CCS-E4300 – Methods of Data Mining D

Project 1

Ayoub Chouak - 881216

Joseph Attieh - 915755

Submitted on Dec 06,2020

Requirements:

In this project, we are required to cluster two datasets as good as possible by trying out different configurations of distance measures, feature selection/reduction techniques and clustering methods in order to achieve the optimal Normalized Mutual Information (NMI) with respect to the ground truth labels provided.
The datasets are described in the table below:

Dataset	Description	Number of objects	Number of features	Target number of classes
Gene Data	Gene expression data	795	7002	5
MS Data	Mass spectrometry data	694	5002	3

Methods:

We are required to compare at least two configurations per dataset. Therefore, we will be presenting two methods that achieved the best results per dataset. Then, we will be presenting a table containing the NMI achieved by other non-optimal configurations. We should note that our results are the best ones achieved in term of NMI.

a) Gene dataset:

As mentioned previously, the gene dataset is an artificially generated dataset which deals with gene expression data.
In order to get an idea about the distribution of the objects within the target classes, we proceed by visualizing the dataset. This is done by first normalizing the dataset using the *scikit-learn* library for each sample to have unit norm. This type of normalization is usually useful whenever the user wants to rescale the vector of each sample to have unit norm, independently of the distribution of the samples. Furthermore, it is quite common in text classification and clustering problems where the Vector Space Model is used. After normalizing the data, we use both PCA and T-SNE to visualize the data in a 2-D space:

- Principal Component Analysis: the PCA method introduced in class is a deterministic linear dimensionality reduction technique hat tries to preserve the global structure of the data by projecting the data on the principal components to preserve a certain amount of variance. This method is prone to lose local structures which can render it inefficient.
- t-distributed stochastic neighbourhood embedding: the t-SNE method is a non-deterministic non-linear dimensionality reduction technique that embeds the points in a lower dimension in which the neighborhood of the point is preserved. This method preserves the local structure of the data but is not able to preserve variance.

Let's visualize our data by reducing the number of dimensions to 2. The visualizations of the 2D data as well as the ground truth labels are showed in *Figure 1* and *Figure 2*. As we can see from both figures, the T-SNE performs a better job preserving the clusters in the dataset. Therefore, it would make more sense to perform the clustering in the 2-D space where the features are generated from T-SNE. Having this clear cluster separation would give us satisfactory NMI results using any clustering method that works well with globular well-separated clusters. Our goal is to find the clustering method which give us the best NMI.

First method:

The first method uses agglomerative clustering with ward linkage to cluster the data into 5 clusters. Ward's linkage, or minimal increase of sum-of-squares ward minimizes the total within-cluster variance. This method works by putting each object in its own cluster and at each step trying to find the pair of clusters that leads to minimum increase in total within-cluster variance after merging. This increase is a weighted squared distance between cluster centers. This will result in very good clustering results whenever the clusters are globular and well-seperated. The Euclidian distance suits ward's linkage and appear to be a good distance measure to separate the clusters.

