Homework 4 - Exs 1,2

21 de outubro de 2023 16:13

Given the following observations, $\left\{\begin{pmatrix}1\\0.6\\0.1\end{pmatrix}, \begin{pmatrix}0\\-0.4\\0.8\end{pmatrix}, \begin{pmatrix}0\\0.2\\0.5\end{pmatrix}, \begin{pmatrix}1\\0.4\\-0.1\end{pmatrix}\right\}$.

Consider a Bayesian clustering that assumes $\{y_1\} \perp\!\!\!\perp \{y_2, y_3\}$, two clusters following a Bernoulli distribution on y_1 (p_1 and p_2), a multivariate Gaussian on $\{y_2, y_3\}$ (N_1 and N_2), and the following initial mixture:

$$\pi_1 = 0.5, \pi_2 = 0.5$$

$$p_1 = P(y_1 = 1) = 0.3, \quad p_2 = P(y_1 = 1) = 0.7$$

$$N_1\left(\mathbf{u}_1 = \begin{pmatrix}1\\1\end{pmatrix}, \mathbf{\Sigma}_1 = \begin{pmatrix}2 & 0.5\\0.5 & 2\end{pmatrix}\right), \quad N_2\left(\mathbf{u}_2 = \begin{pmatrix}0\\0\end{pmatrix}, \mathbf{\Sigma}_2 = \begin{pmatrix}1.5 & 1\\1 & 1.5\end{pmatrix}\right).$$

Note: you can solve this exercise by neglecting y_1 and still scoring up to 70% of its grade.

1) [6v] Perform one epoch of the EM clustering algorithm and determine the new parameters.
Hint: we suggest you to use numpy and scipy, however disclose the intermediary results step by step.

Bernoulli:

$$p = p(X=1) \quad \sigma^2 = p(1-p)$$

$$\gamma_{ik} = p(C_k \mid x_i) = \frac{p(x_i \mid C_k)P(C_k)}{p(x_i)} \overset{1^a\,iter}{=} \frac{P(y_1 \mid C_k)\, p(y_2, y_3 \mid C_k)}{k}$$

$$\mu_k = \frac{1}{N_k}\sum_i \gamma_{ik} x_i \qquad N_k = \sum_i \gamma_{ik}$$

$$\Sigma_k = \frac{1}{N_k}\sum_i \gamma_{ik}(x_i - \mu_k)(x_i - \mu_k)^T$$

atualizar priors: $p(C_k) = \frac{N_k}{N}$

posteriors

X_1:

$$p(C_1 \mid x_1) = P(x_1 \mid C_1)P(C_1) = P(y_1 = 1 \mid C_1)P(y_2 = 0.6, y_3 = 0.1 \mid C_1) \times \frac{1}{2}$$

$$= 0.3 \times 0.06658 \times \frac{1}{2} = 0.00999$$

$$p(C_2 \mid x_1) = P(x_1 \mid C_2)P(C_2) = P(y_1 = 1 \mid C_2)P(y_2 = 0.6, y_3 = 0.1 \mid C_2) \times \frac{1}{2}$$

$$= 0.7 \times 0.11962 \times \frac{1}{2} = 0.04187$$

Normalização:

$$p(C_1 \mid x_1) = \frac{p(C_1 \mid x_1)}{p(C_1 \mid x_1) + p(C_2 \mid x_1)} = \frac{0.00999}{0.00999 + 0.04187} = 0.19263$$

$$p(C_2 \mid x_1) = \frac{0.04187}{0.00999 + 0.04187} = 0.80737$$

X_2:

$$p(C_1 \mid x_2) = P(x_2 \mid C_1)P(C_1) = P(y_1 = 0 \mid C_1)P(y_2 = -0.4, y_3 = 0.8 \mid C_1) \times \frac{1}{2}$$

$$= 0.7 \times 0.05005 \times \frac{1}{2} = 0.01752$$

$$p(C_2 \mid x_2) = P(x_2 \mid C_2)P(C_2) = P(y_1 = 0 \mid C_2)P(y_2 = -0.4, y_3 = 0.8 \mid C_2) \times \frac{1}{2}$$

