

practica_2_ejemplo3

September 16, 2024

1 Práctica 2

Aprendizaje Máquina

Implementa al menos tres de los ejemplos propuestos al final de la documentación.

1.1 K-means Clustering

La trama muestra: arriba a la izquierda: Lo que produciría un algoritmo de K-medias usando 8 grupos. arriba a la derecha: Lo que se lograría usando tres clústeres. abajo a la izquierda: Cuál es el efecto de una mala inicialización en el proceso de clasificación: Al establecer `n_init` en solo 1 (el valor predeterminado es 10), se reduce la cantidad de veces que el algoritmo se ejecutará con diferentes semillas de centroides.

abajo a la derecha: La verdad fundamental.

1.1.1 Autores de implementación en la librería:

Gaël Varoquaux, Jaques Grobler

Recuperado de: https://scikit-learn.org/stable/auto_examples/cluster/plot_cluster_iris.html#sphx-gl-auto-examples-cluster-plot-cluster-iris-py

```
[1]: # Code source: Gaël Varoquaux
     # Modified for documentation by Jaques Grobler
     # License: BSD 3 clause

import matplotlib.pyplot as plt

# Though the following import is not directly being used, it is required
# for 3D projection to work with matplotlib < 3.2
import mpl_toolkits.mplot3d # noqa: F401
import numpy as np

from sklearn import datasets
from sklearn.cluster import KMeans

np.random.seed(5)
```

```

iris = datasets.load_iris()
X = iris.data
y = iris.target

estimators = [
    ("k_means_iris_8", KMeans(n_clusters=8)),
    ("k_means_iris_3", KMeans(n_clusters=3)),
    ("k_means_iris_bad_init", KMeans(n_clusters=3, n_init=1, init="random")),
]

fig = plt.figure(figsize=(10, 8))
titles = ["8 clusters", "3 clusters", "3 clusters, bad initialization"]
for idx, ((name, est), title) in enumerate(zip(estimators, titles)):
    ax = fig.add_subplot(2, 2, idx + 1, projection="3d", elev=48, azimuth=134)
    est.fit(X)
    labels = est.labels_

    ax.scatter(X[:, 3], X[:, 0], X[:, 2], c=labels.astype(float), edgecolor="k")

    ax.xaxis.set_ticklabels([])
    ax.yaxis.set_ticklabels([])
    ax.zaxis.set_ticklabels([])
    ax.set_xlabel("Petal width")
    ax.set_ylabel("Sepal length")
    ax.set_zlabel("Petal length")
    ax.set_title(title)

# Plot the ground truth
ax = fig.add_subplot(2, 2, 4, projection="3d", elev=48, azimuth=134)

for name, label in [("Setosa", 0), ("Versicolour", 1), ("Virginica", 2)]:
    ax.text3D(
        X[y == label, 3].mean(),
        X[y == label, 0].mean(),
        X[y == label, 2].mean() + 2,
        name,
        horizontalalignment="center",
        bbox=dict(alpha=0.2, edgecolor="w", facecolor="w"),
    )

ax.scatter(X[:, 3], X[:, 0], X[:, 2], c=y, edgecolor="k")

ax.xaxis.set_ticklabels([])
ax.yaxis.set_ticklabels([])
ax.zaxis.set_ticklabels([])
ax.set_xlabel("Petal width")
ax.set_ylabel("Sepal length")

```

```
ax.set_zlabel("Petal length")
ax.set_title("Ground Truth")

plt.subplots_adjust(wspace=0.25, hspace=0.25)
plt.show()
```

