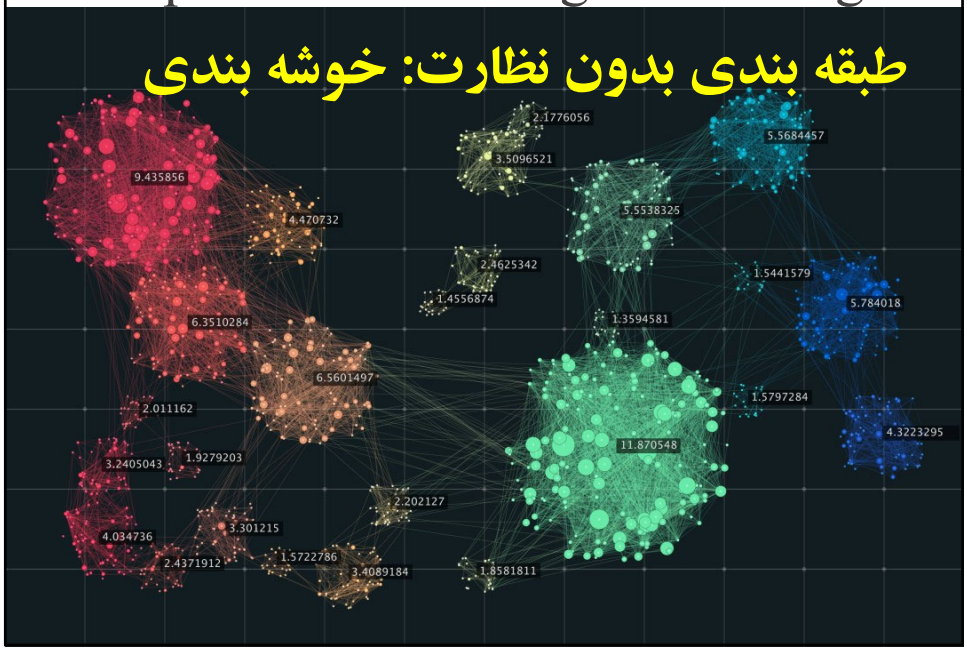
Advanced Application of Artificial Intelligence and Digital Transformation

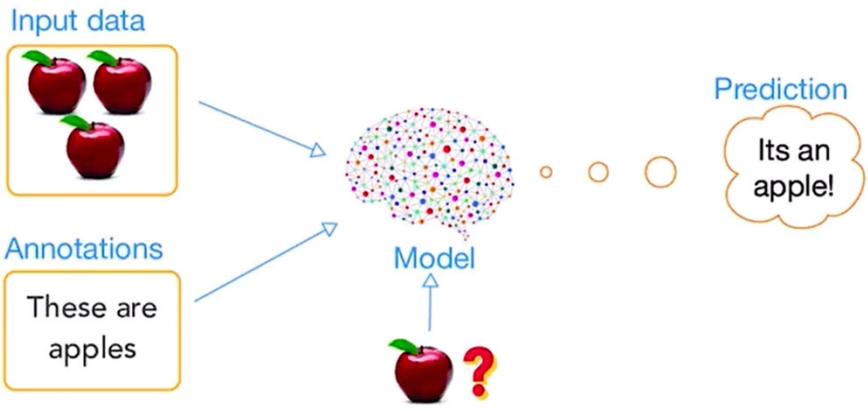
K.N. Toosi University of Technology

Hasan Ghasemzadeh
http://wp.kntu.ac.ir/ghasemzadeh

Unsupervised Learning: Clustering

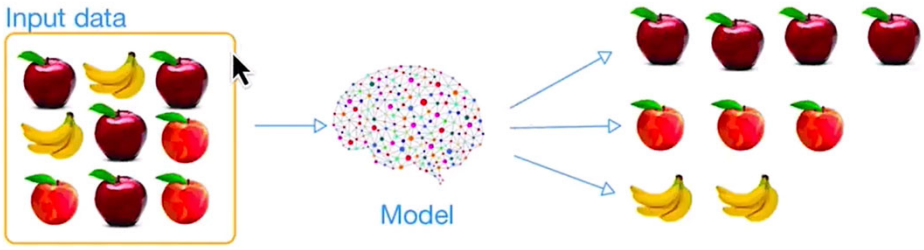


یادگیری نظارت شده



Adv. App. of AI and DT 3

یادگیری بدون نظارت



Adv. App. of AI and DT 4

یادگیری بدون نظارت

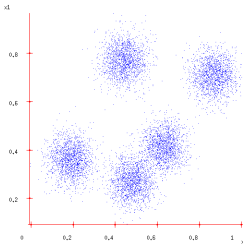
Unsupervised learning: The data have no target attribute.
We want to explore the data to find some intrinsic structures in them.

- ✓ در این قسمت داده‌ها برچسب ندارند.
- ✓ هزینه برچسب گذاری بسیار زیاد است.
- ✓ خوشه‌بندی (clustering)
  - K-means
  - خوشه بندی طیفی
  - خوشه‌بندی سلسله مراتبی
  - خوشه‌بندی مدلهای ترکیبی گوسی
  - DBSCAN
- ✓ کاهش ابعاد
  - تجزیه اصلی مؤلفه‌ها (PCA)
  - t-SNE
  - Autoencoders
- ✓ تشخیص ناهنجاری (Anomaly detection)
- ✓ مدلهای مولد(Generative modelling)

Adv. App. of AI and DT 5

خوشه بندی

- ✓ روشی که شباهت‌های بین گروهی از داده ها را در کل داده ها پیدا می کند خوشه بندی نام دارد.
- ✓ خوشه‌بندی یکی از روشهای بدون نظارت است.
- ✓ بدلیل تاریخچه آن معمولا خوشه بندی مترادف با یادگیری بدون نظارت استفاده می‌شود.
- ✓ روش خوشه بندی بیشترین استفاده را در داده کاوی دارد.