$$= 0.3 \times 0.06819 \times \frac{1}{2} = 0.01023$$

Normalização:

$$p(C_1 \mid x_2) = \frac{0.01752}{0.01752 + 0.01023} = 0.63135$$

$$p(C_2 \mid x_2) = \frac{0.01023}{0.01752 + 0.01023} = 0.36865$$

X_3:

$$p(C_1 \mid x_3) = P(x_3 \mid C_1)P(C_1) = P(y_1 = 0 \mid C_1)P(y_2 = 0.2, y_3 = 0.5 \mid C_1) \times \frac{1}{2}$$

$$= 0.7 \times 0.06837 \times \frac{1}{2} = 0.02393$$

$$p(C_2 \mid x_3) = P(x_3 \mid C_2)P(C_2) = P(y_1 = 0 \mid C_2)P(y_2 = 0.2, y_3 = 0.5 \mid C_2) \times \frac{1}{2}$$

$$= 0.3 \times 0.12958 \times \frac{1}{2} = 0.01944$$

Normalização:

$$p(C_1 \mid x_3) = \frac{0.02393}{0.02393 + 0.01944} = 0.55176$$

$$p(C_2 \mid x_3) = \frac{0.01944}{0.02393 + 0.01944} = 0.44824$$

X_4:

$$p(C_1 \mid x_4) = P(x_4 \mid C_1)P(C_1) = P(y_1 = 1 \mid C_1)P(y_2 = 0.4, y_3 = -0.1 \mid C_1) \times \frac{1}{2}$$

$$= 0.3 \times 0.05905 \times \frac{1}{2} = 0.00886$$

$$p(C_2 \mid x_4) = P(x_4 \mid C_2)P(C_2) = P(y_1 = 1 \mid C_2)P(y_2 = 0.4, y_3 = -0.1 \mid C_2) \times \frac{1}{2}$$

$$= 0.7 \times 0.1245 \times \frac{1}{2} = 0.04358$$

Normalização:

$$p(C_1 \mid x_4) = \frac{0.00886}{0.00886 + 0.04358} = 0.16895$$

$$p(C_2 \mid x_4) = \frac{0.00886}{0.00886 + 0.04358} = 0.83105$$

updates

$$N_k = \sum_i \gamma_{ik}$$

$$N_1 = 0.19263 + 0.63135 + 0.55176 + 0.16895 = 1.54469$$

$$N_2 = 0.80737 + 0.36865 + 0.44824 + 0.83105 = 2.45531$$

$$p(C_1) = \frac{1.54469}{1.54469 + 2.45531} = 0.38617$$

$$p(C_2) = \frac{2.45531}{1.54469 + 2.45531} = 0.61383$$

Bernoulli:

$$p_1 = P(y_1 = 1 \mid C_1) = \frac{1 \times 0.19263 + 0 \times 63135 + 0 \times 0.55176 + 1 \times 0.16895}{1.54469} = 0.23408$$

$$p_2 = P(y_1 = 1 \mid C_2) = \frac{1 \times 0.80737 + 0 \times 0.36865 + 0 \times 0.44824 + 1 \times 0.83105}{2.45531} = 0.66730$$

Normal:

$$\mu_k = \frac{1}{N_k}\sum_i \gamma_{ik} x_i$$

$$\mu_1 = \frac{1}{1.54469} \times \left(0.19263\begin{pmatrix}0.6\\0.1\end{pmatrix} + 0.63135\begin{pmatrix}-0.4\\0.8\end{pmatrix} + 0.55176\begin{pmatrix}0.2\\0.5\end{pmatrix} + 0.16895\begin{pmatrix}0.4\\-0.1\end{pmatrix}\right)$$

$$= \frac{1}{1.54469} \times \left[\begin{pmatrix}0.115578\\0.019263\end{pmatrix} + \begin{pmatrix}-0.25254\\0.50508\end{pmatrix} + \begin{pmatrix}0.110352\\0.27588\end{pmatrix} + \begin{pmatrix}0.06758\\-0.016895\end{pmatrix}\right]$$

