UNIVERSITY OF INFORMATION TECHNOLOGY
FACULTY OF COMPUTER SCIENCE

CLASS
MACHINE LEARNING IN COMPUTER VISION

PRACTICAL EXCERCISE

CLUSTERING

Lecturer: DR. LÊ ĐÌNH DUY
Student: Trần Quốc Long
Student ID: 14520490

Ho Chi Minh, October 19th, 2017

CONTENT

I. What is clustering [1.2]?

- **Clustering** if the process of grouping data into classes or clusters.

The grouping is done in such a manner that the objects within the same cluster are very similar to each other but they are very dissimilar to the objects in some other cluster.

- Clustering is a form of "learning by observation". It is an unsupervised learning method and does not require a training data set to generate a model. Clustering can lead to the discovery of previously unknown groups within the data.

- For example, in business intelligence, clustering can be used to organize a large number of customers into groups, where customers within a group share strong similar characteristics.

1. K-means algorithm [1.3]:

Steps:

1. *k* initial "means" (in this case *k*=3) are randomly generated within the data domain (shown in color).
2. *k* clusters are created by associating every observation with the nearest mean. The partitions here represent the Voronoi diagram generated by the means.
3. The centroid of each of the *k* clusters becomes the new mean.
4. Steps 2 and 3 are repeated until convergence has been reached.

II. Clustering measurement methods:

1. V-measure score [1.4]

- V-measure cluster labeling given a ground truth.
- This score is identical to **normalized_mutual_info_score**.
- The V-measure is the harmonic mean between homogeneity and completeness:

```
v = 2 * (homogeneity * completeness) / (homogeneity + completeness)
```

- This metric is independent of the absolute values of the labels: a permutation of the class or cluster label values won't change the score value in any way.
- This metric is furthermore symmetric: switching label_true with label_pred will return the same score value. This can be useful to measure the agreement of two independent label assignments strategies on the same dataset when the real ground truth is not known.

2. Homogeneity score [1.5]

- Homogeneity metric of a cluster labeling given a ground truth.
- A clustering result satisfies homogeneity if all of its clusters contain only data points which are members of a single class.
- This metric is independent of the absolute values of the labels: a permutation of the class or cluster label values won't change the score value in any way.
- This metric is not symmetric: switching label_true with label_pred will return the **completeness_score** which will be different in general.

3. Completeness score [1.6]

- Completeness metric of a cluster labeling given a ground truth.
- A clustering result satisfies completeness if all the data points that are members of a given class are elements of the same cluster.
- This metric is independent of the absolute values of the labels: a permutation of the class or cluster label values won't change the score value in any way.
- This metric is not symmetric: switching label_true with label_pred will return the **homogeneity_score** which will be different in general.

4. Adjusted rand score (ARI) [1.7]

- Rand index adjusted for chance.
- The Rand Index computes a similarity measure between two clusterings by considering all pairs of samples and counting pairs that are assigned in the same or different clusters in the predicted and true clusterings.
- The raw RI score is then "adjusted for chance" into the ARI score using the following scheme:

```
ARI = (RI - Expected_RI) / (max(RI) - Expected_RI)
```

- The adjusted Rand index is thus ensured to have a value close to 0.0 for random labeling independently of the number of clusters and samples and exactly 1.0 when the clusterings are identical (up to a permutation).
- ARI is a symmetric measure:

```
adjusted_rand_score(a, b) == adjusted_rand_score(b, a)
```

5. Adjusted mutual info score [1.8]

- Adjusted Mutual Information between two clusterings.
- Adjusted Mutual Information (AMI) is an adjustment of the Mutual Information (MI) score to account for chance. It accounts for the fact that the MI is generally higher for two clusterings with a larger number of clusters, regardless of whether there is actually more information shared. For two clusterings U and V, the AMI is given as:

```
AMI(U, V) = [MI(U, V) - E(MI(U, V))] / [max(H(U), H(V)) - E(MI(U, V))]
```

- This metric is independent of the absolute values of the labels: a permutation of the class or cluster label values won't change the score value in any way.
- This metric is furthermore symmetric: switching `label_true` with `label_pred` will return the same score value. This can be useful to measure the agreement of two independent label assignments strategies on the same dataset when the real ground truth is not known.
- Be mindful that this function is an order of magnitude slower than other metrics, such as the Adjusted Rand Index.

