Algorytmy Klasteryzacji Strumieni Danych

Maksymilian Neumann

Listopad 2024

Strumienie Danych

- **Definicja:** Uporządkowany ciąg d-wymiarowych instancji.

$$S = \{x_1, x_2, x_3, \ldots, x_i, \ldots, x_N\}$$

  Gdzie każda instancja(x_i) jest d-wymiarowym wektorem
- **Charakterystyka:**
  - Dane o dużym wolumenie, potencjalnie nieskończone.
  - Dane przychodzą z dużą prędkością.
  - Po przetworzeniu dane są zazwyczaj odrzucane.
- **Przykłady:**
  - Strumienie kliknięć na stronie internetowej.
  - Dane z sensorów w systemach IoT.
  - Transakcje finansowe na giełdach.

Klasteryzacja

- **Definicja:** Klasteryzacja to proces grupowania obiektów w taki sposób, aby obiekty w tej samej grupie (klastrze) były bardziej podobne do siebie niż do obiektów w innych grupach.
- **Cel:** Identyfikacja ukrytych wzorców lub struktur w danych.
- **Zastosowania:**
  - Segmentacja rynku.
  - Segmentacja obrazów.
  - Wykrywanie anomalii.
- **Przykładowe algorytmy:**
  - K-means.
  - DBSCAN.
  - Klasteryzacja hierarchiczna.

Charakterystyka Klasteryzacji Strumieni Danych

- **Dynamiczność:** Algorytmy muszą działać w czasie rzeczywistym.
- **Przetwarzanie przyrostowe:** Dane są przetwarzane raz, bez możliwości ich ponownego odczytu (model jednokrotnego przetwarzania danych - *single-pass*).
- **Ograniczenia pamięci:** Wymagana jest efektywna alokacja pamięci, ponieważ strumień danych może być potencjalnie nieskończony.
- **Skalowalność:** Algorytmy muszą być skalowalne zarówno w odniesieniu do liczby przychodzących punktów danych, jak i liczby klastrów.
- **Ewolucja klastrów:** Klastery mogą zmieniać się w czasie (pojawianie się, zanikanie, zmiana rozmiaru).

Wprowadzenie do BIRCH

- **BIRCH:** *Balanced Iterative Reducing and Clustering using Hierarchies* – algorytm klasteryzacji hierarchicznej dla dużych zbiorów danych.
- **Zalety:**
  - Przetwarzanie przyrostowe i dynamiczne.
  - Optymalizacja jakości klasteryzacji przy ograniczonych zasobach (czas i pamięć).
  - Zwykle wymaga jednego przejścia przez dane.
- **Ciekawostki:**
  - Jeden z pierwszych algorytmów skutecznie radzących sobie ze "szumem" w danych.
  - Wyróżniony nagrodą SIGMOD Test of Time w 2006 roku.

Cechy Klastra

Klaster N d-wymiarowych punktów możemy opisać jako: $K = \{\vec{X}_i\}$ gdzie $i = 1, \ldots, N$ **Centroida** $\vec{C}$ i **Promień** R Klastra zdefiniujemy jako:

$$\vec{C} = \frac{\sum_{t=i}^{N} \vec{X}_i}{N}$$

$$R = \sqrt{\frac{\sum_{i=1}^{N} \left(\vec{X}_i - \vec{X}_0\right)^2}{N}}$$

Oraz dystans między klastrami $d(K_a, K_b)$ jako:

$$d(K_a, K_b) = \sqrt{(\vec{C}_a - \vec{C}_b)^2}$$

Clustering Feature

Clustering Feature(CF): to streszczenie klastra K definiowane jako $(N, \vec{LS}, SS)$

$$N = |K|$$

$$\vec{LS} = \sum_{i=1}^{N} \vec{X_i}$$

$$SS = \sum_{i=1}^{N} \vec{X_i} \cdot \vec{X_i}$$

Cechy CF

Mając $(N, \vec{LS}, SS)$ możemy łatwo obliczyć cechy klastra:

$$\vec{C} = \frac{\sum_{t=i}^{N} \vec{X}_i}{N} = \frac{\vec{LS}}{N}$$

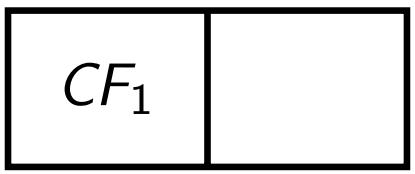
$$R = \sqrt{\frac{\sum_{i=1}^{N} \left(\vec{X}_i - \vec{X}_0\right)^2}{N}} = \sqrt{\frac{SS}{N} - (\frac{\vec{LS}}{N})^2} = \sqrt{\frac{SS}{N} - (\vec{C})^2}$$