$$= \begin{pmatrix}0.02652\\0.50711\end{pmatrix}$$

$$\mu_2 = \frac{1}{2.45531} \times \left[0.80737\begin{pmatrix}0.6\\0.1\end{pmatrix} + 0.36865\begin{pmatrix}-0.4\\0.8\end{pmatrix} + 0.44824\begin{pmatrix}0.2\\0.5\end{pmatrix} + 0.83105\begin{pmatrix}0.4\\-0.1\end{pmatrix}\right]$$

$$= \begin{pmatrix}0.30914\\0.21043\end{pmatrix}$$

$$\Sigma_k = \frac{1}{N_k}\sum_i \gamma_{ik}(x_i - \mu_k)(x_i - \mu_k)^T$$

$$\Sigma_1 = \frac{1}{1.54469} \times \Bigg(0.19263\left[\begin{pmatrix}0.6\\0.1\end{pmatrix} - \begin{pmatrix}0.02652\\0.50711\end{pmatrix}\right]\left[\begin{pmatrix}0.6\\0.1\end{pmatrix} - \begin{pmatrix}0.02652\\0.50711\end{pmatrix}\right]^T$$

$$+ 0.63135\left[\begin{pmatrix}-0.4\\0.8\end{pmatrix} - \begin{pmatrix}0.02652\\0.50711\end{pmatrix}\right]\left[\begin{pmatrix}-0.4\\0.8\end{pmatrix} - \begin{pmatrix}0.02652\\0.50711\end{pmatrix}\right]^T$$

$$+ 0.55176\left[\begin{pmatrix}0.2\\0.5\end{pmatrix} - \begin{pmatrix}0.02652\\0.50711\end{pmatrix}\right]\left[\begin{pmatrix}0.2\\0.5\end{pmatrix} - \begin{pmatrix}0.02652\\0.50711\end{pmatrix}\right]^T$$

$$+ 0.16895\left[\begin{pmatrix}0.4\\-0.1\end{pmatrix} - \begin{pmatrix}0.02652\\0.50711\end{pmatrix}\right]\left[\begin{pmatrix}0.4\\-0.1\end{pmatrix} - \begin{pmatrix}0.02652\\0.50711\end{pmatrix}\right]^T\Bigg)$$

$$= \frac{1}{1.54469} \times \Bigg(0.19263\begin{bmatrix}0.57348\\-0.40711\end{bmatrix}\begin{bmatrix}0.57348\\-0.40711\end{bmatrix}^T + 0.63135\begin{bmatrix}-0.42652\\0.29289\end{bmatrix}\begin{bmatrix}-0.42652\\0.29289\end{bmatrix}^T$$

$$+ 0.55176\begin{bmatrix}0.17348\\-0.00711\end{bmatrix}\begin{bmatrix}0.17348\\-0.00711\end{bmatrix}^T + 0.16895\begin{bmatrix}0.37348\\-0.60711\end{bmatrix}\begin{bmatrix}0.37348\\-0.60711\end{bmatrix}^T\Bigg)$$

$$= \frac{1}{1.54469} \times \Bigg(\begin{bmatrix}0.06335202 & -0.04497322\\-0.04497322 & 0.03192622\end{bmatrix} + \begin{bmatrix}0.11485476 & -0.07887042\\-0.07887042 & 0.05416008\end{bmatrix}$$

$$+ \begin{bmatrix}0.0166053885 & -6.80564399 \times 10^{-4}\\-6.80564399 \times 10^{-4} & 2.78926267 \times 10^{-5}\end{bmatrix} + \begin{bmatrix}0.02356638 & -0.0383083\\-0.0383083 & 0.06227202\end{bmatrix}\Bigg)$$

$$= \begin{bmatrix}0.14137 & -0.10541\\-0.10541 & 0.09606\end{bmatrix}$$

$$\Sigma_2 = \frac{1}{2.45531} \times \Bigg(0.80737\left[\begin{pmatrix}0.6\\0.1\end{pmatrix} - \begin{pmatrix}0.30914\\0.21043\end{pmatrix}\right]\left[\begin{pmatrix}0.6\\0.1\end{pmatrix} - \begin{pmatrix}0.30914\\0.21043\end{pmatrix}\right]^T$$

