Aprendizagem 2022

Homework IV – Group 019

Diogo Gaspar 99207, Rafael Oliveira 99311

Part I: Pen and paper

Given the bivariate observations $\left\{ \begin{bmatrix} 1 \\ 2 \end{bmatrix}, \begin{bmatrix} -1 \\ 1 \end{bmatrix}, \begin{bmatrix} 1 \\ 0 \end{bmatrix} \right\}$, and the following multivariate Gaussian mixture:

$$\mu_1 = \begin{bmatrix} 2 \\ 2 \end{bmatrix}, \quad \mu_2 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \quad \Sigma_1 = \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix}, \quad \Sigma_2 = \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix}, \quad \pi_1 = \pi_2 = 0.5$$

1. **Perform one epoch of the EM clustering algorithm and determine the new parameters.**

   As a side note, we'll be using the k_1 and k_2 notation to represent clusters 1 and 2 - with that, we'll say that $\pi_1 = P(C = k_1)$, with analogous notation for π_2.

   EM-Clustering, being an unsupervised learning algorithm intending to calculate the probability of a sample belonging to a certain cluster, is a method that iteratively updates the parameters of the model until convergence is reached (for a given definition of convergence). Here, we'll perform exactly one epoch of the algorithm, which means we'll be going through two steps:

   - **E-step:** Here, we're aiming to calculate the **posterior probability** of each sample belonging to each cluster. In order to perform this calculation, we'll be using **Bayes' rule**, of course, to decompose the posterior probability into the product of the **likelihood** and the **prior probability** of the sample belonging to the cluster. Let's try, then, to assign each sample to the cluster that maximizes the posterior probability.

     For starters, we must first note that the likelihood of a sample belonging to a cluster is given by the **multivariate Gaussian distribution**, which can be written as (considering $d = 2$):

     $$P(x_i \mid C = k_n) \sim \mathcal{N}(x_i; \mu_n, \Sigma_n) = \frac{1}{\sqrt{(2\pi)^d \det \Sigma_n}} \exp\left(-\frac{1}{2}(x - \mu_n)^T \Sigma_n^{-1} (x - \mu_n)\right)$$

     Moreover, in this step we'll use teal to denote the priors and purple to denote the likelihoods.

As a given, we have that the priors are (for every sample, of course):

$$P(C = k_1) = P(C = k_2) = 0.5$$

Regarding x_1, we have:

$$\begin{aligned}
P(x_1 \mid C = k_1) &= \frac{1}{\sqrt{(2\pi)^d \det \Sigma_1}} \exp\left(-\frac{1}{2}(x_1 - \mu_1)^T \Sigma_1^{-1} (x_1 - \mu_1)\right) \\
&= \frac{1}{\sqrt{(2\pi)^2 \det \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix}}} \exp\left(-\frac{1}{2}\left(\begin{bmatrix} 1 \\ 2 \end{bmatrix} - \begin{bmatrix} 2 \\ 2 \end{bmatrix}\right)^T \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix}^{-1} \left(\begin{bmatrix} 1 \\ 2 \end{bmatrix} - \begin{bmatrix} 2 \\ 2 \end{bmatrix}\right)\right) \\
&= 0.0658407
\end{aligned}$$

$$\begin{aligned}
P(x_1 \mid C = k_2) &= \frac{1}{\sqrt{(2\pi)^d \det \Sigma_2}} \exp\left(-\frac{1}{2}(x_1 - \mu_2)^T \Sigma_2^{-1} (x_1 - \mu_2)\right) \\
&= \frac{1}{\sqrt{(2\pi)^2 \det \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix}}} \exp\left(-\frac{1}{2}\left(\begin{bmatrix} 1 \\ 2 \end{bmatrix} - \begin{bmatrix} 0 \\ 0 \end{bmatrix}\right)^T \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix}^{-1} \left(\begin{bmatrix} 1 \\ 2 \end{bmatrix} - \begin{bmatrix} 0 \\ 0 \end{bmatrix}\right)\right) \\
&= 0.0227993
\end{aligned}$$

