Testy algorytmów analizy skupień na własnych zbiorach benchmarkowych

Paweł Koźmiński

1.05.2019

Wprowadzenie

W celu przetestowania skuteczności różnych algorytmów analizy skupień, stworzyłem trzy zbiory testowe: dwa w $\mathbb{R}^2$ oraz jeden w $\mathbb{R}^3$. Zbiory przedstawiają różne ciekawe kształty, a sposób ich tworzenia został opisany poniżej.

Rysowanie wykresów

Benchmarkowe zbiory danych zostaną przedstawione za pomocą wykresów stworzonych przy pomocy biblioteki *Plotly*. Dla wygodnego tworzenia owych ilustracji w naszym przypadku, stworzona została pomocnicza funkcja `maluj()`. Aby w ładny sposób przedstawić grafiki, można skorzystać z biblioteki *Webshot* bądź wstawić wcześniej przygotowane grafiki. Ze względów estetycznych, skorzystałem z drugiej opcji.

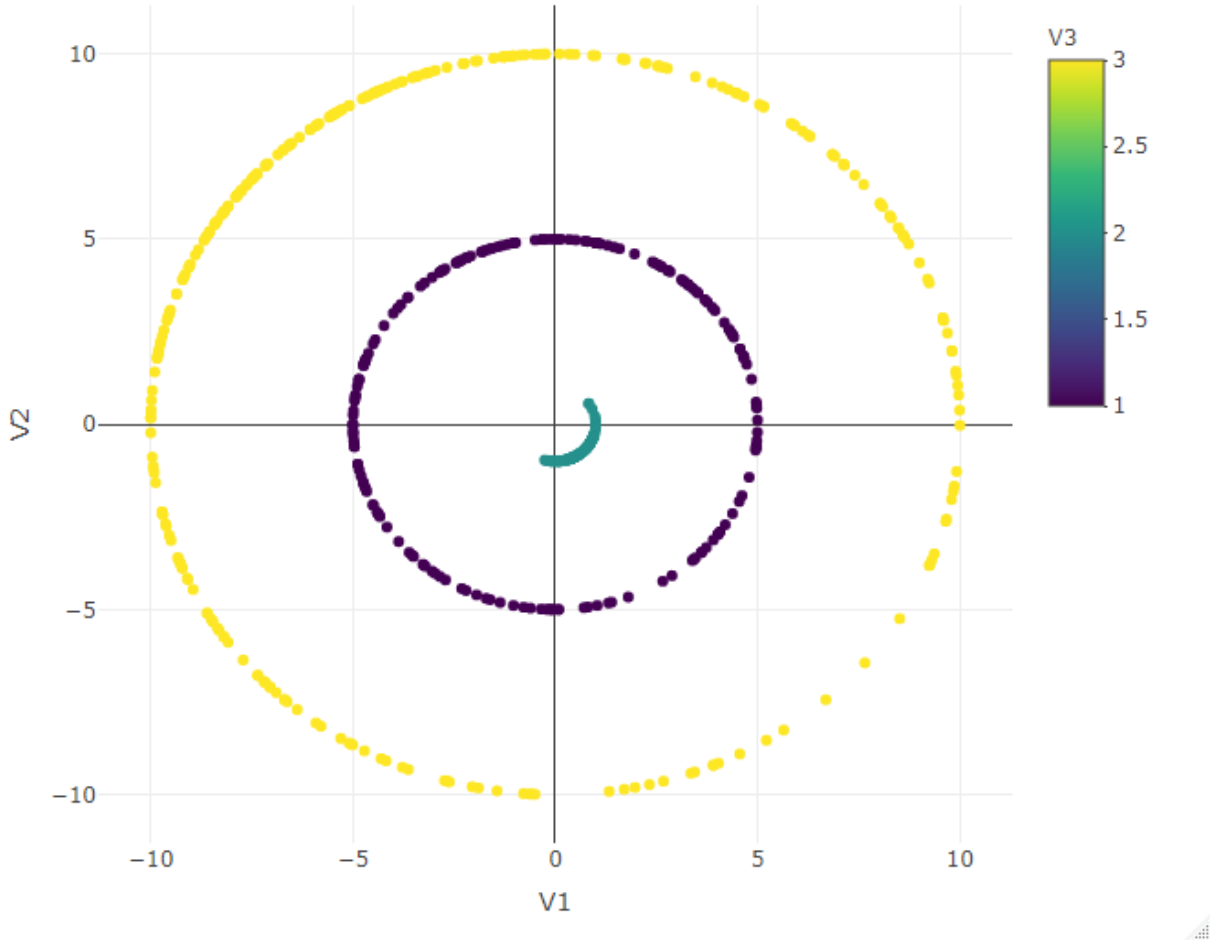


Figure 1: Zbiór nr 1