$$+ 0.36865\left[\begin{pmatrix}-0.4\\0.8\end{pmatrix} - \begin{pmatrix}0.30914\\0.21043\end{pmatrix}\right]\left[\begin{pmatrix}-0.4\\0.8\end{pmatrix} - \begin{pmatrix}0.30914\\0.21043\end{pmatrix}\right]^T$$

$$+ 0.44824\left[\begin{pmatrix}0.2\\0.5\end{pmatrix} - \begin{pmatrix}0.30914\\0.21043\end{pmatrix}\right]\left[\begin{pmatrix}0.2\\0.5\end{pmatrix} - \begin{pmatrix}0.30914\\0.21043\end{pmatrix}\right]^T$$

$$+ 0.83105\left[\begin{pmatrix}0.4\\-0.1\end{pmatrix} - \begin{pmatrix}0.30914\\0.21043\end{pmatrix}\right]\left[\begin{pmatrix}0.4\\-0.1\end{pmatrix} - \begin{pmatrix}0.30914\\0.21043\end{pmatrix}\right]^T\Bigg)$$

$$= \frac{1}{2.45531} \times \Bigg(0.80737\begin{bmatrix}0.29086\\-0.11043\end{bmatrix}\begin{bmatrix}0.29086\\-0.11043\end{bmatrix}^T + 0.36865\begin{bmatrix}-0.70914\\0.58957\end{bmatrix}\begin{bmatrix}-0.70914\\0.58957\end{bmatrix}^T$$

$$+ 0.44824\begin{bmatrix}-0.10914\\0.28957\end{bmatrix}\begin{bmatrix}-0.10914\\0.28957\end{bmatrix}^T + 0.83105\begin{bmatrix}0.09086\\-0.31043\end{bmatrix}\begin{bmatrix}0.09086\\-0.31043\end{bmatrix}^T\Bigg)$$

$$= \begin{bmatrix}0.10829 & -0.08865\\-0.08865 & 0.10412\end{bmatrix}$$

2) [2v] Given the new observation, $\mathbf{x}_{new} = \begin{pmatrix}1\\0.3\\0.7\end{pmatrix}$, determine the cluster memberships (posteriors).

$$P(C_1 \mid x_{new}) = P(x_{new} \mid C_1)P(C_1) = P(y_1 = 1, y_2 = 0.3, y_3 = 0.7 \mid C_1)P(C_1)$$

$$= P(y_1 = 1 \mid C_1)P(y_2 = 0.3, y_3 = 0.7 \mid C_1)P(C_1)$$

$$= 0.23408 \times 0.02707 \times 0.38617$$

$$= 0.00245$$

$$P(C_2 \mid x_{new}) = P(x_{new} \mid C_2)P(C_2) = P(y_1 = 1, y_2 = 0.3, y_3 = 0.7 \mid C_2)P(C_2)$$

$$= P(y_1 = 1 \mid C_2)P(y_2 = 0.3, y_3 = 0.7 \mid C_2)P(C_2)$$

$$= 0.66730 \times 0.06844 \times 0.61383$$

$$= 0.02803$$

Normalização:

$$P(C_1 \mid x_{new}) = \frac{0.00245}{0.00245 + 0.02803} = 0.08038$$

$$P(C_2 \mid x_{new}) = \frac{0.02803}{0.00245 + 0.02803} = 0.91962$$

Homework 4 - Exs 3,4

30 de outubro de 2023 21:16

3) [2.5v] Performing a hard assignment of observations to clusters under a ML assumption, identify the silhouette of the larger cluster under a Manhattan distance.