The (normalized) posteriors can be computed as follows:

$$\begin{aligned}
P(C = k_1 \mid x_1) &= \frac{P(C = k_1)P(x_1 \mid C = k_1)}{P(C = k_1)P(x_1 \mid C = k_1) + P(C = k_2)P(x_1 \mid C = k_2)} \\
&= \frac{0.5 \cdot 0.0658407}{0.5 \cdot 0.0658407 + 0.5 \cdot 0.0227993} \\
&= 0.742788
\end{aligned}$$

$$\begin{aligned}
P(C = k_2 \mid x_1) &= \frac{P(C = k_2)P(x_1 \mid C = k_2)}{P(C = k_1)P(x_1 \mid C = k_1) + P(C = k_2)P(x_1 \mid C = k_2)} \\
&= \frac{0.5 \cdot 0.0227993}{0.5 \cdot 0.0658407 + 0.5 \cdot 0.0227993} \\
&= 0.257212
\end{aligned}$$

Note that, with the aid of the total probability law, we can say that $P(C = k_1 \mid x_1) + P(C = k_2 \mid x_1) = 1$; going forward, we'll calculate the normalized posterior for k_2 utilizing this fact.

We can now repeat the same process for x_2:

$$\begin{aligned}
P(x_2 \mid C = k_1) &= \frac{1}{\sqrt{(2\pi)^d \det \Sigma_1}} \exp\left(-\frac{1}{2}(x_2 - \mu_1)^T \Sigma_1^{-1}(x_2 - \mu_1)\right) \\
&= \frac{1}{\sqrt{(2\pi)^2 \det \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix}}} \exp\left(-\frac{1}{2}\left(\begin{bmatrix} -1 \\ 1 \end{bmatrix} - \begin{bmatrix} 2 \\ 2 \end{bmatrix}\right)^T \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix}^{-1} \left(\begin{bmatrix} -1 \\ 1 \end{bmatrix} - \begin{bmatrix} 2 \\ 2 \end{bmatrix}\right)\right) \\
&= 0.00891057
\end{aligned}$$

$$\begin{aligned}
P(x_2 \mid C = k_2) &= \frac{1}{\sqrt{(2\pi)^d \det \Sigma_2}} \exp\left(-\frac{1}{2}(x_2 - \mu_2)^T \Sigma_2^{-1}(x_2 - \mu_2)\right) \\
&= \frac{1}{\sqrt{(2\pi)^2 \det \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix}}} \exp\left(-\frac{1}{2}\left(\begin{bmatrix} -1 \\ 1 \end{bmatrix} - \begin{bmatrix} 0 \\ 0 \end{bmatrix}\right)^T \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix}^{-1} \left(\begin{bmatrix} -1 \\ 1 \end{bmatrix} - \begin{bmatrix} 0 \\ 0 \end{bmatrix}\right)\right) \\
&= 0.0482662
\end{aligned}$$

The (normalized) posteriors can be computed as follows:

$$\begin{aligned}
P(C = k_1 \mid x_2) &= \frac{P(C = k_1)P(x_2 \mid C = k_1)}{P(C = k_1)P(x_2 \mid C = k_1) + P(C = k_2)P(x_2 \mid C = k_2)} \\
&= \frac{0.5 \cdot 0.00891057}{0.5 \cdot 0.00891057 + 0.5 \cdot 0.0482662} \\
&= 0.155843
\end{aligned}$$

Like stated above, using the total probability law, we can say that $P(C = k_1 \mid x_2) + P(C = k_2 \mid x_2) = 1$; therefore, $P(C = k_2 \mid x_2) = 1 - P(C = k_1 \mid x_2) = 0.844157$.

