АЛГОРИТМ КЛАСТЕРИЗАЦИИ “DBSCAN” В ПРОИЗВОДСТВЕ

Воробьёв Игорь Константинович

студент, Лысьвенский филиал пермского национального исследовательского политехнического университета, РФ, г. Лысьва

Торощин Александр Константинович

научный руководитель, Лысьвенский филиал пермского национального исследовательского политехнического университета, РФ, г. Лысьва

DBScan (Density-Based Spatial Clustering of Applications with Noise) — это алгоритм кластеризации, основанный на плотности. В отличие от методов, таких как K-means, DBScan не требует заранее задавать количество кластеров, что делает его удобным для анализа данных, где количество кластеров неизвестно. Основные параметры DBScan:

1. eps: радиус окрестности точки.
2. min_samples: минимальное количество точек, необходимое для образования кластера.

Алгоритм выделяет кластеры, основываясь на плотности: область считается кластером, если в её пределах находится достаточно большое количество точек.

На производстве задача кластеризации часто сводится к анализу данных, полученных с различных сенсоров и сканеров. В данном случае будем рассматривать процесс сканирования рессоры в кроватке с использованием сканера Rifttek.

Сканер Rifttek представляет собой систему, в которой лазерный луч движется над объектом (в данном случае — рессорой), а камера сканирует отраженные от объекта лучи. В результате получается облако точек, которое представляет собой трехмерную модель объекта.

Облако точек — это совокупность множества точек в пространстве, каждая из которых имеет координаты (X, Y, Z). Эти точки могут быть использованы для восстановления трехмерной формы объекта.

Первым шагом обработки является получение данных с помощью сканера Rifttek. Предположим, что данные хранятся в формате CSV, где каждая строка представляет собой координаты одной точки.

Перед применением алгоритма DBScan, данные необходимо нормализовать, чтобы все координаты находились в одном масштабе.

Воробьёв И.К. АЛГОРИТМ КЛАСТЕРИЗАЦИИ “DBSCAN” В ПРОИЗВОДСТВЕ // Студенческий форум: электрон. научн. журн. 2024. № 22(289). URL: https://nauchforum.ru/journal/stud/289/150893 (дата обращения: 07.09.2025).

```
import pandas as pd
from sklearn.preprocessing import StandardScaler
from sklearn.cluster import DBSCAN
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D
# Загрузка данных
data = pd.read_csv('point_cloud.csv')
points = data[['X', 'Y', 'Z']].values
# Нормализация данных
scaler = StandardScaler()
points_scaled = scaler.fit_transform(points)
# Применение DBScan
dbscan = DBSCAN(eps=0.5, min_samples=10)
clusters = dbscan.fit_predict(points_scaled)
# Визуализация результатов
fig = plt.figure()
ax = fig.add_subplot(111, projection='3d')
scatter = ax.scatter(points[:, 0], points[:, 1],
points[:, 2], c=clusters, cmap='viridis')
legend1 = ax.legend(*scatter.legend_elements(),
title="Кластеры")
ax.add_artist(legend1)
plt.show()
```

Рисунок 1. Простейшее применение DBScan

Далее можно применить алгоритм DBScan к нормализованным данным. Необходимо выбрать параметры eps и min_samples, которые будут соответствовать особенностям нашего набора данных.

После выполнения алгоритма каждая точка будет принадлежать к определенному кластеру или будет отмечена как шум. Для визуализации результатов можно использовать библиотеку matplotlib.

```
# Загрузка идеальной модели
ideal_data = pd.read_csv('ideal_spring.csv')
ideal_points = ideal_data[['X', 'Y', 'Z']].values
ideal_points_scaled = scaler.transform(ideal_points)

# Кластеризация идеальной модели
ideal_clusters = dbscan.fit_predict(ideal_points_scaled)

# Сравнение результатов
defect_points = points[clusters != ideal_clusters]
```

Рисунок 2.Сравнение скана с идеальным сканом

Одной из задач применения DBScan может быть обнаружение дефектов на поверхности рессоры. Предположим, что у нас есть модель идеальной рессоры, с которой можно сравнивать текущие данные. Кластеры, которые не соответствуют идеальной модели, могут быть помечены как дефекты.

Кластеризация может быть использована для оптимизации производственных процессов, например, для контроля качества продукции. Анализируя кластеры и распределение точек, можно выявить участки, которые требуют дополнительного контроля или настройки оборудования.

Список литературы:

1. Ester, M., Kriegel, H.-P., Sander, J., & Xu, X. (1996). A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise. Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD-96), 226-231.

2. Han, J., Kamber, M., & Pei, J. (2011). Data Mining: Concepts and Techniques (3rd ed.). Morgan Kaufmann Publishers.

3. Schubert, E., Sander, J., Ester, M., Kriegel, H.-P., & Xu, X. (2017). DBSCAN Revisited, Revisited: Why and How You Should (Still) Use DBSCAN. ACM Transactions on Database Systems, 42(3), Article 19.

4. Bishop, C. M. (2006). Pattern Recognition and Machine Learning. Springer.

5. Scikit-learn documentation. (n.d.). DBSCAN: Density-Based Spatial Clustering of Applications with Noise. Retrieved from https://scikit-learn.org/stable/modules/generated/sklearn.cluster.DBSCAN.html