6. Silhouette Score [1.9]

- Compute the mean Silhouette Coefficient of all samples.
- The Silhouette Coefficient is calculated using the mean intra-cluster distance (`a`) and the mean nearest-cluster distance (`b`) for each sample. The Silhouette Coefficient for a sample is `(b - a) / max(a, b)`. To clarify, `b` is the distance between a sample and the nearest cluster that the sample is not a part of. Note that Silhouette Coefficient is only defined if number of labels is 2 <= n_labels <= n_samples - 1.
- This function returns the mean Silhouette Coefficient over all samples. To obtain the values for each sample, use **`silhouette_samples`**.

- The best value is 1 and the worst value is -1. Values near 0 indicate overlapping clusters. Negative values generally indicate that a sample has been assigned to the wrong cluster, as a different cluster is more similar.

III. Environments:

- Python: ver. 3.6.2
- Jupyter notebook.

IV. Functions from scikit-learn library for clustering problem

* Taken examples were applied with dataset of handwriten digits from sklearn

1. K-means:

```
#Kmeans
nClusters = 10
kmeans_model = KMeans(nClusters)
labels_kmeans = kmeans_model.fit_predict(digits.data)
```

2. Spectral clustering

```
#Spectral_clustering
graph = cosine_similarity(digits.data)
labels_spectral = spectral_clustering(graph, n_clusters=10)
```

3. DBSCAN

```
#DBSCAN
data = digits.data
labels_dbscan = DBSCAN(eps=0.06, min_samples=10,metric = 'cosine').fit_predict(data)
```

4. Agglomerative

```
#Agglomerative Clustering

Agglomerative_model = AgglomerativeClustering(n_clusters = nClusters)

labels_AgglomerativeClustering = Agglomerative_model.fit_predict(data)
```

5. Clustering method measurement

```
metrics.homogeneity_score(digits.target, labels),

metrics.completeness_score(digits.target, labels),

metrics.v_measure_score(digits.target, labels),

metrics.adjusted_rand_score(digits.target, labels),

metrics.adjusted_mutual_info_score(digits.target,  labels),
```

V. Applying clustering function on datasets

At this part, I applied clustering methods on datasets:

- Handwritten digits
- Face dataset: lfw_people
- The German Traffic Sign Recognition Benchmark
- Columbia University Image Library (COIL-20)

The datasets can download from:

- http://benchmark.ini.rub.de/?section=gtsrb&subsection=news
- http://www.cs.columbia.edu/CAVE/software/softlib/coil-20.php

VI. Brief comparison of clustering methods

* Taken examples were applied with dataset of handwriting digits from sklearn

```
n_digits: 10,    n_samples 1797,         n_features 64
_____________________________________________________________________________________
init            time    homo    compl   v-meas  ARI     AMI     silhouette
K-means         0.18s   0.740   0.748   0.744   0.669   0.737   0.178
spectral        0.48s   0.713   0.717   0.715   0.625   0.710   0.164
Agg.            0.20s   0.858   0.879   0.868   0.794   0.856   0.180
DBSCAN          0.04s   0.709   0.757   0.732   0.520   0.706   0.150
```

VII. Folder structure submitted on Github:

Link to source on Github: https://github.com/tranquoclongt1/ML/

Details:

- BT1, BT2, BT3, BT4: source code for each exercise + report of them.
- Bao_cao: report for the whole homework.

VIII. References:

1. https://machinelearningcoban.com/2017/01/01/kmeans/
2. https://www.quora.com/What-is-clustering
3. https://en.wikipedia.org/wiki/K-means_clustering
4. http://scikit-learn.org/stable/modules/generated/sklearn.metrics.v_measure_score.html
5. http://scikit-learn.org/stable/modules/generated/sklearn.metrics.homogeneity_score.html
6. http://scikit-learn.org/stable/modules/generated/sklearn.metrics.completeness_score.html
7. http://scikit-learn.org/stable/modules/generated/sklearn.metrics.adjusted_rand_score.html
8. http://scikit-learn.org/stable/modules/generated/sklearn.metrics.adjusted_mutual_info_score.html
9. http://scikit-learn.org/stable/modules/generated/sklearn.metrics.silhouette_score.html