Finally, repeating the same process for x_3:

$$\begin{aligned}
P(x_2 \mid C = k_1) &= \frac{1}{\sqrt{(2\pi)^d \det \Sigma_1}} \exp\left(-\frac{1}{2}(x_3 - \mu_1)^T \Sigma_1^{-1}(x_3 - \mu_1)\right) \\
&= \frac{1}{\sqrt{(2\pi)^2 \det \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix}}} \exp\left(-\frac{1}{2}\left(\begin{bmatrix} 1 \\ 0 \end{bmatrix} - \begin{bmatrix} 2 \\ 2 \end{bmatrix}\right)^T \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix}^{-1} \left(\begin{bmatrix} 1 \\ 0 \end{bmatrix} - \begin{bmatrix} 2 \\ 2 \end{bmatrix}\right)\right) \\
&= 0.0338038
\end{aligned}$$

$$
\begin{aligned}
P(x_3 \mid C = k_2) &= \frac{1}{\sqrt{(2\pi)^d \det \Sigma_2}} \exp\left(-\frac{1}{2}(x_3 - \mu_2)^T \Sigma_2^{-1}(x_3 - \mu_2)\right) \\
&= \frac{1}{\sqrt{(2\pi)^2 \det \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix}}} \exp\left(-\frac{1}{2}\left(\begin{bmatrix} 1 \\ 0 \end{bmatrix} - \begin{bmatrix} 0 \\ 0 \end{bmatrix}\right)^T \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix}^{-1} \left(\begin{bmatrix} 1 \\ 0 \end{bmatrix} - \begin{bmatrix} 0 \\ 0 \end{bmatrix}\right)\right) \\
&= 0.061975
\end{aligned}
$$

The (normalized) posteriors can be computed as follows:

$$
\begin{aligned}
P(C = k_1 \mid x_3) &= \frac{P(C = k_1)P(x_3 \mid C = k_1)}{P(C = k_1)P(x_3 \mid C = k_1) + P(C = k_2)P(x_3 \mid C = k_2)} \\
&= \frac{0.5 \cdot 0.0338038}{0.5 \cdot 0.0338038 + 0.5 \cdot 0.061975} \\
&= 0.352936
\end{aligned}
$$

$$
P(C = k_2 \mid x_3) = 1 - P(C = k_1 \mid x_3) = 0.647064
$$

- **M-Step: Having calculated the posteriors, we can now update the parameters of the cluster-defining distributions.**

  For each cluster, we'll want to find the new distribution parameters: in this case, μ_k and Σ_k (for every cluster k). For likelihoods, we'll need to update both μ_k and Σ_k, using all samples weighted by their respective posteriors, as can be seen below; for priors, we'll need to perform a weighted mean of the posteriors.

$$
\begin{aligned}
\mu_k &= \frac{\sum_{i=1}^{3} P(C = k \mid x_i)x_i}{\sum_{i=1}^{3} P(C = k \mid x_i)} \\
\Sigma_k^{nm} &= \frac{\sum_{i=1}^{3} P(C = k \mid x_i)(x_{i,n} - \mu_{k,n})(x_{i,m} - \mu_{k,m})^T}{\sum_{i=1}^{3} P(C = k \mid x_i)} \\
P(C = k) &= \frac{\sum_{i=1}^{3} P(C = k \mid x_i)}{\sum_{c=1}^{2} \sum_{i=1}^{3} P(C = c \mid x_i)}
\end{aligned}
$$

In the equations stated above, we're considering $x_{i,n}$ as the n-th feature's value of the i-th sample, and $\mu_{k,n}$ as the n-th index of centroid μ_k.

We can now estimate the new parameters of the distributions (and the new priors) as can be seen below (note that the new μ_k's are used in the calculation of the new Σ_k's):
For k_1:

$$\begin{aligned}
\mu_1 &= \frac{\sum_{i=1}^{3} P(C = k_1 \mid x_i) x_i}{\sum_{i=1}^{3} P(C = k_1 \mid x_i)} \\
&= \frac{0.742788 \cdot \begin{bmatrix} 1 \\ 2 \end{bmatrix} + 0.155843 \cdot \begin{bmatrix} -1 \\ 1 \end{bmatrix} + 0.352936 \cdot \begin{bmatrix} 1 \\ 0 \end{bmatrix}}{0.742788 + 0.155843 + 0.352936} \\
&= \begin{bmatrix} 0.750964 \\ 1.31149 \end{bmatrix}
\end{aligned}$$

$$\begin{aligned}
\Sigma_1^{nm} &= \frac{\sum_{i=1}^{3} P(C = k_1 \mid x_i)(x_{i,n} - \mu_{k_1,n})(x_{i,m} - \mu_{k_1,m})^T}{\sum_{i=1}^{3} P(C = k_1 \mid x_i)} \\
&= \begin{bmatrix} 0.436053 & 0.0775726 \\ 0.0775726 & 0.778455 \end{bmatrix}
\end{aligned}$$

$$\pi_1 = P(C = k_1) = \frac{\sum_{i=1}^{3} P(C = k_1 \mid x_i)}{\sum_{c=1}^{2} \sum_{i=1}^{3} P(C = c \mid x_i)} = 0.417189$$