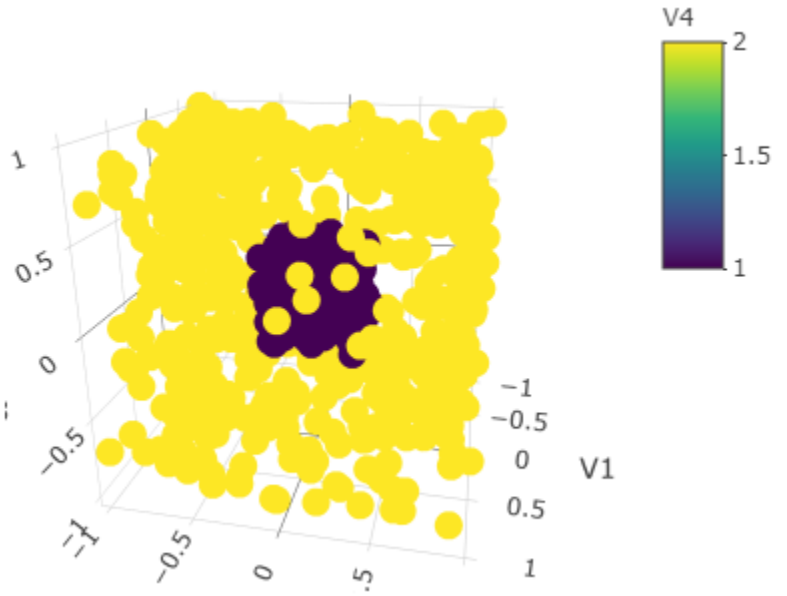
Własne zbiory benchmarkowe

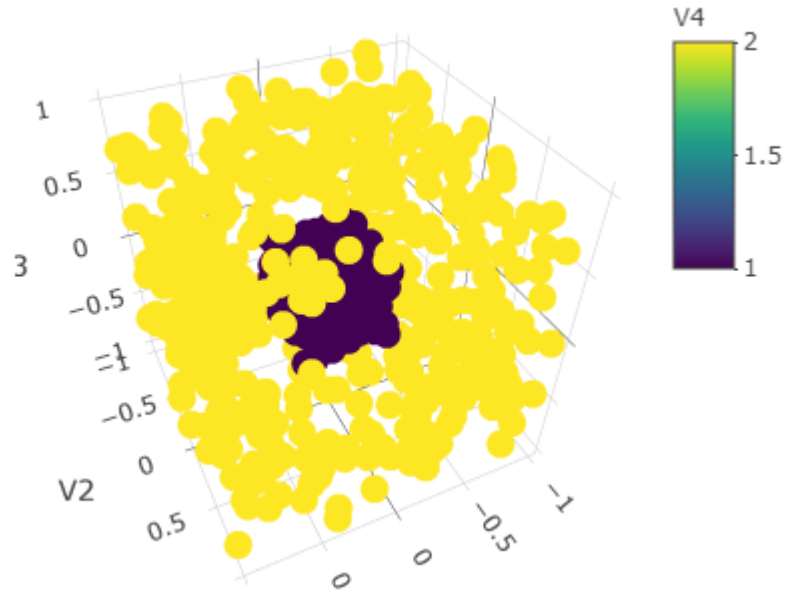
Pierwszy zbiór

Przy stworzeniu pierwszego zbioru danych skorzystałem ze wskazówek znalezionych na stronie internetowej `www.r-bloggers.com` oraz biblioteki *movMF* - z pomocą której można skorzystać z rozkładu von Mises-Fischera. Dzięki temu stworzone zostały kształty przypominające okręgi w układzie współrzędncyh. Zbiór zawiera 600 punktów w 2 wymiarach oraz 3 skupienia.

Drugi zbiór

Drugi zbiór benchmarkowy został stworzony w $\mathbb{R}^3$. Przedstawia sześcian wewnątrz większej bryły. Bryła ta także jest w kształcie sześcianu, jednak z wyciętą wewnątrz kulą. Zbiór zawiera około 800 wylosowanych punktów tworzących wyżej opisany kształt oraz 2 skupienia. W celu lepszej wizualizacji został przedstawiony za pomocą dwóch grafik.