Dla klastra $K = \{\vec{X}_i\}$ potrzebujemy $O(Nd)$ pamięci natomiast dla CF opisującego K tylko $O(d+2)$

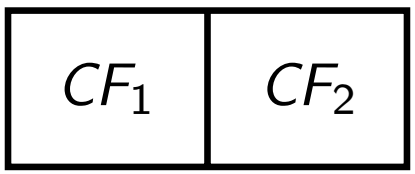
CF Tree

- **Parametry:** branching factor(B) i threshold(T)
- **Węzeł nie-liść:** ma maksymalnie B pozycji $[CF_i, child_i]$ gdzie $child_i$ jest wskaźnikiem do i-tego dziecka gdzie $i = 1, \ldots, B$
- **Węzeł liść:** ma maksymalnie L pozycji $[CF_i]$, gdzie $i = 1, \ldots, L$. Promień każdego CF_i musi być mniejszy niż T

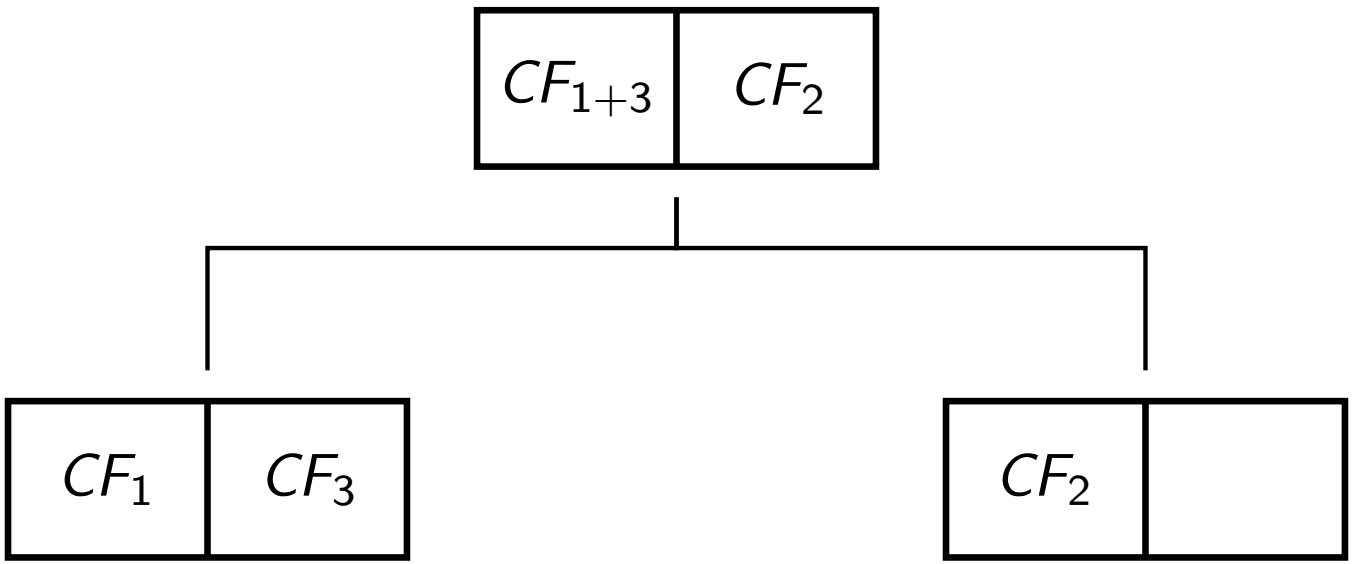
CF Tree



CF Tree



CF Tree



Wprowadzenie do CluStream

- **CluStream:** Algorytm do klasteryzacji ewoluujących strumieni danych
- **Komponenty:**
  - **Komponent online:**
    - Oblicza i przechowuje statystyki podsumowujące w postaci *mikroklastrów*.
    - Wykonuje przyrostowe przetwarzanie i utrzymanie mikroklastrów w czasie rzeczywistym.
  - **Komponent offline:**
    - Wykorzystuje zapisane statystyki do *makroklasteryzacji*.
    - Odpowiada na zapytania klasteryzacyjne.
- **Strumień Danych:** jest w formie punktów $\{\vec{X_1}, \ldots, \vec{X_N}\}$, które przychodzą o czasach $\{T_1, \ldots, T_N\}$