مثال
- افراد با سایز لباس مختلف
- ساختمانهایی که در هنگام زلزله خرابی مشابه دارند
- خوشه‌بندی بندی متن‌های مشابه

الگوریتم‌های معروف:
- خوشه بندی کی میانگین(K-means)
- خوشه بندی طیفی(Spectral clustering)
- خوشه‌بندی سلسله مراتبی (Hierarchical clustering)
- خوشه بندی بر مبنای دانسیته (DBSCAN: Density-based spatial clustering of applications with noise)

Adv. App. of AI and DT 6

خوشه بندی

Clustering: one of the most utilized data mining techniques

- A clustering algorithm
  - Partitional clustering
  - Hierarchical clustering
- A distance (similarity, or dissimilarity) function
- Clustering quality
  - Inter-clusters distance ⇒ maximized
  - Intra-clusters distance ⇒ minimized

کیفیت خوشه‌بندی به الگوریتم، تابع فاصله و نوع مساله بستگی دارد

K-means clustering

- K-means is a partitional clustering algorithm
- Set of data points D

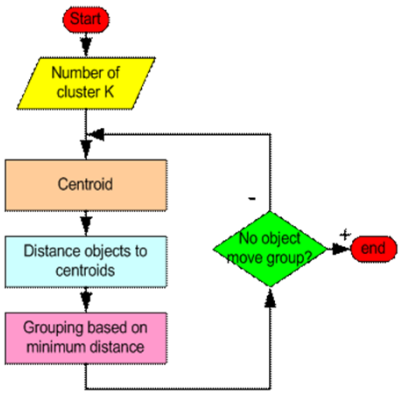
  $D = \{x_1, x_2, ..., x_n\}$ $\quad x_i = (x_{i1}, x_{i2}, ..., x_{ir})$

  n the number of data

  r the number of attributes (dimensions) in the data.
- The k-means partitions the data into k clusters.
  - **Center of** each cluster, called **centroid**.
  - k is specified by the user ($k < n$)

K-means clustering

- the *k-means* algorithm:
  1) Randomly choose *k* data points (seeds) to be the initial centroids, cluster centers
  2) Assign each data point to the closest centroid
  3) Re-compute the centroids using the current cluster memberships.
  4) If a convergence criterion is not met, go to 2).



Stopping/convergence criterion

1. Minimum or no re-assignments of data points to different clusters,
2. minimum or no change of centroids
3. minimum decrease in the ***sum of squared error*** (SSE),

$$SSE = \sum_{j=1}^{k} \sum_{i=1}^{no\ of\ x \in C_j} dist(x, m_j)^2$$

- m_j is the centroid of cluster C_j
- $dist(x, m_j)$ is the distance between data point x and centroid m_j.

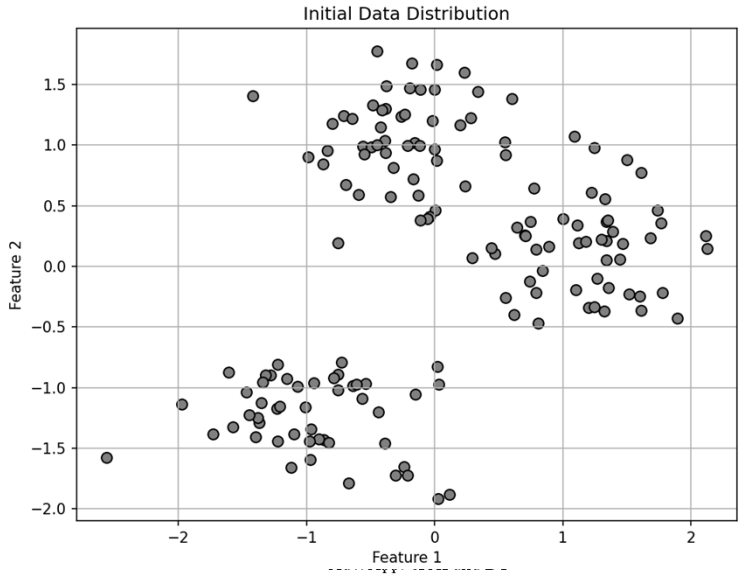
مثال

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

# Generate synthetic 2D data
np.random.seed(42)
X, _ = make_blobs(n_samples=150, centers=3, n_features=2,
cluster_std=2.5)

# Standardize the data
scaler = StandardScaler()
X = scaler.fit_transform(X)

# Plot the initial data distribution
plt.figure(figsize=(8, 6))
plt.scatter(X[:, 0], X[:, 1], c='gray', edgecolor='k', s=50)
plt.title('Initial Data Distribution')
plt.xlabel('Feature 1')
plt.ylabel('Feature 2')
plt.grid(True)
plt.show()
```

12 clustering k means.py

Adv. App. of AI and DT 11

مثال خوشه بندی لغات

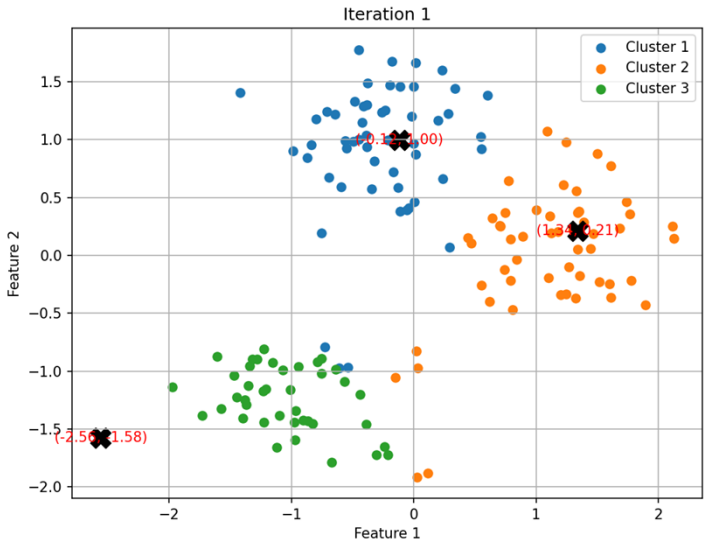


12

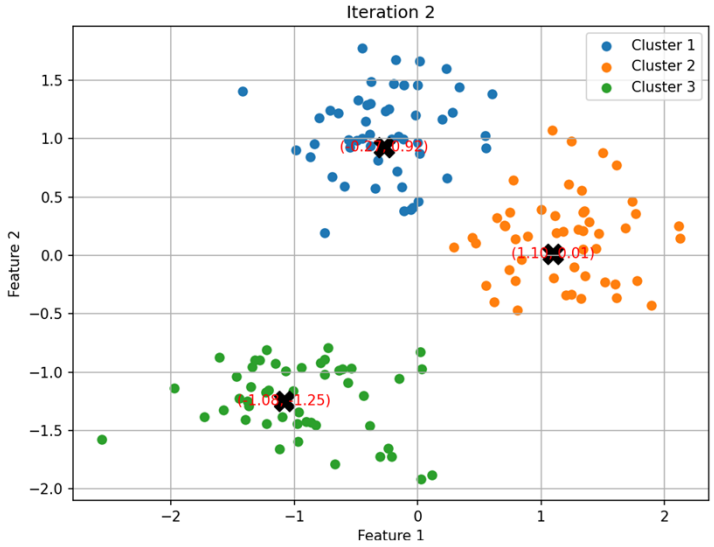
مثال