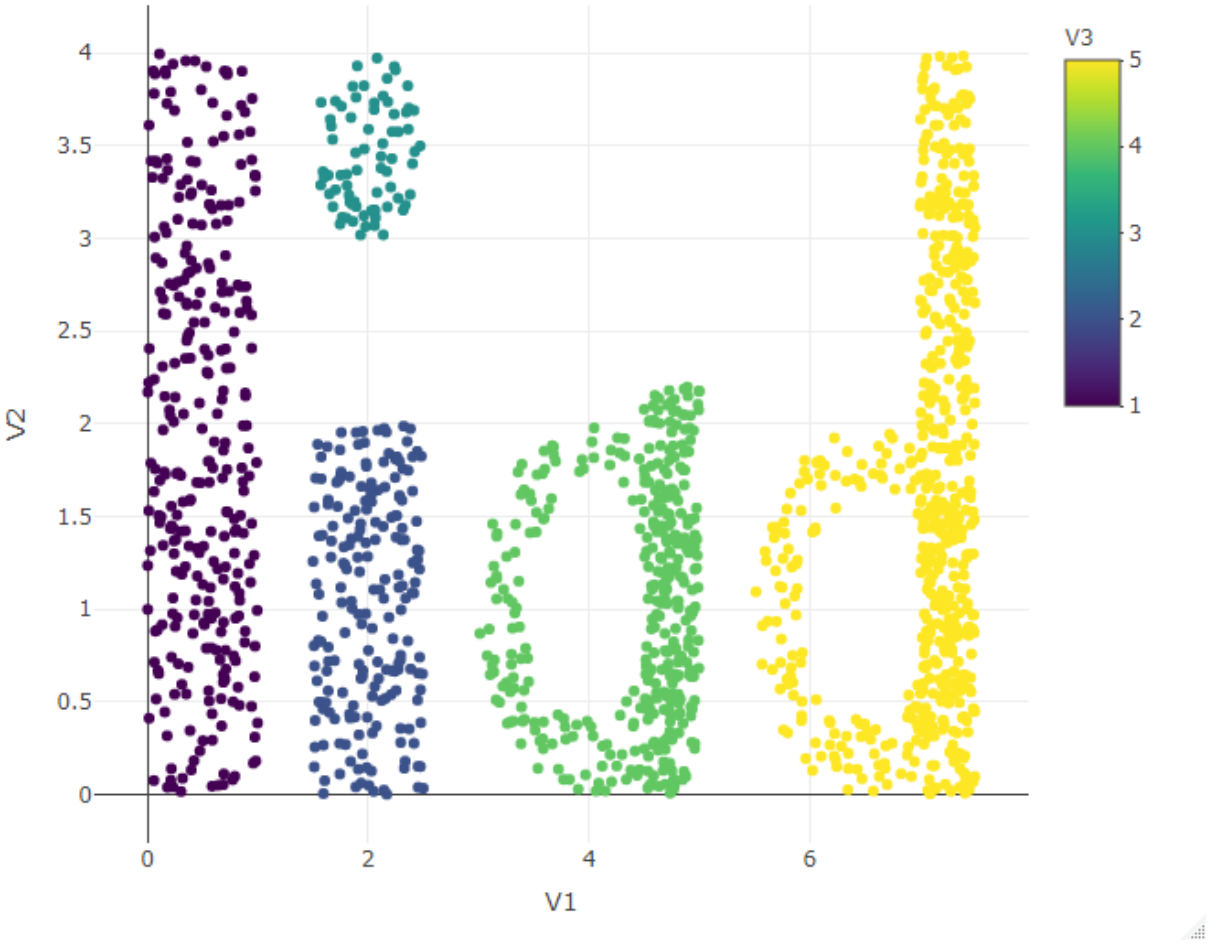


Figure 2: Trzeci zbiór danych

Trzeci zbiór

Ostatni zbiór danych jest puszczeniem oczka w stronę wszystkich osób związanych z naszym kierunkiem. Przy prezentacji jego w układzie współrzędnych ukazuje nam się napis “Iiad”. Trzeci zbiór benchmarkowy składa się z ok. 1550 dwuwymiarowych punktów w 5 skupieniach.

Sprawdzenie algorytmów analizy skupień