Micro-cluster

Definicja: Jest to czasowe rozwinięcie CF. Dla zbioru d-wymiarowych punktów $\{\vec{X}_1, \ldots, \vec{X}_n\}$ o czasach $\{T_1, \ldots, T_n\}$ micro-cluster jest zdefiniowany przez $(\overrightarrow{CF2^x}, \overrightarrow{CF1^x}, CF2^t, CF1^t, n)$ który zajmuje $O(2d + 3)$ pamięci.

$$\overrightarrow{CF2^x} = \sum_{i=1}^{n} \vec{X}_i^2$$

$$\overrightarrow{CF1^x} = \sum_{i=1}^{n} \vec{X}_i$$

$$CF2^t = \sum_{i=1}^{n} T_i^2$$

$$CF1^t = \sum_{i=1}^{n} T_i$$

Online Micro-Cluster Maintenance

Initialize q Microclusters (M_i) with the first *InitNumber* points using k-means.
for $(\vec{X}, T)$ in S **do** ▷ Iterate over the stream
    $M_{\text{closest}} \leftarrow$ find closest M_i by dist$(\vec{X}, M_i)$
    if $\vec{X}$ is within *maximalBoundary*(M_{closest}) **then**
        Merge $\vec{X}$ with M_{closest}
    else
        Add new Microcluster from $\vec{X}$ to Microclusters
        if *safe to delete some Microcluster as outlier* **then**
            Delete the Microcluster
        else
            Merge closest Microclusters
        end if
    end if
    Save Microclusters snapshot with time T
end for

Piramidalna Przestrzeń Czasowa

Order of Snapshots	**Clock Times ($\alpha = 2$ i $l = 2$)**
0	55 54 53 52 51
1	54 52 50 48 46
2	52 48 44 40 36
3	48 40 32 24 16
4	48 32 16
5	32

Piramidalna Przestrzeń Czasowa

Order of Snapshots	**Clock Times ($\alpha = 2$ i $l = 2$)**
0	55 ~~54~~ 53 ~~52~~ 51
1	54 ~~52~~ 50 ~~48~~ 46
2	52 ~~48~~ 44 ~~40~~ 36
3	~~48~~ 40 ~~32~~ 24 ~~16~~
4	48 ~~32~~ 16
5	32

Wprowadzenie do DenStream

- **DenStream:** Algorytm klasteryzacji oparty na gęstości, zaprojektowany do pracy z ewoluującymi strumieniami danych, szczególnie w warunkach obecności szumów.
- **Kluczowe cechy:**
  - Obsługuje klastry o dowolnym kształcie, bez konieczności określania ich liczby z góry.
  - Rozróżnia między:
    - **Potencjalnymi mikroklastrami (potential micro-clusters):** Obszary o niskiej gęstości, które mogą ewoluować w klastry.
    - **Mikroklastrami odszumień (outlier micro-clusters):** Izolowane punkty traktowane jako szum.

p-micro-cluster(c_p)

Definicja: Dla zbioru d-wymiarowych punktów $\{\vec{X}_1, \ldots, \vec{X}_n\}$ o czasach $\{T_1, \ldots, T_n\}$. Z funkcją znikania $f(t) = 2^{-\lambda \cdot t}$ micro-cluster jest zdefiniowany przez $(\overrightarrow{CF^1}, CF^2, w)$.

$$\overrightarrow{CF^1} = \sum_{i=1}^{n} f(t - T_i)\vec{X}_i$$

$$CF^2 = \sum_{i=1}^{n} f(t - T_i)\vec{X}_i^2$$

$$w = \sum_{i=1}^{n} f(t - T_i)$$

Inkrementalność c_p

Dla interwału δt i punktu $\vec{X}$

$$c_p = (2^{-\lambda \delta t} \cdot \overrightarrow{CF^1}, 2^{-\lambda \delta t} \cdot, 2^{-\lambda \delta t} \cdot w)$$

