Learning and Data Mining

Assignment 4 – External Indices in Clustering

Steve Morin, Ph.D.
s.morin@northeastern.edu

Instructions

1. Assignment 4 is a continuation of the midterm exam.
2. You will need to refer to the dim_red_and_clustering_midterm.pdf document (midterm pdf) to understand assignment 4 instructions.
3. In assignment 4 you will start by characterizing your midterm solution using the adjusted rand score and the best contingency matrix.
4. The best contingency matrix is formed by assigning the best cluster labels. See Appendix 1 to better understand these concepts.
5. Include a section in your pdf that summarizes your midterm solution. That section should include:
   a. a tabulation of your midterm solution (see page 17 of the midterm pdf). Add the adjusted rand score to the tabulation.
   b. the best contingency matrix
   c. an analysis of the error being made by your clustering solution (include images of the digits)

Instructions (continued)

6. Next you will rerun your midterm code with n_components equal to the dimensionality of the digits latent manifold determined in assignment 3. We will call this second solution your assignment 3 clustering solution.

7. You will characterize this second solution using the adjusted rand score and the best contingency matrix.

8. Include a section in your pdf that summarizes your assignment 3 solution. That section should include:

    a. a tabulation of your assignment 3 clustering solution (see page 17 of the midterm pdf). Add the adjusted rand score to the tabulation.

    b. the best contingency matrix

    c. an analysis of the error being made by your clustering solution (include images of the digits)

7. Include a section in your pdf that tabulates your midterm solution and your assignment 3 clustering solution. The tabulation should be structured as follows:

Instructions (continued)

number of classes in the digits data set	UMAP n_components	UMAP min_dist	UMAP n_neighbors	UMAP metric	trust-worthiness	clustering algorithm	number of clusters found	validity index or silhouette score	adjusted rand score

8. Make a data frame exactly like the table above and save it to a csv.

Hint: Returning the Clustering Labels

Modify the return df_row_dict from the clustering function to include the clustering labels from the clustering.

Refer to pages 10 through 12 in the midterm pdf for this step.

Hint: Drop the -1 cluster labels from DBSCAN Clustering Labels Before Computing the Adjusted Rand Score and the Best Contingency Table

See cell 11 in lecture_9_lab_2 included with assignment 4.

Deliverables

Submit the following to the canvas portal:

- Your .pdf describing your work.
- Your .csv of tabulated results.
- Your jupyter notebook as a .ipynb file.
- Your jupyter notebook as a .html file.
- Your environment .yml file.
- Your clustering.py module.
- Your dimenionality_reduction.py module

Appendix 1

Clustering Label Assignment and the Contingency Matrix

Clustering Label Assignment and the Contingency Matrix

A clustering algorithm divides the data into different sets.

The name of each set is irrelevant as far as the the algorithm is concerned.

It simply assigns names such as cluster 1, cluster 2, ..., cluster k.

Any cluster assignment, where the cluster names are some permutation of the true classes, can be returned.

```
labels_true = [1, 1, 0, 0, 2, 2]
labels_pred = [0, 0, 1, 1, 2, 2]
```

An example where the cluster labels in labels_pred are a permutation of the true classes in labels_true.

Clustering Label Assignment and the Contingency Matrix (continued)

Assume we have a data set with a class attribute with values: [1,1,0,0,2,2].

Then we use a clustering method to cluster the data set and we get cluster labels: [0,0,1,1,2,2].

If we use the contingency matrix to evaluate the clustering, we should score perfectly.

Instead, as can be seen on the right, what occurs is we get a less then perfect score due to the cluster label mismatch between class 0 and 1.

A perfect contingency matrix is diagonal.

Perfect Clustering

```
labels_true = [0, 0, 1, 1, 2, 2]
labels_pred = [0, 0, 1, 1, 2, 2]
```

cluster

$$\text{class}\quad \begin{array}{c|c} & \begin{array}{ccc} 0 & 1 & 2 \end{array} \\ \hline \begin{array}{c} 0 \\ 1 \\ 2 \end{array} & \begin{bmatrix} 2 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 2 \end{bmatrix} \end{array}$$

```
labels_true = [1, 1, 0, 0, 2, 2]
labels_pred = [0, 0, 1, 1, 2, 2]

labels_true has 3 classes: {0, 1, 2} - these form the rows of the contingency matrix
labels_pred has 3 clusters: {0, 1, 2} - these form the columns of the contingency matrix

array([[0, 2, 0],
       [2, 0, 0],
       [0, 0, 2]])
```

Clustering Label Assignment and the Contingency Matrix (continued)

The output on this slide demonstrates the permutation of the clustering labels [0, 0, 1, 1, 0, 2].

Each permutation is scored against the truth labels [0, 0, 1, 1, 2, 2].

There are three unique cluster labels so there are six permutations.

The external indices do not change as a function of cluster label permutation.

The contingency matrix and its trace do change as a function of cluster label permutation.