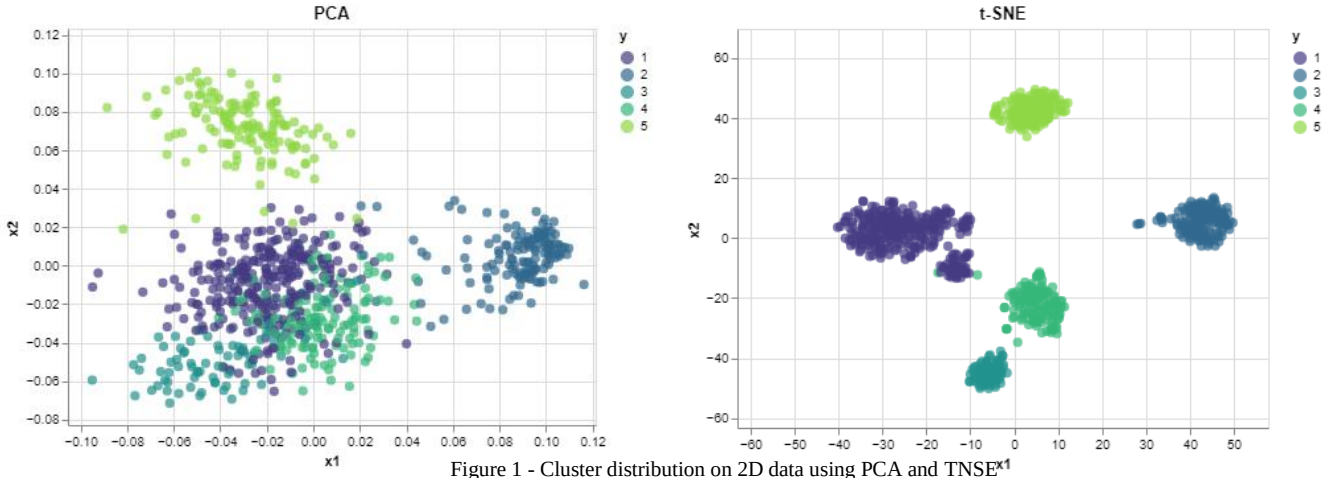

Figure 1 - Cluster distribution on 2D data using PCA and TNSE

Second method:

The second method uses K-means with K=5 to perform clustering. K-means usually produce tighter clusters than hierarchical clustering, especially if the clusters are globular. We will be using the Euclidian distance since it suits the K-means algorithm and our data.

Results:

	Hierarchical + Ward	**K-means with K=5**
Silhouette Score	0.739308	0.738015
Davies-Bouldin Score	0.325233	0.328265
NMI	0.990635	0.984800

We can use the Silhouette and the Davies-Bouldin indexes since they have the same objective that we want to achieve. In fact, those metrics tend to be better metrics when the resulting clusters are more globular, compact and better separated, which matches the clusters that we want to get in the figure.

The first method has a better Silhouette score compared to the second method which means that the resulting clusters of the first method are denser and better separated than the resulting ones from the second method (since the Silhouette score evaluates how closely each item is close to its own cluster and how loosely it is to the closest neighboring cluster). Furthermore, the first method has a better Davies Bouldin score compared to the second method which means that the resulting clusters of the first method are better separated than the ones in the second method.

On a final note, we can see that the normalized mutual information score of the first method is better than the second method's score. That means that the first method is indeed better than the second as it produces a set of clusters that are similar to the ground truth labels (according to the external validation metrics). This validates what we have implied from the internal validation metrics.

b) MS Data:

As mentioned previously, the gene dataset is an artificially generated dataset which deals with mass spectrometry data. In order to get an idea about the distribution of the objects within the target classes, we proceed by visualizing the dataset by first normalizing the data (like we did with the Gene data) then applying using PCA and T-SNE to get the 2D feature representation of the data. The plots are showed in the next page.

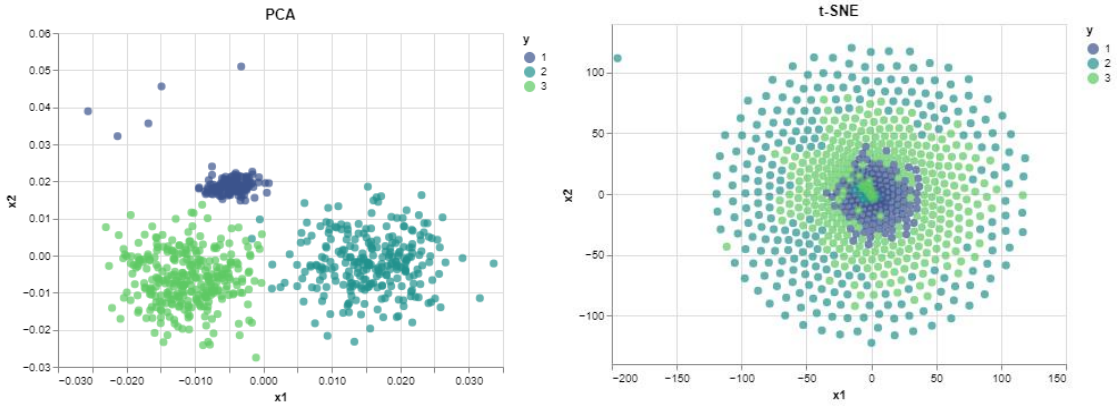

Figure 2 - Cluster distribution on 2D data using PCA and TNSE

As we can see from the figures above, the PCA produces better results as it maintains the shape of the clusters. Most of the clustering algorithms that we know are best at detecting globular clusters, which makes the PCA even more suitable for this dataset. We should note that using Spectral Clustering with T-SNE is not a a better alternative, especially that the clusters are not clearly defined.