$$c_p = (\overrightarrow{CF^1} + \vec{X}, CF^2 + \vec{X}^2, w + 1)$$

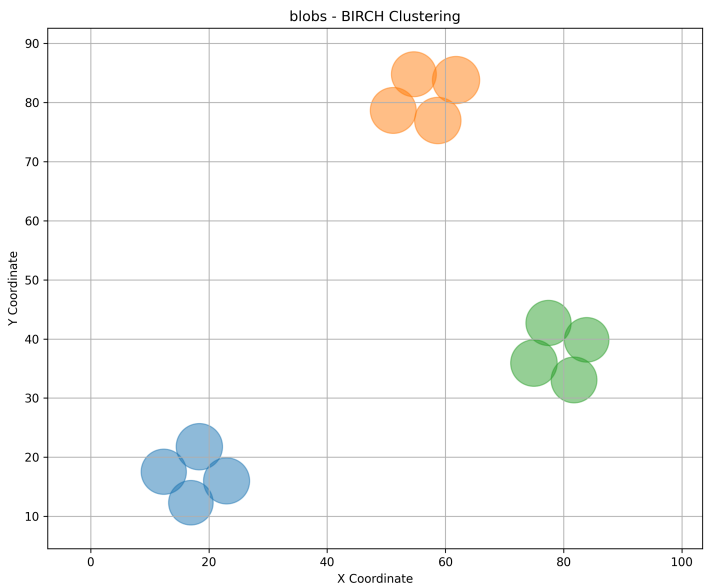
Micro-cluster Maintenance

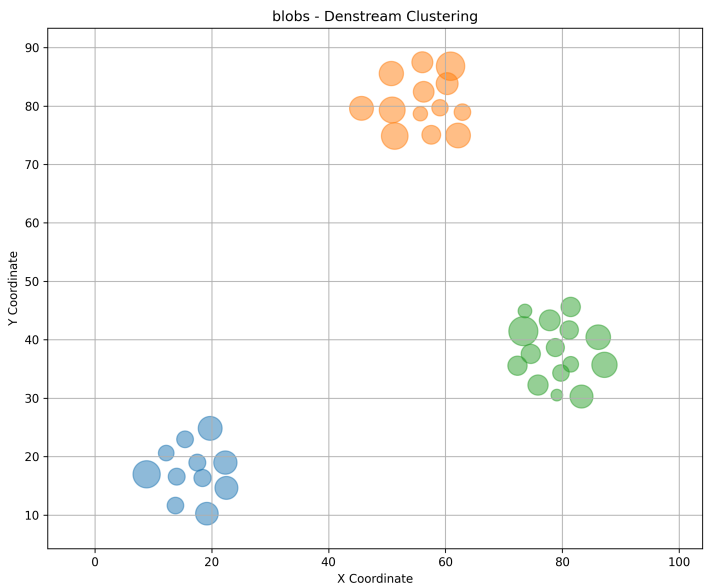
Require: Data stream S, parameters $\epsilon, \beta, \mu, \lambda$

1: $T_p = \left\lceil \frac{1}{\lambda} \log\left(\frac{\beta\mu}{\beta\mu - 1}\right) \right\rceil$
2: **while** receiving new point $\vec{X}$ at current time t from S **do**
3: Perform **Merging($\vec{X}$)**
4: **if** $t \mod T_p = 0$ **then**
5: **for** each potential micro-cluster c_p **do**
6: **if** w_p (weight of c_p) $< \beta\mu$ **then**
7: Delete c_p
8: **end if**
9: **end for**
10: **for** each outlier micro-cluster c_o **do**
11: $\xi = 2^{-\lambda(t - t_{c_o} + T_p) - 1} / (2^{-\lambda T_p} - 1)$
12: **if** w_o (weight of c_o) $< \xi$ **then**
13: Delete c_o
14: **end if**
15: **end for**
16: **end if**
17: **end while**

Porównanie z klasycznymi algorytmami

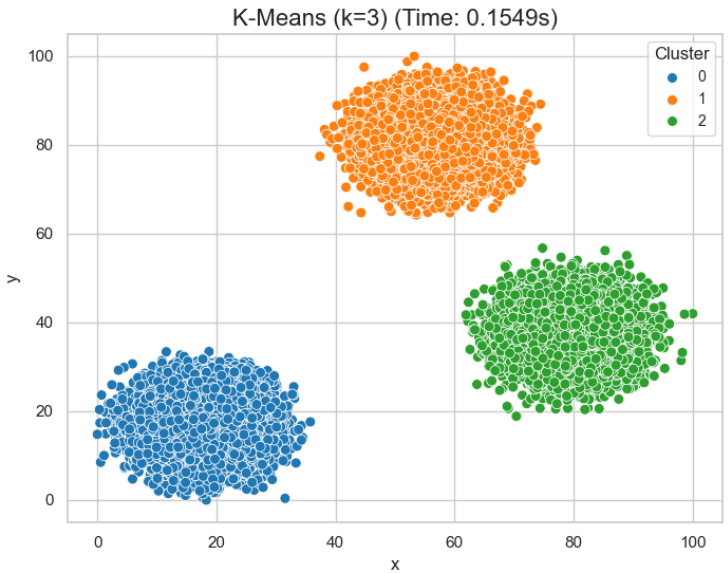
Algorytm: "BIRCH" Time: 746.9637ms
Algorytm: "DenStream" Time: 529.3339ms