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix}$$

$$tr(A) = a_{11} + a_{22} + a_{33}$$

	permutation	mapping	jaccard_coefficient	rand_score	adjusted_rand_score	entropy	f_measure	purity	contingency_matrix	contingency_matrix_trace
0	original	{0: 0, 1: 1, 2: 2}	0.4	0.8	0.444444	0.459148	0.822222	0.833333	[[2, 0, 0], [0, 2, 0], [1, 0, 1]]	5
1	1	{0: 0, 1: 2, 2: 1}	0.4	0.8	0.444444	0.459148	0.822222	0.833333	[[2, 0, 0], [0, 0, 2], [1, 1, 0]]	2
2	2	{0: 1, 1: 0, 2: 2}	0.4	0.8	0.444444	0.459148	0.822222	0.833333	[[0, 2, 0], [2, 0, 0], [0, 1, 1]]	1
3	3	{0: 1, 1: 2, 2: 0}	0.4	0.8	0.444444	0.459148	0.822222	0.833333	[[0, 2, 0], [0, 0, 2], [1, 1, 0]]	0
4	4	{0: 2, 1: 0, 2: 1}	0.4	0.8	0.444444	0.459148	0.822222	0.833333	[[0, 0, 2], [2, 0, 0], [0, 1, 1]]	1
5	5	{0: 2, 1: 1, 2: 0}	0.4	0.8	0.444444	0.459148	0.822222	0.833333	[[0, 0, 2], [0, 2, 0], [1, 0, 1]]	3

Clustering Label Assignment and the Contingency Matrix (continued)

The contingency matrix changes column order as the cluster labels are permuted.

The maximum trace gives the contingency matrix that is closest to perfect.

```
*****************************************************

original – labels_pred are not permuted
labels_true: [0, 0, 1, 1, 2, 2]
labels_pred: [0, 0, 1, 1, 0, 2]

contingency_matrix:
[[2 0 0]
 [0 2 0]
 [1 0 1]]

contingency_matrix_trace:
5

*****************************************************

labels_pred are permuted
labels_true:      [0, 0, 1, 1, 2, 2]
labels_pred:      [0, 0, 1, 1, 0, 2]
perm_labels_pred: [0, 0, 2, 2, 0, 1]
mapping (original -> permuted): {0: 0, 1: 2, 2: 1}

contingency_matrix:
[[2 0 0]
 [0 0 2]
 [1 1 0]]

contingency_matrix_trace:
2
```

```
*****************************************************

labels_pred are permuted
labels_true:      [0, 0, 1, 1, 2, 2]
labels_pred:      [0, 0, 1, 1, 0, 2]
perm_labels_pred: [1, 1, 0, 0, 1, 2]
mapping (original -> permuted): {0: 1, 1: 0, 2: 2}

contingency_matrix:
[[0 2 0]
 [2 0 0]
 [0 1 1]]

contingency_matrix_trace:
1

*****************************************************

labels_pred are permuted
labels_true:      [0, 0, 1, 1, 2, 2]
labels_pred:      [0, 0, 1, 1, 0, 2]
perm_labels_pred: [1, 1, 2, 2, 1, 0]
mapping (original -> permuted): {0: 1, 1: 2, 2: 0}

contingency_matrix:
[[0 2 0]
 [0 0 2]
 [1 1 0]]

contingency_matrix_trace:
0
```

```
*****************************************************

labels_pred are permuted
labels_true:      [0, 0, 1, 1, 2, 2]
labels_pred:      [0, 0, 1, 1, 0, 2]
perm_labels_pred: [2, 2, 0, 0, 2, 1]
mapping (original -> permuted): {0: 2, 1: 0, 2: 1}

contingency_matrix:
[[0 0 2]
 [2 0 0]
 [0 1 1]]

contingency_matrix_trace:
1

*****************************************************

labels_pred are permuted
labels_true:      [0, 0, 1, 1, 2, 2]
labels_pred:      [0, 0, 1, 1, 0, 2]
perm_labels_pred: [2, 2, 1, 1, 2, 0]
mapping (original -> permuted): {0: 2, 1: 1, 2: 0}

contingency_matrix:
[[0 0 2]
 [0 2 0]
 [1 0 1]]

contingency_matrix_trace:
3
```

Clustering Label Assignment and the Contingency Matrix (continued)

When clustering error analysis is required getting the contingency matrix that is closet to perfect is a good place to start.

```
******************************************************
original - labels_pred are not permuted
labels_true: [0, 0, 1, 1, 2, 2]
labels_pred: [0, 0, 1, 1, 0, 2]

contingency_matrix:
[[2 0 0]
 [0 2 0]
 [1 0 1]]

contingency_matrix_trace:
5
```

get the best cluster label permutation

```
return_dict = ei.get_best_cluster_label_permutation(labels_pred, labels_true, print_out=False)
best_contingency_matrix = return_dict['best_contingency_matrix']
best_contingency_matrix
```

```
array([[2, 0, 0],
       [0, 2, 0],
       [1, 0, 1]])
```

Clustering Label Assignment and the Contingency Matrix (continued)

If we assume that class annotation is accurate the contingency matrix provides an opportunity to analyze clustering errors.

```
clustering_contingency_matrix

labels_true has 2 classes - these form the rows of the contingency matrix
labels_pred has 3 clusters - these form the columns of the contingency matrix

contingency_matrix:
[[2 1 0]
 [0 1 2]]
```

Consider the contingency matrix above.

The most obvious clustering error is that 3 clusters are found when only 2 classes exist.

A further observation is that clusters 0 and 2 are pure, that is, they contain objects of a single class.

Cluster 1 is not pure.

Further clustering error analysis might include an investigation of the cluster 1 data objects.

Clustering Label Assignment and the Contingency Matrix (continued)

If we assume that class annotation is accurate the contingency matrix provides an opportunity to analyze clustering errors.

```
clustering_contingency_matrix

labels_true has 3 classes - these form the rows of the contingency matrix
labels_pred has 2 clusters - these form the columns of the contingency matrix

contingency_matrix:
[[2 1]
 [0 1]
 [1 1]]
```

Consider the contingency matrix above.

The most obvious clustering error is that only 2 clusters were found when 3 classes exist.

A further observation is that none of the clusters are pure, that is, both clusters contain objects from more then one class.

Further clustering error analysis might include an investigation of the least pure cluster.

See lecture_9_lab_3 included with assignment 4 to learn more.