First method:

The first method uses agglomerative clustering with ward linkage to cluster the data into 3 clusters. As we have mentioned previously, this will result in very good clustering results whenever the clusters are globular and well-separated (which is our case). The Euclidian distance suits ward's linkage and appear to be a good distance measure to separate the clusters.

Second method:

The second method uses K-means with K=3 to perform clustering. K-means usually produce tighter clusters than hierarchical clustering, especially if the clusters are globular. We will be using the Euclidian distance since it suits the K-means algorithm and our data.

Results:

	Hierarchical + Ward	**K-means with K=3**
Silhouette Score	0.624004	0.625858
Davies-Bouldin Score	0.501754	0.498606
NMI	0.950785	0.922516

We can use the Silhouette and the Davies-Bouldin indexes since they have the same objective that we want to achieve. The second method has a better Silhouette score and a better Davies-Bouldin score compared to the second method which means that the resulting clusters of the second method are denser and better separated than the resulting ones from the first method. That makes sense especially that K-means tend to produce tighter clusters. However, the normalized mutual information score of the first method is better than the second method's score. That means that, even though the internal validation metrics favors the second method, the first method is able to better detect the clusters in our data as the resulting clusters are the closest to the ground truth labels. The hierarchical clustering with ward linkage seems to be better than K-means in our case.

Other configurations:

As we have mentioned previously, we verified that our results were optimal by trying out different combinations of methods. We populated the data by performing multiple clustering configurations on different sets of features. We tried performing hierarchical clustering with *single*, *complete*, *average* and *ward* linkages in combination with the *Euclidean, Cosine* and *Manhattan* distances. We also tried performing K-means and spectral clustering (not documented since it performed poorly). Furthermore, we varied the number of target clusters between 2 and 11. Trying those configurations on the original dataset yielded unsatisfactory results, so we decided to try those clustering configurations on a number of dimensions varying from 1 to 100 with PCA and from 1 to 10 with TNSE. The configurations yielding the maximum NMI values are showed in the tables in the appendix. As we can see, the agglomerative clustering applied on the two-dimensional data (whether PCA or TNSE) achieved the best results on both datasets. This make perfect sense as we have seen that the PCA/TNSE yielded globular clusters that are suitable with those algorithms. And we should note that it happened that the number of clusters detected maximizing the NMI matched the number of clusters in the ground truth. We could see the variation of the NMI with the number of clusters detected using agglomerative with ward linkage in the graphs in the appendix. The graphs clearly show that the maximum NMI is achieved when the number of clusters match the number of ground truth labels.

Conclusions:

As we can see, we can imply the following:

- Depending on the dataset, we should use a linear or non-linear feature reduction technique to be able to bring the data to a lower dimension where we could visualize it and apply the clustering techniques that we know. In the first dataset, the clusters were conserved using a non-linear feature reduction technique (t-SNE) while in the second dataset, the clusters were conserved using a linear feature reduction technique (PCA).
- The hierarchical clustering with ward linkage performed better than K-means in both datasets. We cannot generalize the fact that it is better in capturing globular clusters since the internal validation metrics did not match the external validation metrics. This might be due to the dimensionality reduction process that is prone to data loss.
- We can see that in our case, the NMI was maximum when the number of clusters detected was equal to the number of ground truth labels. We should note that this is not always the case.

Brief instructions how to run:

- Make sure to open the Jupyter notebook file and run the cells within (*final submission* for compared approaches and *exploring the data* for the data in the appendix).
- We should note that we have used sklearn to normalize the data, perform the clustering and to compute the clustering evaluation metrics and seaborn and matplotlib to visualize the data.