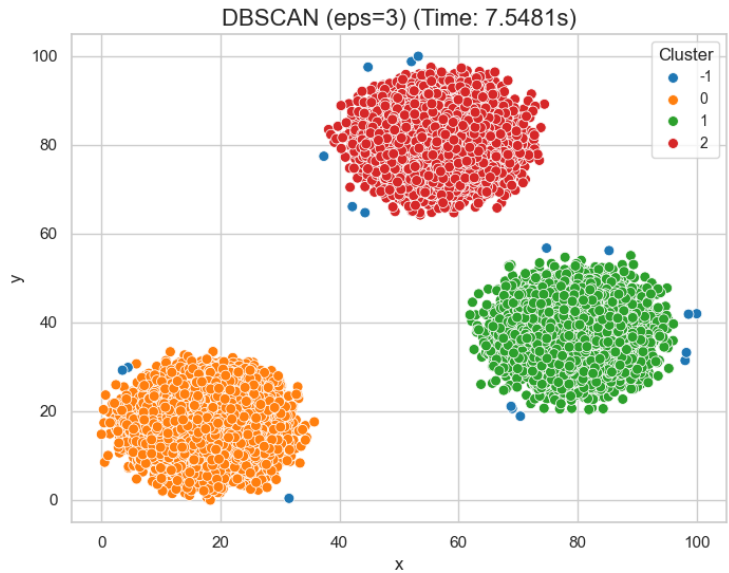




Porównanie z klasycznymi algorytmami

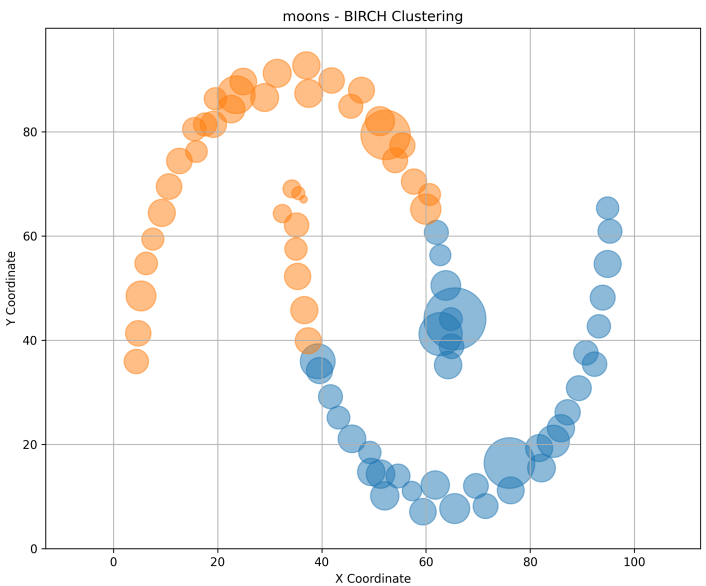
Implementacja: scikit-learn

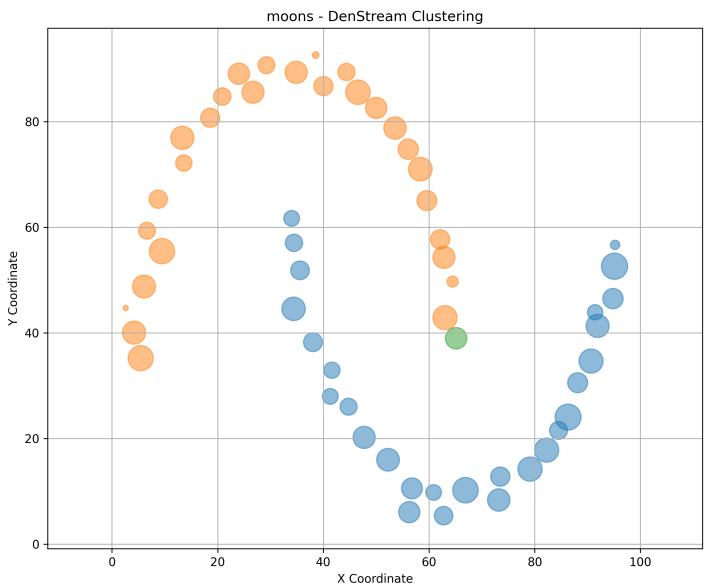




Porównanie z klasycznymi algorytmami

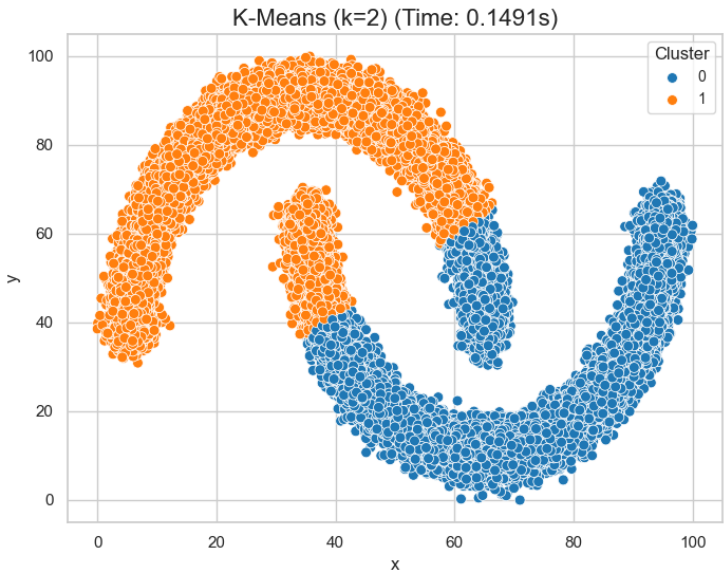
Algorytm: "BIRCH" Time: 944.8691ms
Algorytm: "DenStream" Time: 776.7824ms

