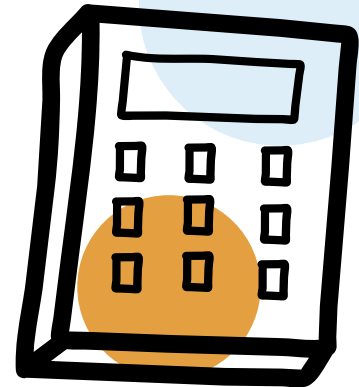




Algoritmo Mean Shift

Algoritmo Mean Shift

Mean Shift es un algoritmo de agrupamiento no paramétrico que busca los modos de densidad en un conjunto de datos para encontrar los centroides de los clusters. Funciona mediante la definición de una función de densidad de probabilidad y moviendo iterativamente los puntos de datos hacia las regiones de mayor densidad. Esto permite que el algoritmo encuentre automáticamente el número de clusters y su forma sin necesidad de especificarlos de antemano.



Concepto de ventana de búsqueda y convergencia hacia los modos de densidad



La ventana de búsqueda es un parámetro clave en el algoritmo Mean Shift que determina el radio de la región alrededor de cada punto de datos dentro de la cual se busca el centroide del cluster. Durante cada iteración del algoritmo, los puntos de datos se desplazan hacia las regiones de mayor densidad, lo que gradualmente conduce a la convergencia hacia los modos de densidad, es decir, los centroides de los clusters.

Implementación de Mean Shift en Python y ejemplos de uso en agrupamiento de datos

Mean Shift se puede implementar fácilmente en Python utilizando bibliotecas como scikit-learn. En la implementación, se ajusta un modelo de Mean Shift a los datos y se utilizan las etiquetas de cluster asignadas para visualizar y analizar los resultados. Ejemplos prácticos de uso de Mean Shift incluyen la segmentación de imágenes, la detección de objetos en imágenes y la agrupación de puntos de datos en conjuntos no lineales.



Implementación

```
#Implementación de Mean Shift en Python
from sklearn.cluster import MeanShift
from sklearn.datasets import make_blobs
import matplotlib.pyplot as plt

# Generar datos de ejemplo
X, _ = make_blobs(n_samples=300, centers=4, cluster_std=0.60,
random_state=0)

# Aplicar Mean Shift
ms = MeanShift()
ms.fit(X)
labels = ms.labels_
cluster_centers = ms.cluster_centers_
```

```
# Visualizar los resultados
```

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

```
plt.scatter(cluster_centers[:,0], cluster_centers[:,1], marker='x',  
color='red', s=300, linewidth=5)
```

```
plt.show()
```

```
#Ejemplo de uso en agrupamiento de datos
```

```
from sklearn.datasets import make_circles
```

```
# Generar datos ficticios en forma de anillo concéntrico
```

```
X, _ = make_circles(n_samples=1000, noise=0.05, random_state=42)
```

```
# Aplicar Mean Shift
```

```
ms = MeanShift()
```

```
ms.fit(X)
```

```
labels = ms.labels_
```

```
cluster_centers = ms.cluster_centers_
```

```
# Visualizar los resultados
```

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

```
plt.scatter(cluster_centers[:,0], cluster_centers[:,1], marker='x',  
color='red', s=300, linewidth=5)
```

```
plt.show()
```