$$P(C_1|X_1) = P(X_1|C_1) = P(y_1=1, y_2=0.6, y_3=0.1|C_1)$$
$$= P(y_1=1|C_1)\,P(y_2=0.6, y_3=0.1|C_1)$$
$$= 0.23408 \times 0.98907$$
$$= 0.23152$$

$$P(C_1|X_2) = P(X_2|C_1) = P(y_1=0, y_2=-0.4, y_3=0.8|C_1)$$
$$= 1.26620$$

$$P(C_1|X_3) = P(X_3|C_1) = P(y_1=0, y_2=0.2, y_3=0.5|C_1)$$
$$= 1.43787$$

$$P(C_1|X_4) = P(X_4|C_1) = P(y_1=1, y_2=0.4, y_3=-0.1|C_1)$$
$$= 0.02078$$

$$P(C_2|X_1) = P(X_1|C_2) = P(y_1=1, y_2=0.6, y_3=0.1|C_2)$$
$$= P(y_1=1|C_2)\,P(y_2=0.6, y_3=0.1|C_2)$$
$$= 0.66730 \times 1.42292$$
$$= 0.94951$$

$$P(C_2|X_2) = P(X_2|C_2) = P(y_1=0, y_2=-0.4, y_3=0.8|C_2)$$
$$= 0.08874$$

$$P(C_2|X_3) = P(X_3|C_2) = P(y_1=0, y_2=0.2, y_3=0.5|C_2)$$
$$= 0.45422$$

$$P(C_2|X_4) = P(X_4|C_2) = P(y_1=1, y_2=0.4, y_3=-0.1|C_2)$$
$$= 0.72327$$

Para realizar os hard assignments apenas precisamos de saber o valor maior entre os 2 clusters, pelo que não é necessário normalizar.

Hard assignments:

$X_1: C_2 \quad X_2: C_1 \quad X_3: C_1 \quad X_4: C_2$

silhouette:

$$s(x) = \begin{cases} 1 - \frac{a(x)}{b(x)}, & a(x) \leq b(x) \\ \frac{b(x)}{a(x)} - 1, & a(x) > b(x) \end{cases}$$

$a(x)$ → distância média aos pontos do próprio cluster

$b(x)$ → distância média aos pontos do cluster mais próximo

distâncias de Manhattan

	x_1	x_2	x_3	x_4	
x_1	0	2.7	1.8	0.4	C_2
x_2	2.7	0	0.9	2.7	C_1
x_3	1.8	0.9	0	1.8	C_1
x_4	0.4	2.7	1.8	0	C_2

X_1: $a(x_1) = 0.4 \quad b(x_1) = \frac{2.7+1.8}{2} = 2.25 \quad s(x_1) = 1 - \frac{0.4}{2.25} = 0.82(2)$

X_2: $a(x_2) = 0.9 \quad b(x_2) = \frac{2.7+2.7}{2} = 2.7 \quad s(x_2) = 1 - \frac{0.9}{2.7} = 0.6(6)$

X_3: $a(x_3) = 0.9 \quad b(x_3) = \frac{1.8+1.8}{2} = 1.8 \quad s(x_3) = 1 - \frac{0.9}{1.8} = 0.5$

X_4: $a(x_4) = 0.4 \quad b(x_4) = \frac{2.7+1.8}{2} = 2.25 \quad s(x_4) = 1 - \frac{0.4}{2.25} = 0.82(2)$

$$s(C_1) = \frac{s(x_2)+s(x_3)}{2} = 0.58(3) \qquad s(C_2) = \frac{s(x_1)+s(x_4)}{2} = 0.82(2)$$

4) [0.5v] Knowing the purity of the clustering solution is 0.75, identify the number of possible classes (ground truth).

R: A pureza é uma medida externa que nos permite avaliar a qualidade dos clusters, nomeadamente o quão correctamente as observações estão agrupadas.

Um valor de 0.75 indica que 75% das observações estão correctamente agrupadas no seu cluster.

Como temos 4 observações, 3 delas estão adequadamente agregadas enquanto que a restante está num cluster ao qual não pertence.