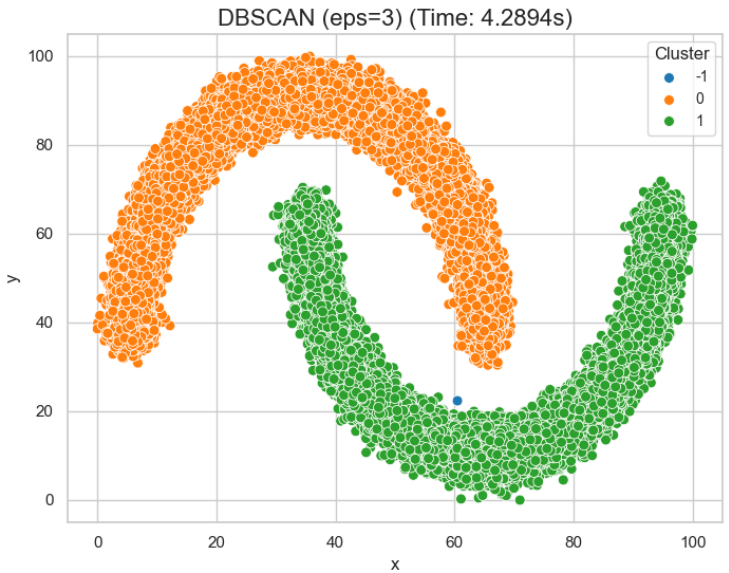




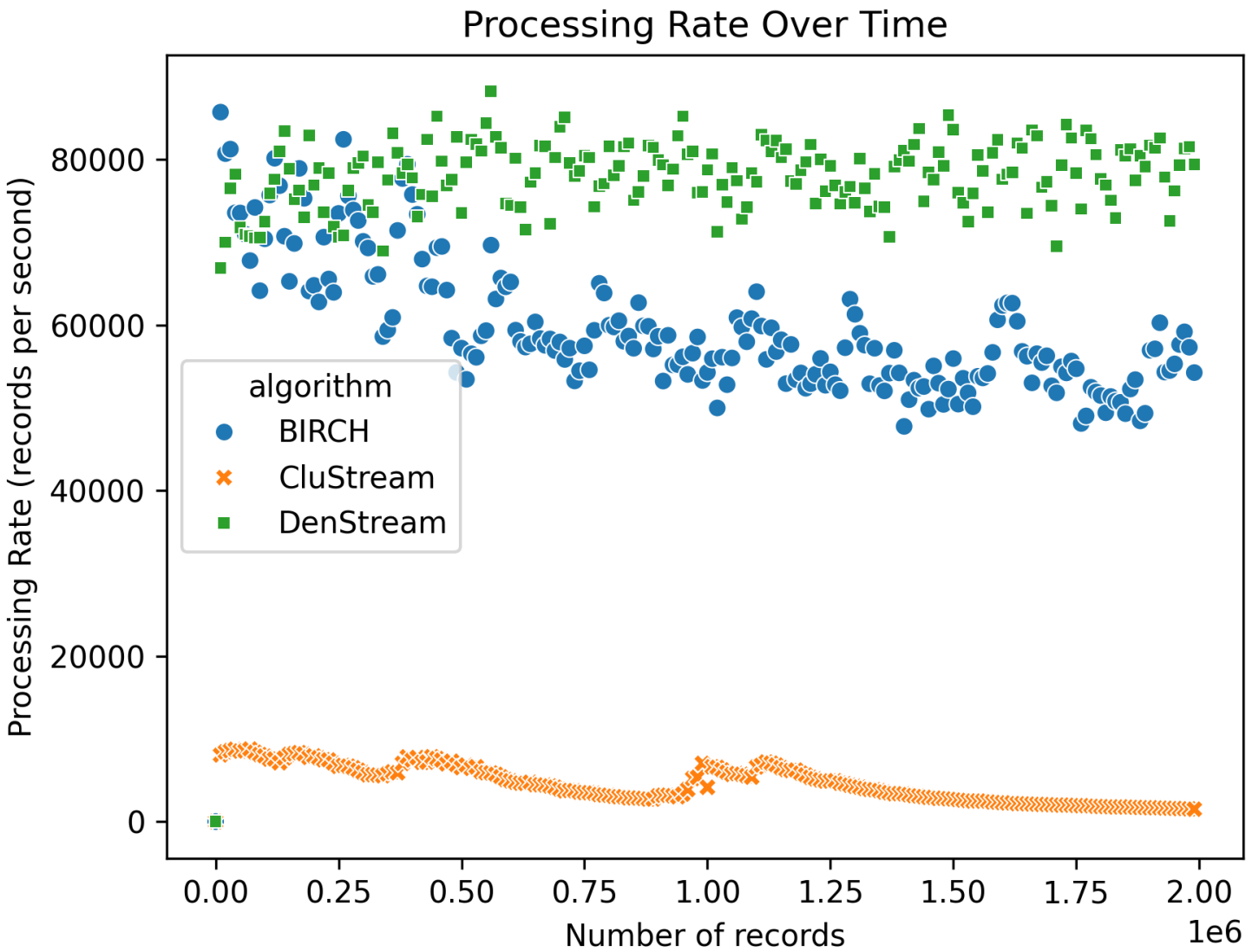
Porównanie z klasycznymi algorytmami

Implementacja: scikit-learn

