Dokumentacja końcowa MED

Adam Małkowski

10 stycznia 2018

1 Treść zadania

Implementacja algorytmu grupowania dla dużych zbiorów danych (np. CLARA, CLARANS). Wizualizacja wyników.

2 Założenia

1. Zaimplementowany zostanie algorytm CLARA, a pośrednio również algorytm PAM.
2. Implementacja odbędzie przy wykorzystaniu języka R.
3. W ramach projektu jako metryka odległości zostanie wybrana metryka Minkowskiego.
4. Algorytm CLARA zostanie zaimplementowany następująco:
   (a) Na wejściu przyjmuje zbiór danych - *data*, liczbę grup - n, rozmiar próbki losowej m, liczbę próbek losowych k.
   (b) Ze zbioru danych losowane jest k podzbiorów liczących po m elementów.
   (c) Dla każdego z wylosowanych podzbiorów uruchamiany jest algorytm PAM mający znaleźć n grup.
   (d) Oceniana jest jakość znalezionych grup na całym zbiorze danych *data* - jakość oceny wyraża funkcja będąca sumą po wszystkich grupach sum odległości pomiędzy elementami należącymi do grup, a medoidami reprezentującymi ich grupy.
   (e) Jako wynikowe grupowanie wybierane jest grupowanie posiadające najniższa wartość funkcji oceny.
5. Algorytm PAM zostanie zaimplementowany następująco:
   (a) Na wejściu przyjmuje zbiór danych - *data* oraz liczbę grup - n.
   (b) Losowane jest n elementów które zostaną medoidami.
   (c) Przeglądany jest cały zbiór danych - do każdego elementu przypisywana jest grupa reprezentowana przez medoid będący najbliżej danego elementu.
   (d) Porównywane są wszystkie pary medoid-niemedoid i analizowane jest, jak potencjalna ich zamiana, wpłynęłaby na jakość grupowania - jeżeli pozytywnie, jest dokonywana. W ramach tego punktu wpływ zamiany liczy się addytywnie analizując każdą trójkę - medoid, potencjalny nowy medoid, inny element. W ramach takiej trójki może wystąpić jeden z czterech przypadków oparatych na cechach:
   $u_{j,1}$ - najbliższy medoid elementu j,
   $u_{j,2}$ - drugi najbliższy medoid elementu j,
   $o(a, b)$ - odległość między elementami a i b,
   m_i - medoid o indeksie i
      i. $u_{j,1} < o(x_j, m_i)$ i $o(x_j, x_k) \geq u_{j,1}$ to zmiana wynosi 0
      ii. $u_{j,1} < o(x_j, m_i)$ i $o(x_j, x_k) < u_{j,1}$ to zmiana wynosi $o(x_j, x_k) - u_{j,1}$
      iii. $u_{j,1} = o(x_j, m_i)$ i $o(x_j, m_i) \leq u_{j,2}$ to zmiana wynosi $o(x_j, x_k) - u_{j,1}$
      iv. $u_{j,1} = o(x_j, m_i)$ i $o(x_j, m_i) > u_{j,2}$ to zmiana wynosi $u_{j,2} - u_{j,1}$

Jeżeli wartość sumaryczna jest mniejsza niż 0 dokonywana jest zamiana.

(e) Wynik stanowi ostatni stan grup.

6. Algorytm zostanie przetestowany na danych z http://archive.ics.uci.edu/ml. Kryterium wyboru zbioru będzie wysoka liczebność przykładów oraz przystosowanie do zadania grupowania.

3 Realizacja

1. Algorytmy PAM i Clara zostały zaimplementowane zgodnie z dokumentacją wstępną.

2. Implementacja wykorzystuje podstawowe pakiety środowiska R.

3. Dla przyśpieszenia obliczeń w metodzie PAM macierz odległości między punktami zostaje obliczona raz, na początku pracy metody, i jest wykorzystywana w czasie stałym - ponoszony jest tutaj koszt pamięciowy rzędu n^2.

4. Metoda *my_pam* przyjmuje argumenty:
- *data* - ramka danych wejściowych,
- n - liczba grup,
- *max_iter* - maksymalna liczba iteracji algorytmu (domyślnie 5),
- *minkowski_lvl* - wybór poziomu metryki Minkowskiego (domyślnie 2, euclidesowa).
Metoda zwraca listę której pierwszym elementem jest ramka danych z znalezionymi medoidami, a drugim elementem wektor grup.