For k_2:

$$\begin{aligned}
\mu_2 &= \frac{\sum_{i=1}^{3} P(C = k_2 \mid x_i) x_i}{\sum_{i=1}^{3} P(C = k_2 \mid x_i)} \\
&= \frac{0.257212 \cdot \begin{bmatrix} 1 \\ 2 \end{bmatrix} + 0.844157 \cdot \begin{bmatrix} -1 \\ 1 \end{bmatrix} + 0.647064 \cdot \begin{bmatrix} 1 \\ 0 \end{bmatrix}}{0.257212 + 0.844157 + 0.647064} \\
&= \begin{bmatrix} 0.0343846 \\ 0.777028 \end{bmatrix}
\end{aligned}$$

$$\begin{aligned}
\Sigma_2^{nm} &= \frac{\sum_{i=1}^{3} P(C = k_2 \mid x_i)(x_{i,n} - \mu_{k_2,n})(x_{i,m} - \mu_{k_2,m})^T}{\sum_{i=1}^{3} P(C = k_2 \mid x_i)} \\
&= \begin{bmatrix} 0.998818 & -0.215305 \\ -0.215305 & 0.467476 \end{bmatrix}
\end{aligned}$$

$$\pi_2 = P(C = k_2) = \frac{\sum_{i=1}^{3} P(C = k_2 \mid x_i)}{\sum_{c=1}^{2} \sum_{i=1}^{3} P(C = c \mid x_i)} = 0.582811$$

2. **Given the updated parameters computed in previous question:**

(a) **Perform a hard assignment of observations to clusters under a MAP assumption.**

Just like in the first question's answer, we'll need to compute the posterior probabilities of each sample belonging to each cluster (now utilizing the newly updated parameters); however, instead of proceeding to the **M-Step**, we'll just assign each sample to the cluster with the highest posterior probability. Note that, since all calculations follow the same formulas utilized in the previous question's **E-Step**, we're not going to repeat them here, opting instead to just write the final results.

The priors have been updated in the previous question's answer to:

$$\pi_1 = 0.417189, \quad \pi_2 = 0.582811$$

Moreover, we've also updated the means and covariances of the distributions to:

$$\mu_1 = \begin{bmatrix} 0.750964 \\ 1.31149 \end{bmatrix} \qquad \Sigma_1 = \begin{bmatrix} 0.436053 & 0.0775726 \\ 0.0775726 & 0.778455 \end{bmatrix}$$
$$\mu_2 = \begin{bmatrix} 0.0343846 \\ 0.777028 \end{bmatrix} \qquad \Sigma_2 = \begin{bmatrix} 0.998818 & -0.215305 \\ -0.215305 & 0.467476 \end{bmatrix}$$

Therefore, for each sample, we'll have:

x_1:	x_2:	x_3:
$P(x_1 \mid C = k_1) = 0.1957$	$P(x_2 \mid C = k_1) = 0.0081953$	$P(x_3 \mid C = k_1) = 0.077166$
$P(x_1 \mid C = k_2) = 0.01352$	$P(x_2 \mid C = k_2) = 0.14365$	$P(x_3 \mid C = k_2) = 0.10478$
$P(C = k_1 \mid x_1) = \underline{0.91198}$	$P(C = k_1 \mid x_2) = 0.039237$	$P(C = k_1 \mid x_3) = 0.34519$
$P(C = k_2 \mid x_1) = 0.088017$	$P(C = k_2 \mid x_2) = \underline{0.96076}$	$P(C = k_2 \mid x_3) = \underline{0.65481}$