Zgodnie z treścią polecenia, przeprowadzimy teraz test róznych algorytmów na samodzielnie stworzonych testowych zbiorach danych. Skorzystałem z pomocy przygotowanej w tym celu funkcji `test_bench()`. Efektywność algorytmów sprawdzana była za pomocą indeksu Fowlkesa-Mallowsa oraz skorygowanego indeksu Randa.

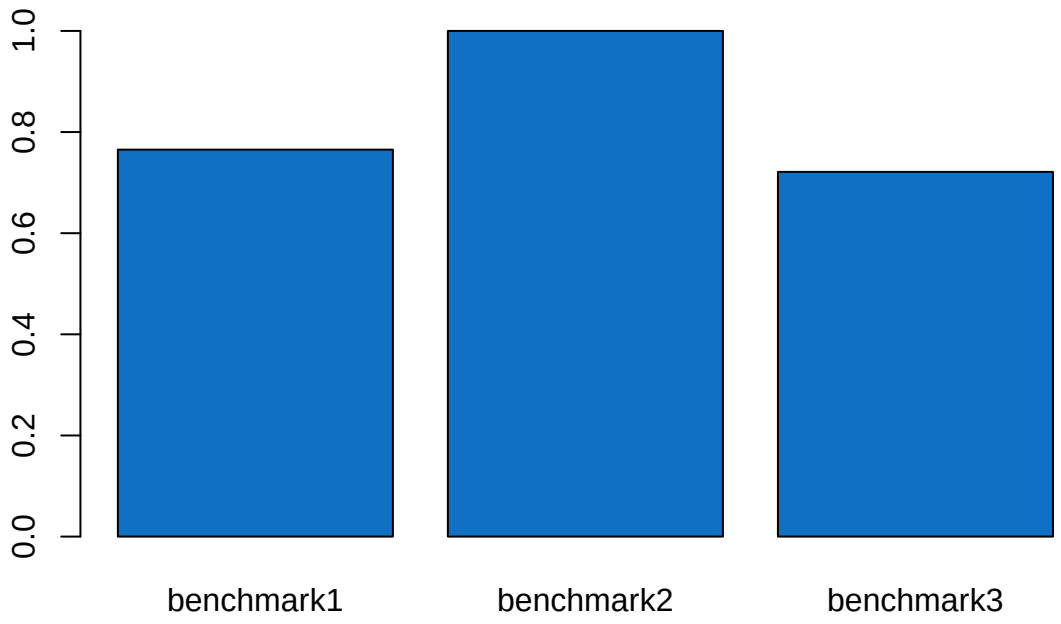
Sprawdzimy wpierw, czy standaryzacja danych poprawia skuteczność algorytmów.