Nesta situação existem 2 possibilidades:

- Ou esta observação pertence a um cluster já existente mas diferente daquele onde está agrupado, existindo 2 classes
- Ou pertence a um cluster que não foi criado, visto ter uma classificação diferente das outras observações, existindo 3 classes

I. Pen-and-paper

Valores da pdf das normais originais nos pontos dados

In [ ]:
```
import numpy as np
from scipy import stats

normal1 = ((1, 1), np.array([[2, 0.5], [0.5, 2]]))
normal2 = ((0, 0), np.array([[1.5, 1], [1, 1.5]]))

observations = [(0.6, 0.1),
                (-0.4, 0.8),
                (0.2, 0.5),
                (0.4, -0.1)]

for index, (mean, cov) in enumerate((normal1, normal2)):
    multnorm = stats.multivariate_normal(mean, cov)

    print(f'Normal {index + 1}:')
    for point in observations:
        print(f'\t{point}:', round(multnorm.pdf(point), 5))
```

```
Normal 1:
        (0.6, 0.1): 0.06658
        (-0.4, 0.8): 0.05005
        (0.2, 0.5): 0.06837
        (0.4, -0.1): 0.05905
Normal 2:
        (0.6, 0.1): 0.11962
        (-0.4, 0.8): 0.06819
        (0.2, 0.5): 0.12958
        (0.4, -0.1): 0.1245
```

Cálculo da cov1 atualizada

In [ ]:
```
a = 0.19263
X1 = np.array([[0.57348], [-0.40711]])
b = 0.63135
X2 = np.array([[-0.42652], [0.29289]])
c = 0.55176
X3 = np.array([[0.17348], [-0.00711]])
d = 0.16895
X4 = np.array([[0.37348], [-0.60711]])

cov1 = (1 / 1.54469) * (a * np.multiply(X1, X1.T) + b * np.multiply(X2, X2.T) +
c * np.multiply(X3, X3.T) + d * np.multiply(X4, X4.T))
print(cov1)
```

```
[[ 0.1413737  -0.10541436]
 [-0.10541436  0.09606213]]
```

Cálculo da cov2 atualizada

In [ ]:
```
a = 0.80737
X1 = np.array([[0.29086], [-0.11043]])
b = 0.36865
```

```
X2 = np.array([[-0.70914], [0.58957]])
c = 0.44824
X3 = np.array([[-0.10914], [0.28957]])
d = 0.83105
X4 = np.array([[0.09086], [-0.31043]])

cov2 = (1 / 2.45531) * (a * np.multiply(X1, X1.T) + b * np.multiply(X2, X2.T) +
c * np.multiply(X3, X3.T) + d * np.multiply(X4, X4.T))
print(cov2)
```

```
[[ 0.10829169 -0.08865146]
 [-0.08865146  0.10412399]]
```

Valores da pdf das normais atualizadas nos pontos dados

In [ ]:
```
points = [(0.6, 0.1), (-0.4, 0.8), (0.2, 0.5), (0.4, -0.1)]

for point in points:
    print(f'Point {point}')

    multnorm = stats.multivariate_normal(np.array([0.02652, 0.50711]), cov1)
    print('\tNormal 1:', round(multnorm.pdf(np.array(point)), 5))

    multnorm = stats.multivariate_normal(np.array([0.30914, 0.21043]), cov2)
    print('\tNormal 2:', round(multnorm.pdf(np.array(point)), 5))
```

```
Point (0.6, 0.1)
        Normal 1: 0.98907
        Normal 2: 1.42292
Point (-0.4, 0.8)
        Normal 1: 1.65318
        Normal 2: 0.26673
Point (0.2, 0.5)
        Normal 1: 1.87731
        Normal 2: 1.36526
Point (0.4, -0.1)
        Normal 1: 0.08877
        Normal 2: 1.08387
```

II. Programming and critical analysis

Recall the column_diagnosis.arff dataset from previous homeworks. For the following exercises, normalize the data using sklearn's MinMaxScaler.