```
def kmeans_clustering(X, n_clusters, max_iters=10):
    # Randomly initialize centroids
    np.random.seed(42)
    centroids = X[np.random.choice(X.shape[0], n_clusters, replace=False)]

    for i in range(max_iters):
        # Assign clusters based on closest centroid
        distances = np.linalg.norm(X[:, np.newaxis] - centroids, axis=2)
        labels = np.argmin(distances, axis=1)
        # Plot current state
        plt.figure(figsize=(8, 6))
        for j in range(n_clusters):
            plt.scatter(X[labels == j][:, 0], X[labels == j][:, 1], label=f'Cluster {j+1}')
            plt.scatter(centroids[j, 0], centroids[j, 1], s=200, c='black', marker='X')
            # Add centroid coordinates as text
            plt.text(centroids[j, 0], centroids[j, 1], f'({centroids[j, 0]:.2f},
{centroids[j, 1]:.2f})',
                     fontsize=10, color='red', ha='center', va='center')
        plt.title(f'Iteration {i+1}')
        plt.xlabel('Feature 1')
        plt.ylabel('Feature 2')
        plt.legend()
        plt.grid(True)
        plt.show()
        # Update centroids
        new_centroids = np.array([X[labels == j].mean(axis=0) for j in range(n_clusters)])
        # Check for convergence
        if np.all(centroids == new_centroids):
            break
        centroids = new_centroids
# Perform K-means clustering and visualize iterations
kmeans_clustering(X, n_clusters=3, max_iters=10)
```

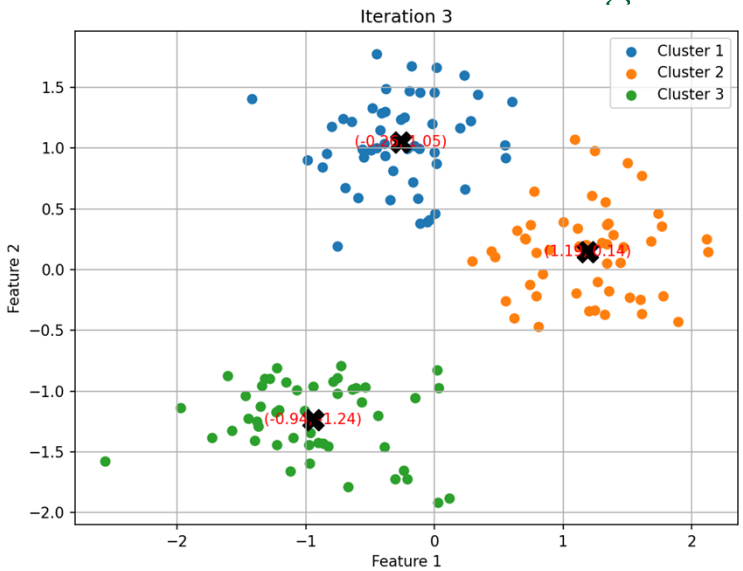

K-means clustering



K-means clustering



K-means clustering



خوشه بندی K-means

مزایا:

- ✓ سریع و مقیاس‌پذیر
- ✓ پیاده‌سازی ساده
- ✓ مناسب برای داده‌های عددی و با توزیع نسبتاً کروی

محدودیت‌ها:

- ✓ نیاز به تعیین تعداد خوشه‌ها از قبل
- ✓ حساس به مقادیر اولیه
- ✓ حساس به داده‌های پرت.

Adv. App. of AI and DT 17

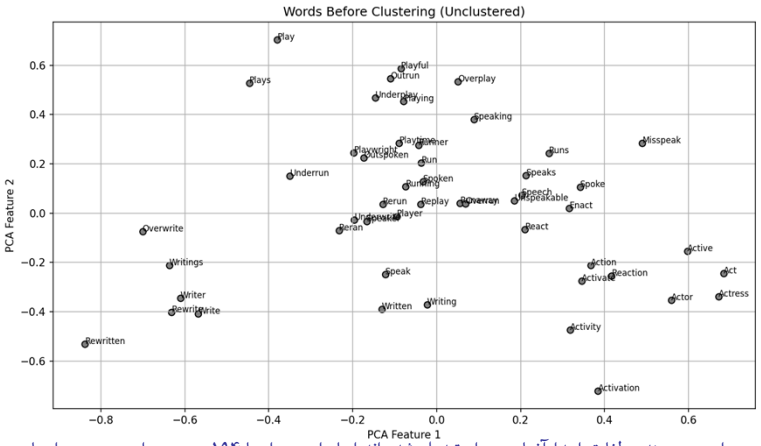
مثال خوشه بندی لغات

```
import pandas as pd
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.cluster import KMeans
import matplotlib.pyplot as plt

# Load the uploaded file
file_path = "family_words.csv"
data = pd.read_csv(file_path)

# Extract the words from the file (correcting the column
name)
words = data['Words_Family']
```

12 Hierarchical Clustering.py

Adv. App. of AI and DT 18

ترسیم به روش کاهش بعد

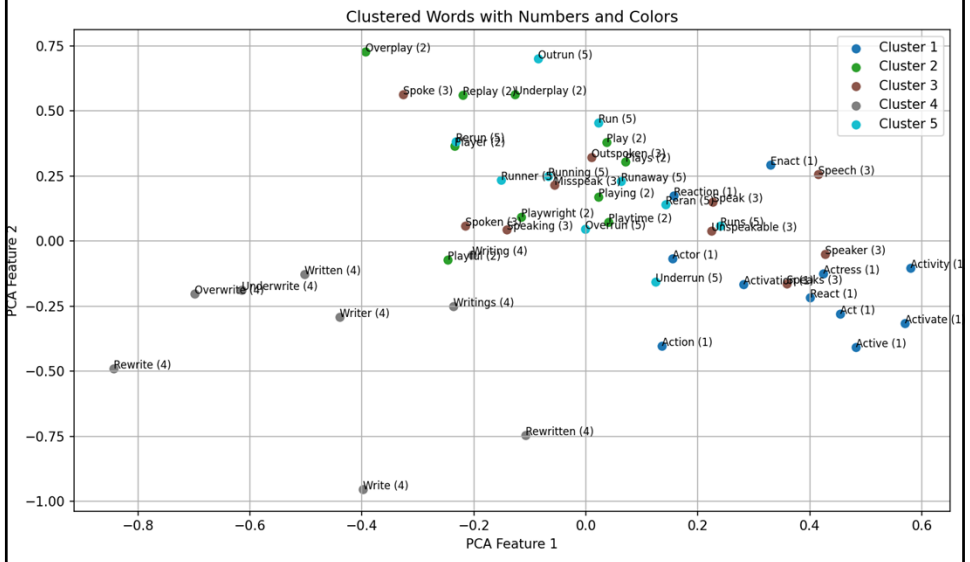


برای دسته بندی لغات ابتدا آنها به بردار تبدیل شده اند ابعاد این بردارها ۱۹۴ بوده و برای رسم بردارهای نقاط داده از یکی از روشهای معروف یادگیری بدون نظارت –کاهش ابعاد استفاده شده است. نام الگوریتم استفاده شده تجزیه اصلی مؤلفه‌ها (PCA: Principal Component Analysis) است.

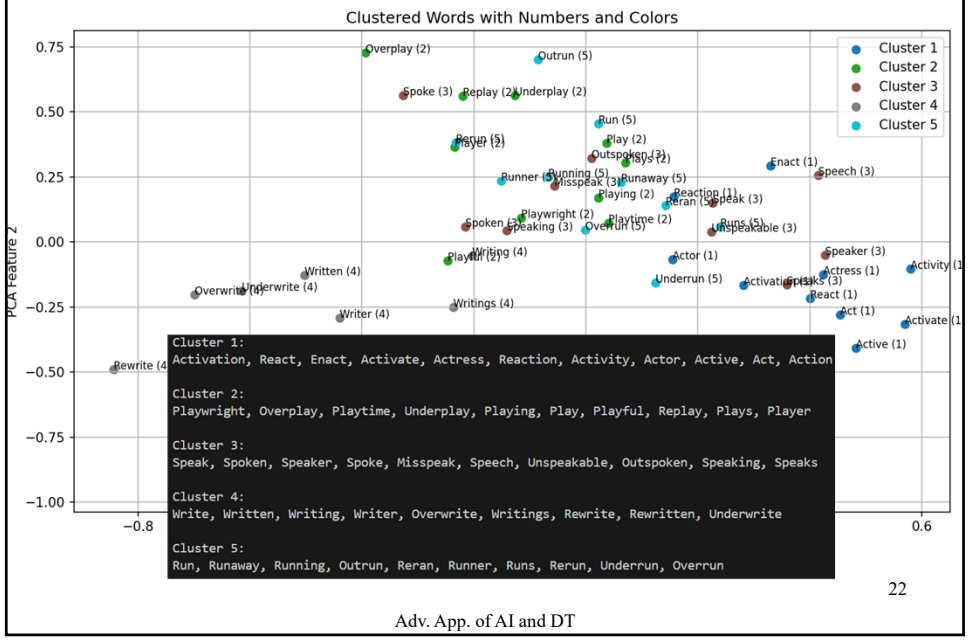
مثال خوشه بندی لغات