After performing these calculations, under a MAP (Maximum A Posteriori) assumption, we'll assign each sample to the cluster with the highest posterior probability:

$$\begin{aligned} \text{MAP}(x_1) &\mapsto k_1 \\ \text{MAP}(x_2) &\mapsto k_2 \\ \text{MAP}(x_3) &\mapsto k_2 \end{aligned}$$

(b) **Compute the silhouette of the larger cluster using the Euclidean distance.**

As we know, the silhouette of a given sample x_i is defined as:

$$s_i = \frac{b_i - a_i}{\max(a_i, b_i)}$$

where a_i is the average distance between x_i and all other samples in the same cluster, and b_i is the average distance between x_i and all other samples in its **neighboring cluster** - the neighboring cluster being, therefore, the cluster minimizing such average distance.

Moreover, the silhouette of a given cluster k_n with m assigned samples, is defined as:

$$s(k_n) = \frac{\sum_{i=1}^{m} s_i}{m}$$

Here, the **largest cluster** will be the cluster with the biggest associated prior value (π_k). As was computed in 1., $\pi_2 = 0.582811 > 0.417189 = \pi_1$, hence the larger cluster will be k_2. Its assigned samples, considering a MAP assumption, are x_2 and x_3, so we'll have the following:

x_2:

$$\begin{aligned} a_2 &= \|x_2 - x_3\|_2 \\ &= \left\| \begin{bmatrix} -1 \\ 1 \end{bmatrix} - \begin{bmatrix} 1 \\ 0 \end{bmatrix} \right\|_2 = 2.2361 \\ b_2 &= \|x_2 - x_1\|_2 \\ &= \left\| \begin{bmatrix} -1 \\ 1 \end{bmatrix} - \begin{bmatrix} 1 \\ 2 \end{bmatrix} \right\|_2 = 2.2361 \\ s_2 &= \frac{b_2 - a_2}{\max(a_2, b_2)} \\ &= \frac{2.2361 - 2.2361}{\max(2.2361, 2.2361)} = 0 \end{aligned}$$

x_3:

$$\begin{aligned} a_3 &= \|x_3 - x_2\|_2 \\ &= \left\| \begin{bmatrix} 1 \\ 0 \end{bmatrix} - \begin{bmatrix} -1 \\ 1 \end{bmatrix} \right\|_2 = 2.2361 \\ b_3 &= \|x_3 - x_1\|_2 \\ &= \left\| \begin{bmatrix} 1 \\ 0 \end{bmatrix} - \begin{bmatrix} 1 \\ 2 \end{bmatrix} \right\|_2 = 2 \\ s_3 &= \frac{b_3 - a_3}{\max(a_3, b_3)} \\ &= \frac{2 - 2.2361}{\max(2.2361, 2)} = -0.11803 \end{aligned}$$

With this, we can compute the silhouette of the larger cluster:

$$s(k_2) = \frac{s_2 + s_3}{2} = -0.059015$$

Part II: Programming and critical analysis

The code utilized to answer the following questions is available in this report's appendix.

Recall the `pd_speech.arff` dataset from earlier homeworks, centered on the Parkinson diagnosis from speech features. For the following exercises, normalize the data using `sklearn`'s `MinMaxScaler`.

3. **Using `sklearn`, apply k-means clustering fully unsupervisedly (without targets) on the normalized data with $k = 3$ and three different seeds (using random $\in \{0, 1, 2\}$). Assess the silhouette and purity of the produced solutions.**

   `sklearn.metrics` offers us the `silhouette_score` method, which computes the silhouette of a given clustering solution. `purity_score` is a custom method defined in the appendix of this report, which computes the purity of a given clustering solution.