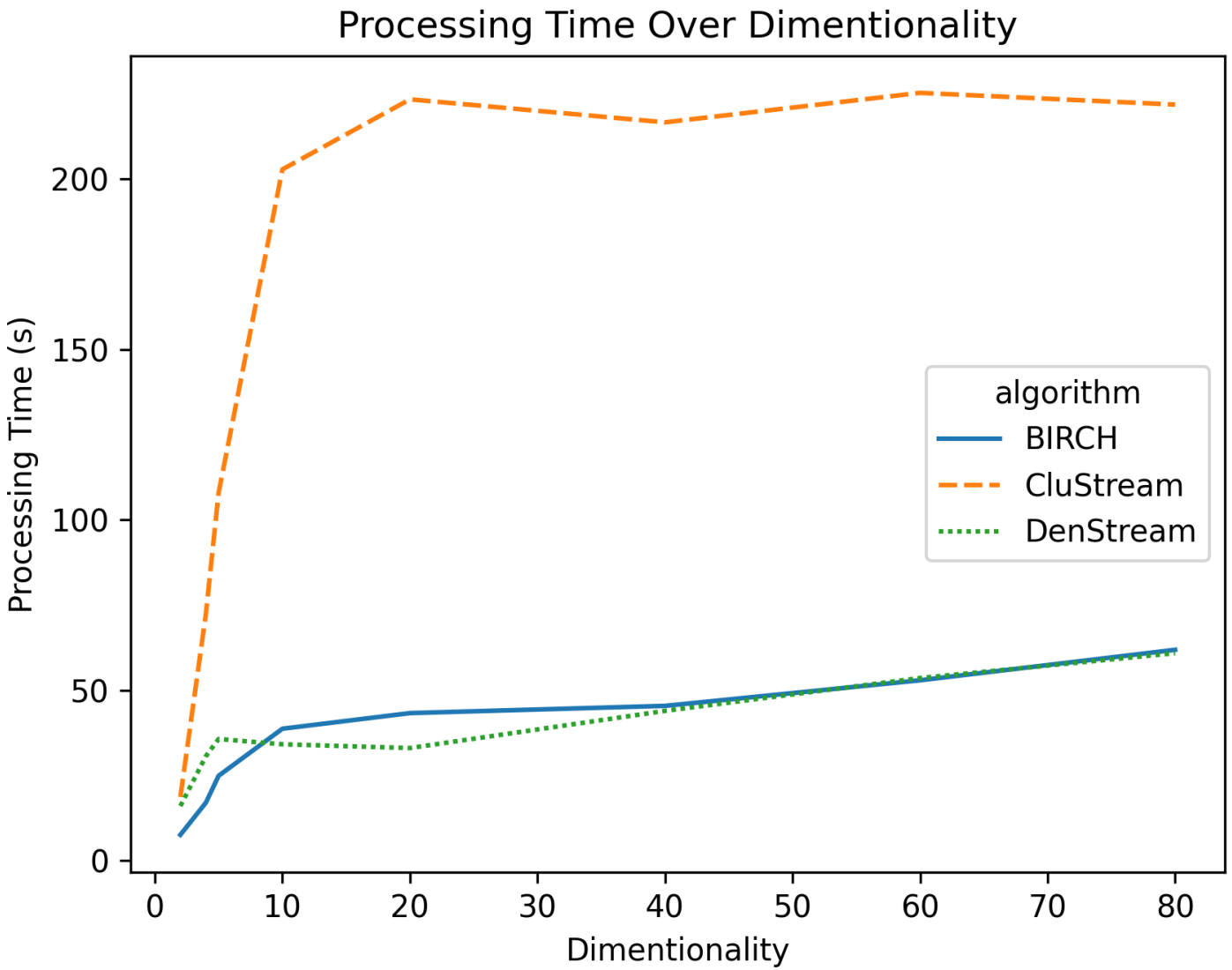




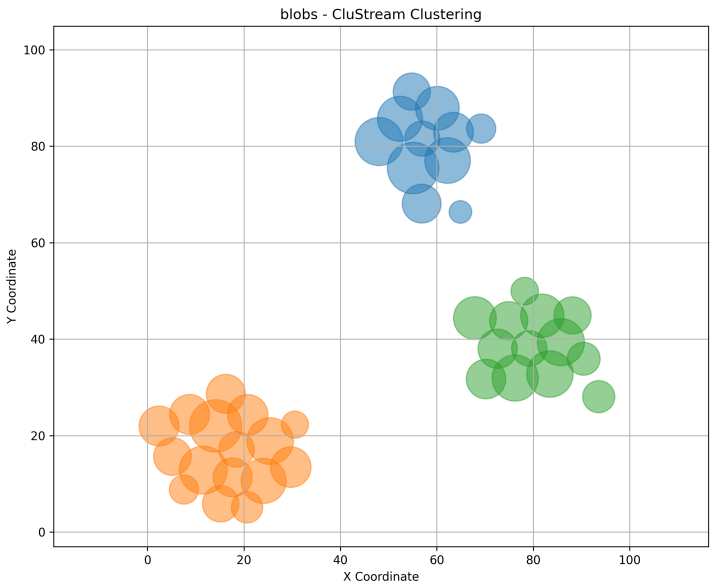
Efektywność w czasie

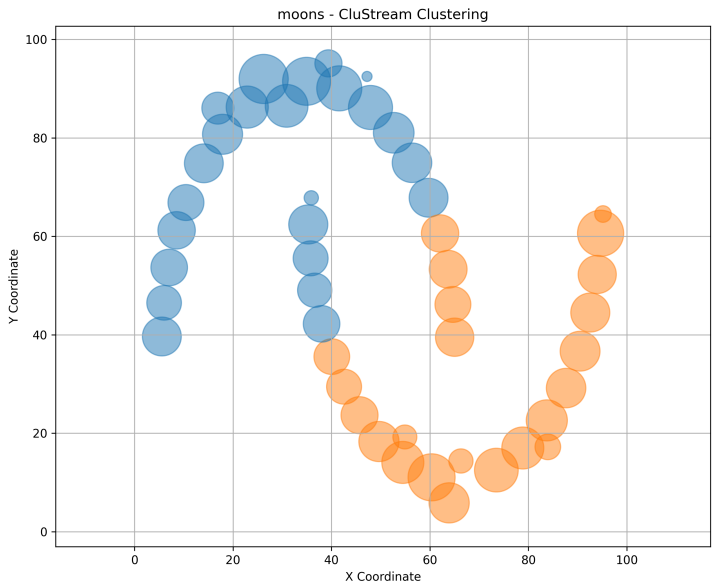


Efektywność a wymiar



CluStream


blobs - CluStream Clustering
Y Coordinate
X Coordinate
0
20
40
60
80
100


moons - CluStream Clustering
Y Coordinate
X Coordinate
0
20
40
60
80
100