```
# Vectorize the words using TF-IDF
vectorizer = TfidfVectorizer(analyzer='char', ngram_range=(2, 3))
X = vectorizer.fit_transform(words)

# Apply K-means clustering
n_clusters = 5  # You can adjust the number of clusters
kmeans = KMeans(n_clusters=n_clusters, random_state=42)
kmeans.fit(X)
# Assign clusters to words
data['Cluster'] = kmeans.labels_
# Visualize the clustering results
for cluster in range(n_clusters):
    cluster_words = data[data['Cluster'] ==
cluster]['Words_Family'].values
    print(f"Cluster {cluster + 1}:")
    print(", ".join(cluster_words))
    print()
# Save the clustered data for further use
clustered_file_path = "family_words_clustered.csv"
data.to_csv(clustered_file_path, index=False)
```

مثال خوشه بندی لغات



مثال خوشه بندی لغات



K-means clustering – Disk version

وقتی داده‌ها خیلی زیاد است نمی توان روی حافظه رایانه تمام داده ها را با بارگذاری نمود در این حالت بایستی از ویرایش دیسک برای خوشه بندی استفاده نمود

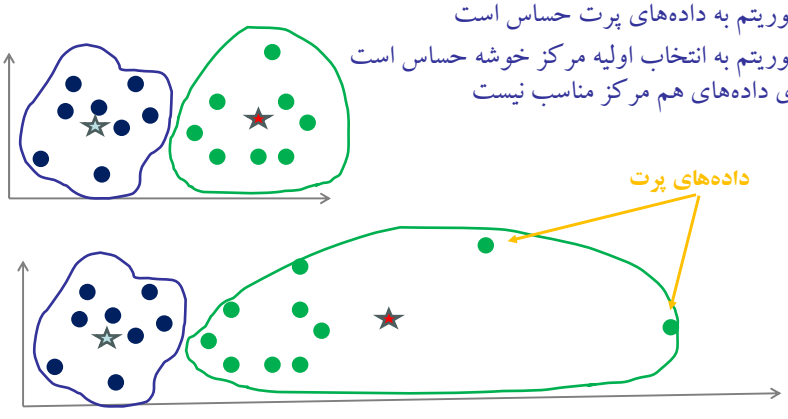
در این روش بسته‌های کوچکی (mini-batches) از داده در هر مرحله انتخاب می‌شود و سپس خوشه‌بندی (با سرعت زیاد و اشغال حافظه کم) بر روی آنها انجام می‌شود.
تعداد تکرارها محدود است و ممکن است کمی خطا ایجاد شود (جواب به نقطه بهینه محلی ختم شود و به نقطه بهینه کلی نرسیم)

We need to limit the number of iterations (< 50 normally).
There are other scale-up algorithms, e.g., BIRCH
(Balanced Iterative Reducing and Clustering using Hierarchies)

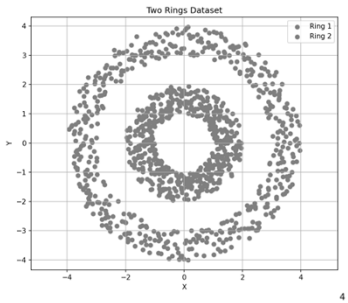
Adv. App. of AI and DT 23

نقاط ضعف K-means

- مقدار k توسط کاربر بایستی تعیین شود
- مرکز خوشه‌ها (سنترویید) تحت تاثیر داده‌های با فراوانی زیاد قرار می‌گیرد
- الگوریتم به داده‌های پرت حساس است
- الگوریتم به انتخاب اولیه مرکز خوشه حساس است
- برای داده‌های هم مرکز مناسب نیست

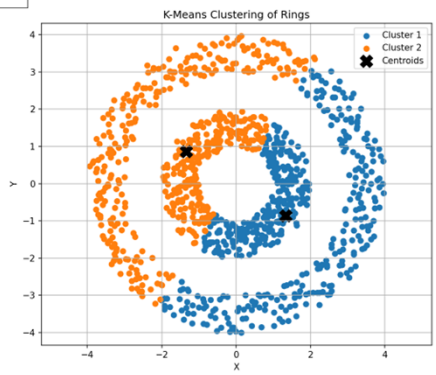


Adv. App. of AI and DT 24



نقاط ضعف K-means

داده‌های هم مرکز



خوشه بندی طیفی (Spectral Clustering)

- برخلاف روش‌های سنتی مانند K-Means که بر پایه فاصله‌های اقلیدسی عمل می‌کنند، خوشه‌بندی طیفی از نظریه گراف و جبر خطی بهره می‌برد.
- این روش به‌ویژه زمانی مؤثر است که خوشه‌ها شکل‌های غیرخطی یا ساختارهای پیچیده‌ای داشته باشند.

بین هر دو نمونه در داده، یک مقدار شباهت محاسبه می‌شود.
این شباهت می‌تواند به شکل گوسی باشد:

$$S(i,j) = \exp\left(-\frac{\|x_i - x_j\|^2}{2\sigma^2}\right)$$

یا می‌تواند یک گراف k -نزدیک‌ترین همسایه باشد.

D ماتریس درجه گره ها و L ماتریس لاپلاسین گراف است

$$L = D - S$$

و L_{sym} لاپلاسین نرمالایز شده است.

$$L_{sym} = D^{-1/2} L D^{-1/2}$$

چند بردار ویژه (معمولاً k تا، به تعداد خوشه‌ها) از ماتریس لاپلاسین استخراج می‌شود.

داده‌ها را در فضای جدیدی که توسط بردارهای ویژه تعریف شده نمایش داده شده و سپس یک الگوریتم ساده مثل K-Means را روی آن اجرا می‌شود.

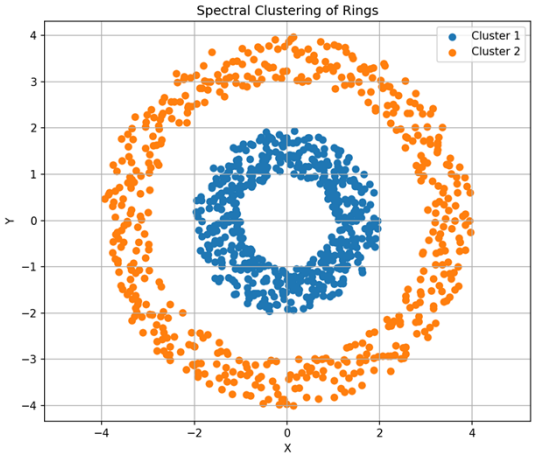
خوشه بندی طیفی (Spectral Clustering)