1. **[4v] Using sklearn, apply k-means clustering fully unsupervisedly on the normalized data with k ∈{2,3,4,5} (random=0 and remaining parameters as default). Assess the silhouette and purity of the produced solutions.**

In [ ]:
```
import pandas as pd
import numpy as np
from scipy.io.arff import loadarff
from sklearn import metrics, cluster
from sklearn.preprocessing import MinMaxScaler

def purity_score(y_true, y_pred):
```

```
    # compute contingency/confusion matrix
    confusion_matrix = metrics.cluster.contingency_matrix(y_true, y_pred)
    return np.sum(np.amax(confusion_matrix, axis=0)) / np.sum(confusion_matrix)

data = loadarff('column_diagnosis.arff')
df = pd.DataFrame(data[0])
df['class'] = df['class'].str.decode('utf-8')
X = df.drop('class', axis=1)
y = df['class']

X_scaled = MinMaxScaler().fit_transform(X)

for k in (2, 3, 4, 5):
    print("Clustering results with k =", k)

    kmeans_model = cluster.KMeans(n_clusters=k, random_state=0).fit(X_scaled)

    print("\tsilhouette:", round(metrics.silhouette_score(X_scaled,
kmeans_model.labels_, metric='euclidean'), 5))
    print("\tpurity:", round(purity_score(y, kmeans_model.labels_), 5))
```

```
Clustering results with k = 2
        silhouette: 0.36044
        purity: 0.63226
Clustering results with k = 3
        silhouette: 0.29579
        purity: 0.66774
Clustering results with k = 4
        silhouette: 0.27442
        purity: 0.66129
Clustering results with k = 5
        silhouette: 0.23824
        purity: 0.67742
```

A *silhouette* indica o quão coesos e bem separados estão os clusters, tomando valores entre -1 e 1, sendo que 1 indica alta coesão e separação. A pureza reflete o quão correctamente as observações estão agrupadas, tomando valores entre 0 e 1, sendo 1 a pureza máxima.
Podemos observar que à medida que o número de clusters (k) aumenta, a *silhouette* diminui mas que a pureza se mantém essencialmente constante em torno de 66%. Este valor indica-nos que a maioria dos elementos de cada cluster pertence à mesma classe, apesar de não ser uma pureza realmente significativa, já que apenas se costumam considerar valores superiores a 70%.
Dependendo do contexto, o critério de escolha do nº de clusters (k) difere:

- Se priorizarmos cluster bem separados, devemos olhar para o que apresenta melhor *silhouette*, neste caso k = 2;
- Se por outro lado a prioridade for preservar informação, no sentido de ter clusters que refletem as *ground truths*, devemos escolher o que apresenta melhor pureza, que neste caso é k = 5, apesar de não denotar uma grande diferença em relação aos outros modelos.

2. **[2v] Consider the application of PCA after the data normalization:**
   i. Identify the variability explained by the top two principal components.

ii. For each one of these two components, sort the input variables by relevance by inspecting the absolute weights of the linear projection.

In [ ]:
```
from sklearn.decomposition import PCA
from pprint import pprint

pca = PCA(n_components=2, svd_solver='full')
pca.fit(X_scaled)
print("1st component explained variance ratio:",
round(pca.explained_variance_ratio_[0], 5))
print("2nd component explained variance ratio:",
round(pca.explained_variance_ratio_[1], 5))
print()

columns = X.columns.to_list()
for i in range(len(pca.components_)):
    impt_features = {columns[j] : abs(pca.components_[i][j]) for j
in range(len(columns))}
    print(f"Features by importance on {'1st' if i == 0 else '2nd'} component:")
    pprint([(round(value[0], 5), value[1]) for value in
sorted(zip(impt_features.values(), impt_features.keys()), reverse=True)])
```

```
1st component explained variance ratio: 0.56181
2nd component explained variance ratio: 0.20956

Features by importance on 1st component:
[(0.59162, 'pelvic_incidence'),
 (0.51508, 'lumbar_lordosis_angle'),
 (0.46704, 'pelvic_tilt'),
 (0.32569, 'sacral_slope'),
 (0.21693, 'degree_spondylolisthesis'),
 (0.11582, 'pelvic_radius')]
Features by importance on 2nd component:
[(0.67037, 'pelvic_tilt'),
 (0.58107, 'pelvic_radius'),
 (0.4433, 'sacral_slope'),
 (0.10004, 'pelvic_incidence'),
 (0.08005, 'lumbar_lordosis_angle'),
 (0.00458, 'degree_spondylolisthesis')]
```