	Średni indeks
przed standaryzacją	0.65373
po standaryzacji	0.62673

Jak widać - standaryzacja danych wpłynęła minimalnie niekorzystnie na skuteczność badanych metod analizy skupienia.
W takim razie sprawdzimy które sposoby były najskuteczniejsze na danych nieskalowanych. Poniżej przedstawiono dziesięć najlepszych algorytmów pod względem średnich współczynników czterech testów.

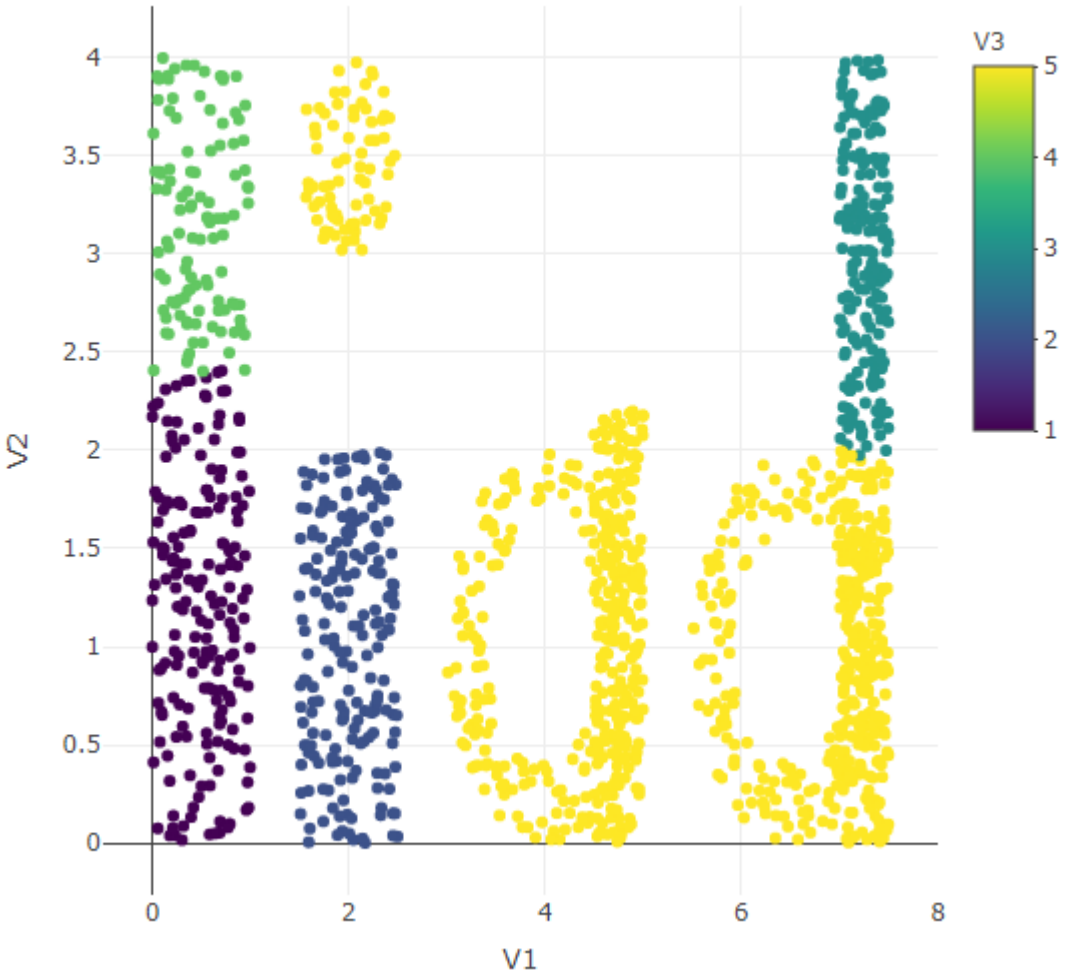
HSingle	Genie0.5	Genie0.6	Genie0.7	ASingle	Genie	Genie0.2	Genie0.4	own12	own15
1	1	1	1	1	0.98006	0.98006	0.98006	0.8815917	0.6445817

