



# Gaussian Mixture Models (GMM) y Spectral Clustering

# Gaussian Mixture Models (GMM) y Spectral Clustering

Los modelos de mezclas gaussianas son un enfoque probabilístico para el agrupamiento de datos. En GMM, se asume que los datos provienen de una mezcla de varias distribuciones gaussianas y se busca estimar los parámetros de estas distribuciones para modelar la estructura de los datos. GMM es útil cuando los datos no se agrupan claramente en grupos distintos y cuando los grupos pueden tener formas y tamaños diferentes.



## Concepto de clustering espectral y su aplicación en grafos



El clustering espectral es una técnica que se basa en la representación de datos como grafos y en la exploración de las propiedades de sus matrices de afinidad. Se utiliza para encontrar estructuras de grupos en datos que pueden no ser linealmente separables en el espacio de características original. El algoritmo busca los autovectores asociados con los autovalores más grandes de la matriz de afinidad para determinar los grupos.

## Comparación entre GMM y Spectral Clustering

GMM y Spectral Clustering son métodos de agrupamiento diferentes. Vamos a conocerlos a continuación:

GMM es un enfoque probabilístico que modela la distribución de los datos mediante gaussianas.

Puede funcionar bien en datos con distribuciones complejas.

Spectral Clustering se basa en la estructura del grafo de afinidad de los datos.

Es útil para datos no linealmente separables.

### Modelos de Mezclas Gaussianas (GMM)

- Método probabilístico que modela la distribución de los datos como una combinación de distribuciones gaussianas.
- Útil para datos con formas complejas y superpuestas.
- Asigna probabilidades de pertenencia a cada clúster para cada punto de datos.

### Clustering Espectral

- Basado en la teoría de grafos para encontrar estructuras de agrupamiento en datos.
- Utiliza la estructura del grafo y los autovectores de la matriz de afinidad para agrupar los datos.
- Útil para datos no linealmente separables y estructuras de clústeres no convencionales.

## Ejemplos prácticos de aplicación de ambos algoritmos en Python

En Python, GMM se puede implementar utilizando bibliotecas como scikit-learn. Se puede aplicar a conjuntos de datos para modelar la distribución de los datos y realizar el agrupamiento. Por otro lado, Spectral Clustering también está disponible en scikit-learn y se puede usar para agrupar datos representados como grafos. Se pueden proporcionar ejemplos prácticos de aplicación de ambos algoritmos en conjuntos de datos de muestra para ilustrar su uso y comprender sus resultados.



## Introducción a GMM

### Paso 1: Generación de datos

```
import numpy as np
import matplotlib.pyplot as plt
from sklearn.datasets import make_blobs

# Generar datos sintéticos con 3 clústeres
X, _ = make_blobs(n_samples=300, centers=3, cluster_std=1.0,
random_state=42)

# Visualizar los datos
plt.scatter(X[:, 0], X[:, 1], s=50)
plt.title("Datos de ejemplo")
plt.xlabel("Característica 1")
plt.ylabel("Característica 2")
plt.show()
```

## Paso 2: Ajuste del modelo GMM

```
from sklearn.mixture import GaussianMixture

# Crear el modelo GMM con 3 componentes
gmm = GaussianMixture(n_components=3, random_state=42)

# Ajustar el modelo a los datos
gmm.fit(X)

# Obtener las etiquetas de clúster para cada punto de datos
labels = gmm.predict(X)

# Visualizar los resultados
plt.scatter(X[:, 0], X[:, 1], c=labels, s=50, cmap='viridis')
plt.title("Agrupamiento con GMM")
plt.xlabel("Característica 1")
plt.ylabel("Característica 2")
plt.show()
```

# Introducción a Spectral Clustering

## Paso 1: Generación de datos

```
import numpy as np

import matplotlib.pyplot as plt

from sklearn.datasets import make_moons

# Generar datos sintéticos con forma de media luna
X, _ = make_moons(n_samples=300, noise=0.1, random_state=42)
```

```
# Visualizar los datos

plt.scatter(X[:, 0], X[:, 1], s=50)

plt.title("Datos de ejemplo")

plt.xlabel("Característica 1")

plt.ylabel("Característica 2")

plt.show()
```

## Paso 2: Ajuste del modelo Spectral Clustering

```
from sklearn.cluster import SpectralClustering

# Crear el modelo Spectral Clustering con 2 clústeres

spectral = SpectralClustering(n_clusters=2,
                              affinity='nearest_neighbors', random_state=42)

# Ajustar el modelo a los datos

spectral.fit(X)

# Obtener las etiquetas de clúster para cada punto de datos

labels = spectral.labels_

# Visualizar los resultados

plt.scatter(X[:, 0], X[:, 1], c=labels, s=50, cmap='viridis')

plt.title("Agrupamiento con Spectral Clustering")

plt.xlabel("Característica 1")

plt.ylabel("Característica 2")

plt.show()
```

## Ejercicio 3: Comparación entre GMM y Spectral Clustering

```
# Comparación visual de GMM y Spectral Clustering en los mismos datos

plt.figure(figsize=(12, 4))

# GMM

plt.subplot(1, 2, 1)

plt.scatter(X[:, 0], X[:, 1], c=labels_gmm, s=50, cmap='viridis', alpha=0.5)

plt.title("GMM")

plt.xlabel("Característica 1")

plt.ylabel("Característica 2")

# Spectral Clustering

plt.subplot(1, 2, 2)

plt.scatter(X[:, 0], X[:, 1], c=labels_spectral, s=50, cmap='viridis', alpha=0.5)

plt.title("Spectral Clustering")

plt.xlabel("Característica 1")

plt.ylabel("Característica 2")

plt.tight_layout()

plt.show()
```