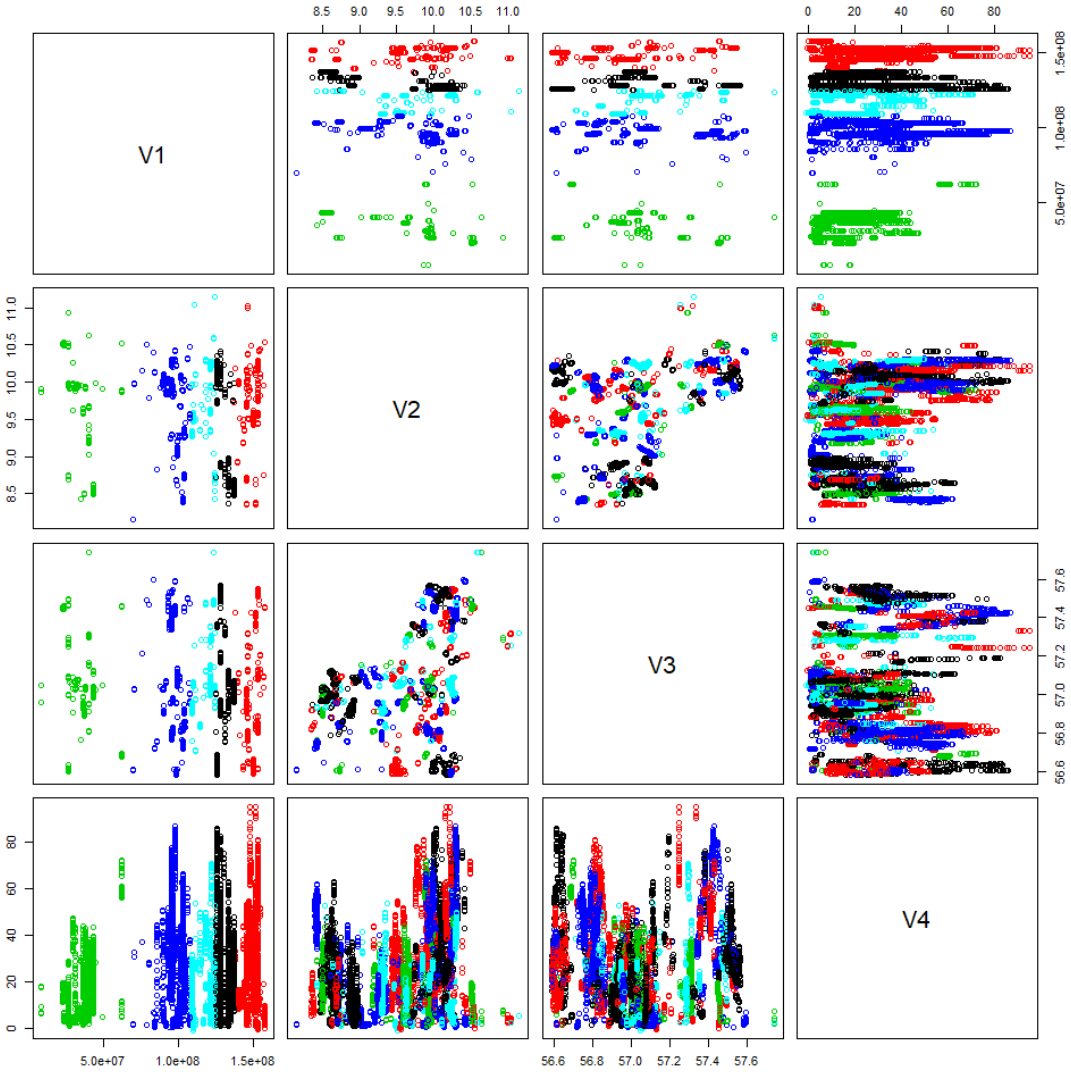
5. Metoda *my_clara* przyjmuje argumenty:
- *data* - ramka danych wejściowych,
- n - liczba grup,
- m - liczba elementów w próbce (domyślnie 100),
- k - liczba analizowanych podzbiorów danych,
- *max_iter* - maksymalna liczba iteracji algorytmu PAM, domyślnie 5,
- *minkowski_lvl* - wybór poziomu metryki Minkowskiego (domyślnie 2, euclidesowa).
Metoda zwraca listę której pierwszym elementem jest ramka danych z znalezionymi medoidami, a drugim elementem wektor grup.