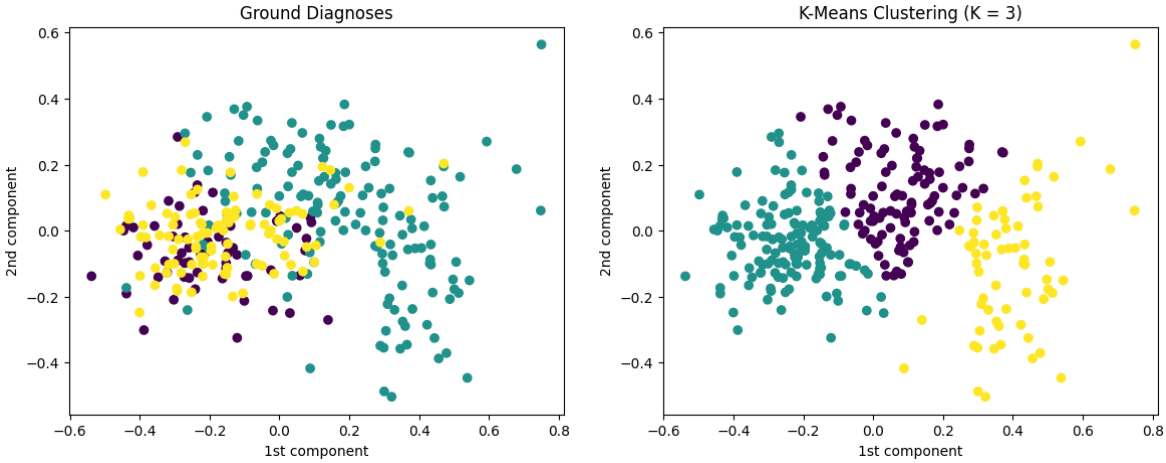
3. **[2v] Visualize side-by-side the data using: i) the ground diagnoses, and ii) the previously learned k =3 clustering solution. To this end, projected the normalized data onto a 2-dimensional data space using PCA and then color observations using the reference and cluster annotations.**

In [ ]:
```
import matplotlib.pyplot as plt

X_pca = pca.transform(X_scaled)
kmeans_model = cluster.KMeans(n_clusters=3, random_state=0).fit(X_pca)

classes = ('Hernia', 'Spondylolisthesis', 'Normal')
plt.figure(figsize=(14, 5))
plt.subplot(121)
plt.scatter(X_pca[:, 0], X_pca[:, 1], c=[classes.index(label) for label in y])
plt.title("Ground Diagnoses")
plt.xlabel("1st component")
```

```
plt.ylabel("2nd component")
plt.subplot(122)
plt.scatter(X_pca[:, 0], X_pca[:, 1], c=kmeans_model.labels_)
plt.title("K-Means Clustering (K = 3)")
plt.xlabel("1st component")
plt.ylabel("2nd component")
plt.show()
```



4. **[1v] Considering the results from questions (1) and (3), identify two ways on how clustering can be used to characterize the population of ill and healthy individuals.**

1. Se a pureza do modelo for alta, é possível essencialmente fazer um mapeamento direto entre cada cluster e um diagnóstico, já que em cada cluster grande parte das observações pertencem à mesma classe. Podemos assim fazer *hard-assign* das observações e, com base no cluster resultante, caracterizá-la, isto é, atribuir um diagnóstico.
2. Caso não se obtenha tal pureza, podemos recorrer a uma redução de dimensionalidade (através do PCA) e com base nas novas features, que à partida contam com menos ruído, concluir um diagnóstico através da separação e distribuição dos pontos no gráfico. Desta forma, é possível obter melhores resultados quando a *silhouette* é alta, já que existe uma grande coesão (baixa distância intra-cluster) e elevada separação (distância inter-cluster).