   Regarding the k-means clustering solutions for $k = 3$, for each of the given seeds, we were able to gather the following scores:

`random_state`	**Silhouette score**	**Purity score**
0	0.11362028	0.76719577
1	0.11403554	0.76322751
2	0.11362028	0.76719577

   Table 1: Silhouette and Purity scores for each clustering solution

4. **What is causing the non-determinism?**

   The non-determinism present in the k-means clustering solutions gathered in the previous exercise is caused by the fact that the algorithm is inherently random: `sklearn`'s `KMeans` class sets up the centroids' initial positions in a randomly generated fashion, thus leading to possible different convergence points for the same data and number of clusters. `random_state`, here, works as a mere manner of controlling the random seed used to generate the initial centroid positions: for the same seed, the same initial centroids' positions will be generated, thus leading to the same convergence point. For different seeds, different initial centroid positions will be generated, which could lead to possible different convergence points.

5. **Using a scatter plot, visualize side-by-side the labeled data using as labels: i) the original Parkinson diagnoses, and ii) the previously learned $k = 3$ clusters (`random = 0`). To this end, select the two most informative features as axes and color observations according to their label. For feature selection, select the two input variables with highest variance on the MinMax normalized data.**

   In order to select the two most informative features, we'll use the custom method `select_most_informative_features` defined in the appendix of this report. After gathering that the two input variables presenting the highest variance are both `tqwt_entropy_shannon_dec_16` and `tqwt_kurtosisValue_dec_34`, we were able to gather the scatter plots present in Figure 1.

6. **The fraction of variance explained by a principal component is the ratio between the variance of that component (i.e., its eigenvalue) and total variance (i.e., sum of all eigenvalues). How many principal components are necessary to explain more than 80% of variability?**

   The `PCA` class from `sklearn.decomposition` can be used to compute the **principal components** of a dataset. Moreover, its `n_components` parameter, if set to a value between 0 and 1, will automatically select the number of components necessary to explain a fraction of variability greater than the given value - in our case, 0.8 - which can be consulted in the `n_components_` attribute.

   After running the `calculate_pca` method, present in the latter section of this report's appendix, we were able to gather that the number of principal components necessary to explain 80% of variability is 31.

Appendix

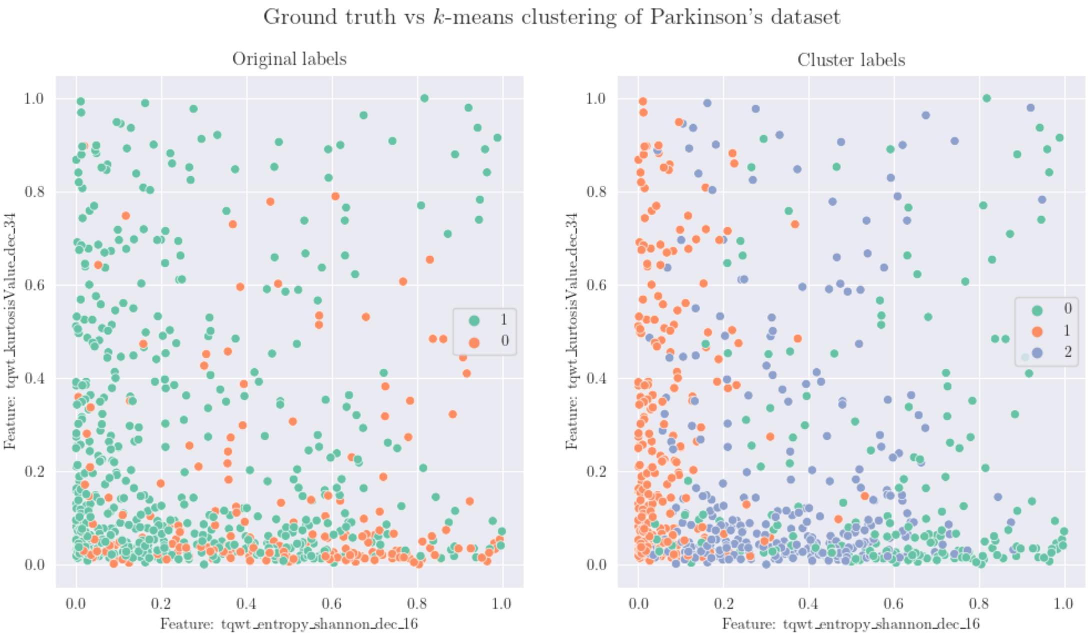

Figure 1: Ground truth vs k-means clustering of Parkinson's dataset