```
import pandas as pd
from sklearn.cluster import SpectralClustering
import matplotlib.pyplot as plt

# Load the rings data from the CSV file
csv_file_path = "two_rings_data.csv"
data = pd.read_csv(csv_file_path)
# Extract the coordinates
X = data[['x', 'y']]
# Apply Spectral Clustering
spectral = SpectralClustering(n_clusters=2,
affinity='nearest_neighbors', random_state=42)
data['Cluster'] = spectral.fit_predict(X)

# Plot the clustered data
plt.figure(figsize=(8, 8))
for cluster in range(2):
    cluster_data = data[data['Cluster'] == cluster]
    plt.scatter(cluster_data['x'], cluster_data['y'], label=f'Cluster
{cluster + 1}')

plt.title("Spectral Clustering of Rings")
plt.xlabel("X")
plt.ylabel("Y")
plt.legend()
plt.grid(True)
plt.axis('equal')
plt.show()

# Print cluster assignments
print(data.head())
```

12 spectral clustering.py

خوشه بندی طیفی (Spectral Clustering)

داده‌های هم مرکز یک روش استفاده از خوشه بندی طیفی است که فاصله بین نقاط را اندازه می گیرد.



خوشه بندی طیفی (Spectral Clustering)

مزایا:

- ✓ مناسب برای خوشه‌های با شکل یا با مرزهای پیچیده
- ✓ مبتنی بر گراف
- ✓ مناسب برای داده‌هایی با ساختارهای غیرخطی
- ✓ بهره‌گیری از ساختار جهانی داده‌ها (نه فقط فاصله‌های محلی)

محدودیت‌ها:

- ✓ نیاز به محاسبه مقادیر ویژه که هزینه محاسباتی بالایی دارد.
- ✓ محاسبات سنگین برای داده‌های بزرگ
- ✓ حساس به انتخاب پارامترهایی مانند σ در تابع گوسی یا k در k-NN

Adv. App. of AI and DT 29

خوشه بندی DBSCAN

(Density-Based Spatial Clustering of Applications with Noise)

- ✓ DBSCANیک الگوریتم خوشه‌بندی مبتنی بر چگالی است و برخلاف K-Meansیا خوشه‌بندی طیفی، این روش نیازی به تعیین تعداد خوشه‌ها از قبل ندارد و می‌تواند خوشه‌هایی با شکل‌های نامنظم را به‌خوبی تشخیص دهد. دارای مراحل زیر است
- ✓ برای هر نقطه، تعداد نقاط موجود در شعاع همسایگی (eps: ε) محاسبه می‌شود.
- ✓ هر نقطه اگر دارای نقاط همسایه بیشتر از تعداد حداقل نقاط ناحیه چگال (min_samples:MinPts) باشد، آن نقطه به‌عنوان یک نقطه مرکزی (Core Point) در نظر گرفته شده و یک خوشه جدید آغاز می‌شود.
- ✓ سپس تمامی همسایگان آن بررسی شده و در صورت شرایط مشابه، به همان خوشه افزوده می‌شوند.
- ✓ این فرآیند به‌صورت بازگشتی ادامه می‌یابد تا دیگر نقطه‌ای قابل افزودن به خوشه نباشد.
- ✓ خارج از شعاع همسایگی نقاط دیگر دارای شرط تقطه مرکزی یک خوشه جدید تشکیل می دهند.
- ✓ نقاطی که به هیچ خوشه‌ای تعلق ندارند، به‌عنوان نویز (Noise/Outlier) علامت‌گذاری می‌شوند.

12 DBSCAN 2.py Adv. App. of AI and DT 30

DBSCAN خوشه بندی

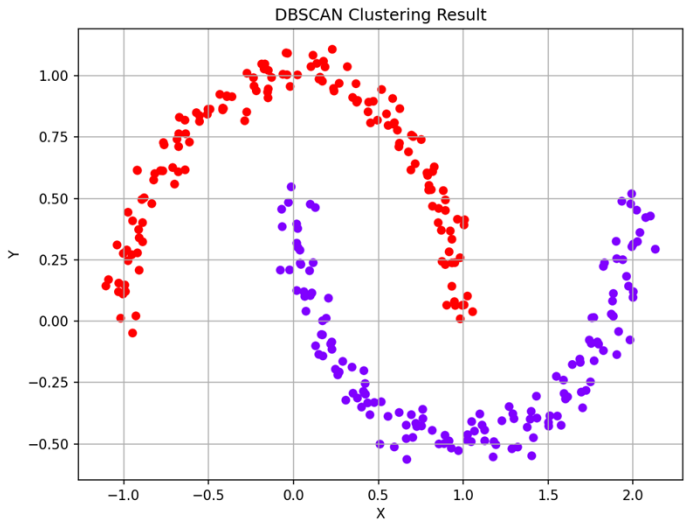
(Density-Based Spatial Clustering of Applications with Noise)

```python
# dbscan_clustering.py

import numpy as np
import matplotlib.pyplot as plt
from sklearn.datasets import make_moons
from sklearn.cluster import DBSCAN