Algorytmy `Single` z rodziny `hclust()` i biblioteki `cluster` oraz `Genie` dla progów (*threshold*) 0.5, 0.6, 0.7 poradziy sobie z zadaniem najlepiej. Skuteczność ich działania wynosiła 100%. Kolejne miejsca zajęły inne wersje `genie`, a także własne implementacje algorytmu spektralnego, korzystającego odpowiednio z: 12 i 15 sąsiadów.
Sprawdźmy wyniki najlepszej wersji algorytmu spektralnego - `own12`:



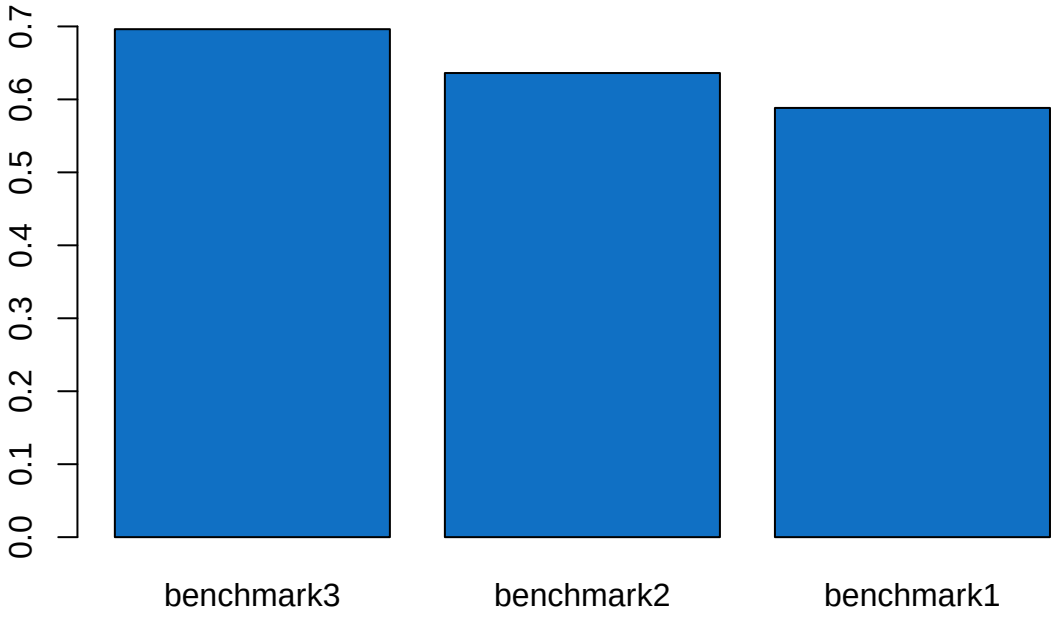
Jak możemy zauważyć, najgorzej sobie poradził ze zbiorem nr 3.

Oto wizualizacja efektów jego działania na owych danych:



Niestety, algorytm postanowił rozdzielić pierwszą literę `I` na dwa skupienia oraz połączyć literę “a”, dół “d” oraz kropkę nad “i”.

A które ze zbiorów okazały się najtrudniejsze do zbadania dla algorytmów?



Okazuje się, że całościowo, średnio algorytmy najlepiej radziły sobie z trzecim zbiorem danych, z którym nie najlepiej poradził sobie chociażby `own12`. Najgorzej wypadły trzy okręgi ze zbioru nr 1.