Appendix:

Gene Dataset								
	Distance	Linkage	PCA dimensions	Number of clusters	NMI	TNSE dimensions	Number of clusters	NMI
Agglomerative clustering	Euclidean	Single	6	6	0.53526	2	6	0.988197
		Complete	10	7	0.879363	2	5	0.990635
		Average	9	9	0.906341	2	5	0.990635
		Ward	82	5	0.959532	2	5	0.990635
	Cosine	Single	5	8	0.744114	2	5	0.931465
		Complete	15	7	0.871866	2	4	0.929698
		Average	21	5	0.967516	2	4	0.929698
	Manhattan	Single	5	9	0.524102	2	6	0.988197
		Complete	5	9	0.74843	2	6	0.944572
		Average	3	9	0.822957	2	5	0.990635
Kmeans	Euclidean	N/A	-	-	-	2	5	0.9848

Table 1 - NMI of optimal configurations on the first dataset

(varying the PCA dimensions between 1 and 100, TNSE dimensions between 1 and 10 and number of clusters between 2 and 10)

MS Dataset								
	Distance	Linkage	PCA dimensions	Number of clusters	NMI	TNSE dimensions	Number of clusters	NMI
Agglomerative clustering	Euclidean	Single	1	5	0.746328	1	4	0.152137
		Complete	1	2	0.760435	1	6	0.500554
		Average	2	3	0.941584	1	7	0.419503
		Ward	2	3	0.950785	1	9	0.530005
	Cosine	Single	3	5	0.947344	1	2	0.390995
		Complete	2	3	0.844027	1	2	0.390995
		Average	4	3	0.940033	1	2	0.390995
	Manhattan	Single	1	5	0.746328	1	4	0.152137
		Complete	2	6	0.767178	1	6	0.50054
		Average	3	3	0.921698	1	7	0.419503
Kmeans	Euclidean	N/A	2	3	0.9225158	-		-

Table 2 - NMI score of the optimal configurations on the second dataset

(varying the PCA dimensions between 1 and 100, TNSE dimensions between 1 and 10 and number of clusters between 2 and 10)

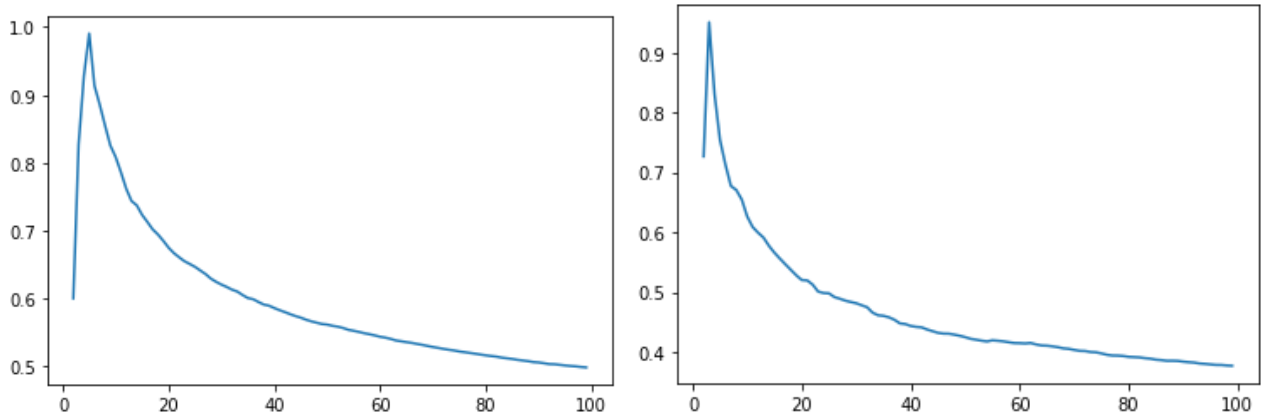

Figure 3 - Plot showing the variation of the NMI in function of the number of clusters we are trying to detect using agglomerative clustering