# ایجاد داده‌های نمونه با ساختار غیر کروی
X, _ = make_moons(n_samples=300, noise=0.05, random_state=42)

# اعمال الگوریتم DBSCAN
dbscan = DBSCAN(eps=0.2, min_samples=5)
labels = dbscan.fit_predict(X)

# رسم نتایج خوشه‌بندی
plt.figure(figsize=(8, 6))
plt.scatter(X[:, 0], X[:, 1], c=labels, cmap='rainbow', s=30)
plt.title("DBSCAN Clustering Result")
plt.xlabel("X")
plt.ylabel("Y")
plt.grid(True)
plt.show()
```

12 DBSCAN 2.py Adv. App. of AI and DT 31

DBSCAN خوشه بندی

(Density-Based Spatial Clustering of Applications with Noise)



12 DBSCAN 2.py Adv. App. of AI and DT 32

خوشه بندی DBSCAN

(Density-Based Spatial Clustering of Applications with Noise)

مزایا:

- ✓ خوشه‌بندی بر اساس چگالی
- ✓ تشخیص داده‌های پرت
- ✓ نیازی به تعیین تعداد خوشه‌ها نیست
- ✓ مناسب برای داده‌هایی با شکل خوشه‌ای پیچیده و خوشه بندی تصاویر
  مناسب برای خوشه بندی داده‌های فضایی مثل GPS

محدودیت‌ها:

- ✓ حساس به پارامترهای شعاع همسایگی(eps) و تعداد حداقل نقاط برای یک ناحیه چگال (min_samples)
- ✓ عملکرد ضعیف در داده‌های با چگالی‌های متفاوت.
- ✓ در داده‌های با بُعد بالا عملکرد و دقت کاهش می‌یابد

Adv. App. of AI and DT 33

خوشه بندی مدلهای ترکیبی گوسی

(GMM :Gaussian Mixture Models)

بر خلاف الگوریتم‌هایی مانند K-Means که هر نقطه را به‌طور قطعی به یک خوشه اختصاص می‌دهند، GMM به هر نقطه یک احتمال تعلق به هر خوشه می‌دهد، که به آن «خوشه‌بندی نرم» (soft clustering) گفته می‌شود.

در این روش هر خوشه توسط یک توزیع گوسی با پارامترهای خاص خود (میانگین، ماتریس کوواریانس و وزن) مدل‌سازی می‌شود. هدف، تخمین این پارامترها به‌گونه‌ای است که توزیع ترکیبی حاصل، بیشترین شباهت را با داده‌های مشاهده‌شده داشته باشد.

الگوریتم EM (Expectation-Maximization) برای تخمین پارامترهای GMM به‌کار می‌رود

- ✓ تعیین تعداد خوشه‌ها (K) و مقداردهی اولیه به میانگین‌ها (μ)، ماتریس‌های کوواریانس (Σ) و ضرایب اختلاط (π) برای هر خوشه.
- ✓ مرحله E: محاسبه احتمال تعلق هر نقطه داده به هر خوشه، با استفاده از توزیع گوسی و ضرایب اختلاط. این احتمال‌ها به‌عنوان «مسئولیت‌ها» (responsibilities) شناخته می‌شوند
- ✓ مرحله M: به‌روزرسانی پارامترهای مدل (μ، Σ، π) با استفاده از مسئولیت‌های محاسبه‌شده در مرحله قبل
- ✓ محاسبه تابع لگاریتم احتمال (log-likelihood) و بررسی تغییرات آن. اگر تغییرات کمتر از یک آستانه معین باشد، الگوریتم متوقف می‌شود؛ در غیر این صورت، مراحل E و M تکرار می‌شوند.

Adv. App. of AI and DT 34

خوشه بندی مدلهای ترکیبی گوسی

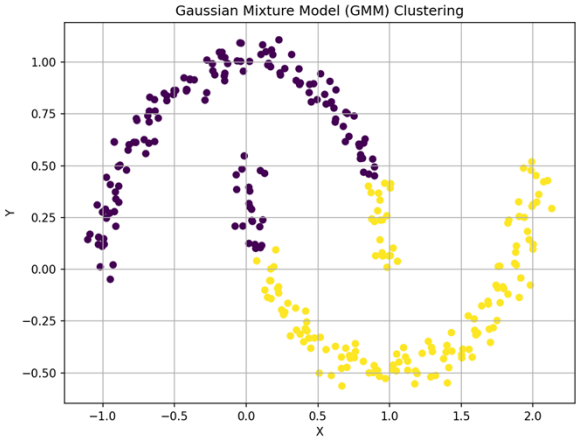
GMM بر اساس احتمال تعلق نقاط به هر خوشه کار می‌کند.

```
# gmm_clustering.py

import numpy as np
import matplotlib.pyplot as plt
from sklearn.datasets import make_moons
from sklearn.mixture import GaussianMixture

# تولید داده‌های مصنوعی به شکل نیم‌دایره
X, _ = make_moons(n_samples=300, noise=0.05,
random_state=42)

# اعمال GMM با دو خوشه
gmm = GaussianMixture(n_components=2,
covariance_type='full', random_state=42)
labels = gmm.fit_predict(X)

# رسم نتایج خوشه‌بندی
plt.figure(figsize=(8, 6))
plt.scatter(X[:, 0], X[:, 1], c=labels,
cmap='viridis', s=30)
plt.title("Gaussian Mixture Model (GMM)
Clustering")
plt.xlabel("X")
plt.ylabel("Y")
plt.grid(True)
plt.show()
```

12 GMM Clustering.py

خوشه بندی مدلهای ترکیبی گوسی

(GMM :Gaussian Mixture Models)



خوشه‌ها می‌توانند هم‌پوشانی داشته باشند و شکل بیضوی بگیرند (خوشه‌ها می‌توانند کشیده و در یک جهت خاص امتداد داشته باشند)

خوشه بندی مدلهای ترکیبی گوسی

(Gaussian Mixture Models (GMM))

مزایا:

- ✓ خوشه‌بندی نرم: مدل‌سازی احتمالی خوشه‌ها یعنی هر هر نقطه به چند خوشه می تواند تعلق داشته باشد
- ✓ انعطاف‌پذیری در شکل خوشه‌ها: قابلیت مدل‌سازی خوشه‌هایی با اشکال بیضوی و اندازه‌های مختلف.
- ✓ مناسب برای داده‌هایی با توزیع نرمال یا مخلوطی از توزیع‌های گوسی
- ✓ توانایی تشخیص خوشه‌ها حتی در صورت همپوشانی آن‌ها

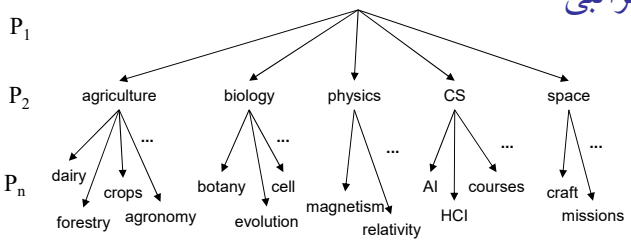
محدودیت‌ها:

- ✓ نیاز به تعیین تعداد خوشه‌ها
- ✓ حساسیت به مقداردهی اولیه
- ✓ پیچیدگی محاسباتی به‌ویژه در داده‌های با ابعاد بالا یا تعداد زیاد خوشه‌ها
- ✓ عملکرد ضعیف در داده‌های غیر نرمال

Adv. App. of AI and DT 37

Hierarchical Clustering

خوشه‌بندی سلسله مراتبی



یک توالی تو در تو از خوشه ها یا دیاگرام درختی(دندروگرام) تولید می کند

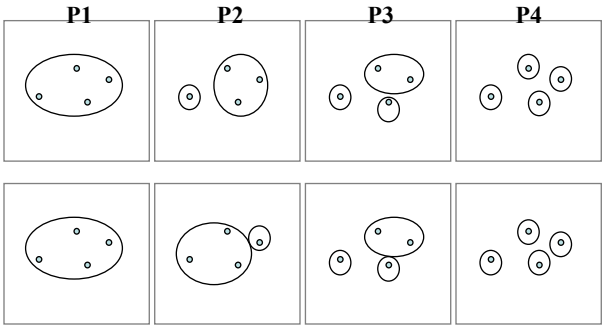
خوشه‌بندی سلسله مراتبی، سلسله‌ای از پارتیشن‌ها ارایه می کند

a partition P_1 into 1 clusters (the entire collection)
• a partition P_2 into 2 clusters
…
• a partition P_n into n clusters (each object forms its own cluster)
It is then up to the user to decide which of the partitions reflects actual sub-populations in the data.

Adv. App. of AI and DT 38

Hierarchical Clustering

در صورتی دنباله‌ای از پارتیشن‌بندی سلسله مراتبی است که هر خوشه در یک پارتیشن معین اجتماعی از خوشه‌ها در پارتیشن بزرگتر بعدی باشد



hierarchical sequence of partitions

non hierarchical sequence

Hierarchical Clustering

روشهای خوشه‌بندی سلسله مراتبی

روش تجمیعی (از پایین به بالا)

Agglomerative methods:

- Start with partition P_n, where each object forms its own cluster.
- Merge the two closest clusters, obtaining P_{n-1}.
- Repeat merge until only one cluster is left.

روش تقسیمی(از بالا به پایین)

Divisive methods:

- Start with P_1.
- Split the collection into two clusters that are as homogenous (and as different from each other) as possible.
- Apply splitting procedure recursively to the clusters.

در روش تجمیعی نیاز به قاعده ای برای تجمیع خوشه‌ها و در روش تقسیمی نیاز به قاعده ای برای تقسیم خوشه‌ها است که معمولا با استفاده از مقدار فاصله انجام می‌شوند.

خوشه‌بندی سلسله مراتبی تجمیعی

تعیین فاصله بین خوشه‌ها

d(P,Q) فاصله بین خوشه‌های P و Q بر اساس فاصله مشاهدات در دو خوشه d(x,y) تعیین می‌شود

For x in P, y in Q

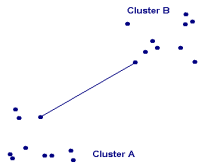
1. $d_1(P,Q) = \min d(x,y)$ (single linkage) — حداقل فاصله بین اعضای دو خوشه
2. $d_2(P,Q) = \text{ave } d(x,y)$ (average linkage) — میانگین فاصله بین تمام اعضای دو خوشه
3. $d_3(P,Q) = \max d(x,y)$ (complete linkage) — حداکثر فاصله بین اعضای دو خوشه
4. $d_4(P,Q) = \left\| \bar{x}_P - \bar{x}_Q \right\|$ (centroid method)
5. $d_5(P,Q) = 2\frac{|P||Q|}{|P|+|Q|}\left\| \bar{x}_P - \bar{x}_Q \right\|^2$ (Ward's method) — کمینه‌سازی واریانس درون خوشه‌ها هنگام ادغام

d_5 is called Ward's distance.

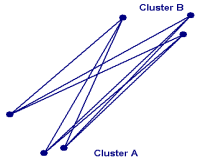
خوشه‌بندی سلسله مراتبی تجمیعی

تعیین فاصله بین خوشه‌ها

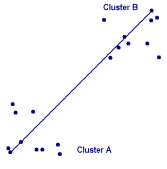
$d_1(P,Q) = \min d(x,y)$ (single linkage)



$d_2(P,Q) = \text{ave } d(x,y)$ (average linkage)



$d_3(P,Q) = \max d(x,y)$ (complete linkage)



خوشه‌بندی سلسله مراتبی تجمیعی

تعیین فاصله بین خوشه‌ها – فاصله Ward

Let $\mathbf{P}_k = P_1, \ldots, P_k$ be a partition of the observations into k groups.

- Measure goodness of a partition by the sum of squared distances of observations from their cluster means:

$$RSS(\mathbf{P}_k) = \sum_{i=1}^{k} \sum_{j \in P_i} \left\| x_j - \bar{x}_{P_i} \right\|^2$$

- Consider all possible (k-1)-partitions obtainable from $\mathbf{P}_k$ by a merge
- Merging two clusters with smallest Ward's distance optimizes goodness of new partition.

خوشه‌بندی سلسله مراتبی تقسیمی

There are divisive versions of single linkage, average linkage, and Ward's method.

Divisive version of single linkage:

- Compute minimal spanning tree (graph connecting all the objects with smallest total edge length.
- Break longest edge to obtain 2 subtrees, and a corresponding partition of the objects.
- Apply process recursively to the subtrees.

Agglomerative and divisive versions of single linkage give identical results (more later).

خوشه‌بندی سلسله مراتبی تقسیمی

Divisive version of Ward's method.

Given cluster R.

Need to find split of R into 2 groups P,Q to minimize

$$RSS(P,Q) = \sum_{i \in P} \left\| \mathbf{x}_i - \bar{\mathbf{x}}_P \right\|^2 + \sum_{j \in Q} \left\| \mathbf{x}_j - \bar{\mathbf{x}}_Q \right\|^2$$

or, equivalently, to maximize Ward's distance between P and Q.

Note: No computationally feasible method to find optimal P, Q for large |R|. Have to use approximation.

Adv. App. of AI and DT 45

خوشه‌بندی سلسله مراتبی تقسیمی

Iterative algorithm to search for the optimal Ward's split

Project observations in R on largest principal component.

Split at median to obtain initial clusters P, Q.

Repeat { Assign each observation to cluster with closest mean

Re-compute cluster means

} Until convergence

Note:

- Each step reduces RSS(P, Q)
- No guarantee to find optimal partition.

Divisive version of average linkage

Algorithm Diana, Struyf, Hubert, and Rousseuw

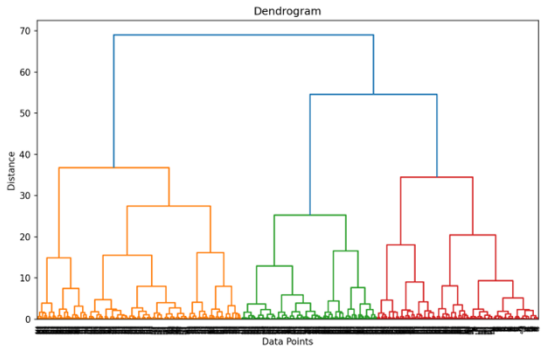
Adv. App. of AI and DT 46

خوشه‌بندی سلسله مراتبی

خوشه بندی سلسله مراتبی بوسیله دیاگرام درختی(دندروگرام) نمایش داده می‌شود

Result of hierarchical clustering can be represented as binary tree:
- Root of tree represents entire collection
- Terminal nodes represent observations
- Each interior node represents a cluster
- Each subtree represents a partition

y-coordinate of vertex = distance between daughter clusters.



Adv. App. of AI and DT 47

بررسی خوشه‌بندی

Silhouette Score

It measures how well each data point fits within its assigned cluster compared to other clusters.

For a single data point *i*, the silhouette score *s(i)* is calculated as:

$$s(i) = \frac{b(i) - a(i)}{\max(a(i), b(i))}$$

a(i): The mean intra-cluster distance
b(i): The mean nearest-cluster distance

The score ranges from **-1 to 1**, where:
s(i) = **1** **well-clustered**: the data point is well-clustered and far from neighboring clusters.
s(i) = **0** **between clusters**: indicates that the data point lies on the boundary between clusters.
s(i) = **-1** **wrong cluster**: indicates that the data point may be assigned to the wrong cluster.

Adv. App. of AI and DT 48

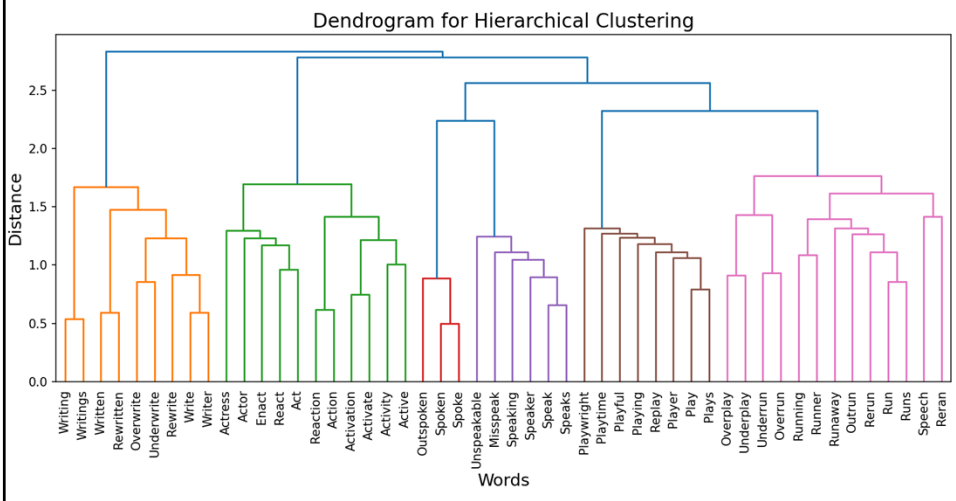
خوشه‌بندی سلسله مراتبی

مثال

```
import matplotlib.pyplot as plt
from sklearn.datasets import make_blobs
from scipy.cluster.hierarchy import dendrogram, linkage, fcluster
from sklearn.metrics import silhouette_score
from sklearn.cluster import AgglomerativeClustering

# Generate artificial data using make_blobs
X, y = make_blobs(n_samples=500, centers=3, cluster_std=1.0, random_state=42)


# Perform hierarchical clustering
# 1. Compute linkage matrix
linkage_matrix = linkage(X, method='ward')  # 'ward' linkage minimizes variance

# 2. Plot the dendrogram
plt.figure(figsize=(12, 8))
dendrogram(linkage_matrix, truncate_mode='level', p=5)  # Display only the top 5
levels
plt.title("Dendrogram")
plt.xlabel("Data Points")
plt.ylabel("Distance")
plt.show()

# Apply Agglomerative Clustering
n_clusters = 3  # Number of clusters
agg_clustering = AgglomerativeClustering(n_clusters=n_clusters, linkage='ward')
labels = agg_clustering.fit_predict(X)
```

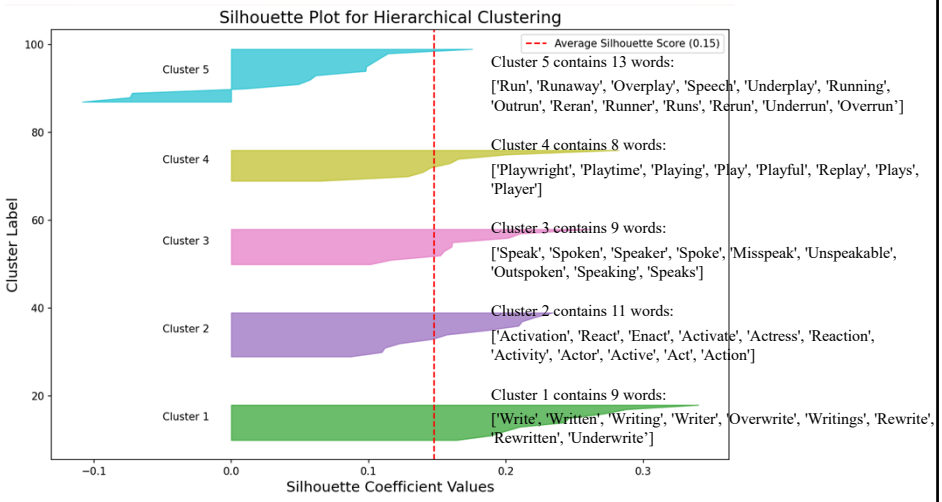
12 Hierarchical Clustering.py Adv. App. of AI and DT 49

خوشه‌بندی سلسله مراتبی



Silhouette Score for 5 clusters: 0.15

Adv. App. of AI and DT 50



خوشه بندی سلسله مراتبی

مزایا:

- ✓ درخت دندروگرام می‌سازد (قابل تفسیر)
- ✓ نیازی به تعیین تعداد خوشه‌ها از ابتدا نیست
- ✓ مناسب برای تحلیل اکتشافی داده‌های کوچک تا متوسط

محدودیت‌ها:

- ✓ مقیاس‌پذیری پایین برای داده‌های بزرگ

تمرین برنامه نویسی

تمرین ششم : یک برنامه به زبان پایتون بنویسید که یک فایل داده را خوانده و خوشه‌بندی نماید.

۱- به روش K means

۲- با استفاده از خوشه‌بندی سلسله مراتبی

۳- بررسی صحت نتایج برای دوحالت فوق