6. Wizualizacja wyników odbywa się przy użyciu funkcji *plot* pakietu R. Kolory reprezentują grupy.

7. W ramach projektu powstał skrypt uruchamiający - należy posiadać zainstalowane pakiety clusterCrit oraz dplyr. Struktura argumentów wejściowych jest następująca :
nazwa pliku, liczba grup, rozmiar próbki (domyślnie = 100), liczba próbek (domyślnie = 10), maks iteracji PAM (domyślnie = 5), poziom metryki Minkowskiego (domyślnie = 2).

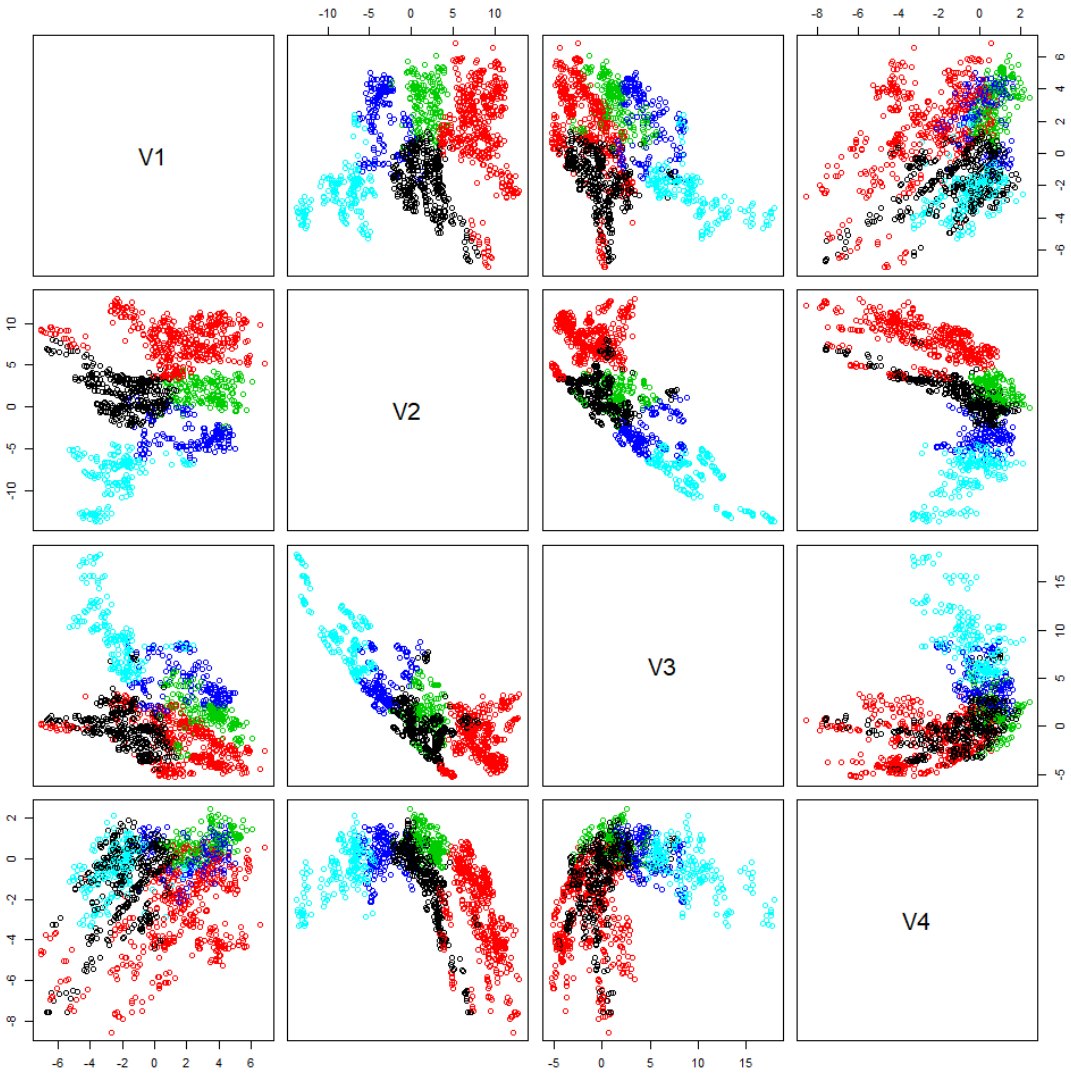
4 Przykładowe wyniki

1. Dla zbioru danych: https://archive.ics.uci.edu/ml/datasets/3D+Road+Network+(North+Jutland,+Denmark) oraz parametrów $n = 5$, $m = 100$, $k = 10$, $max\_iter = 10$, $minkowski\_lvl = 2$



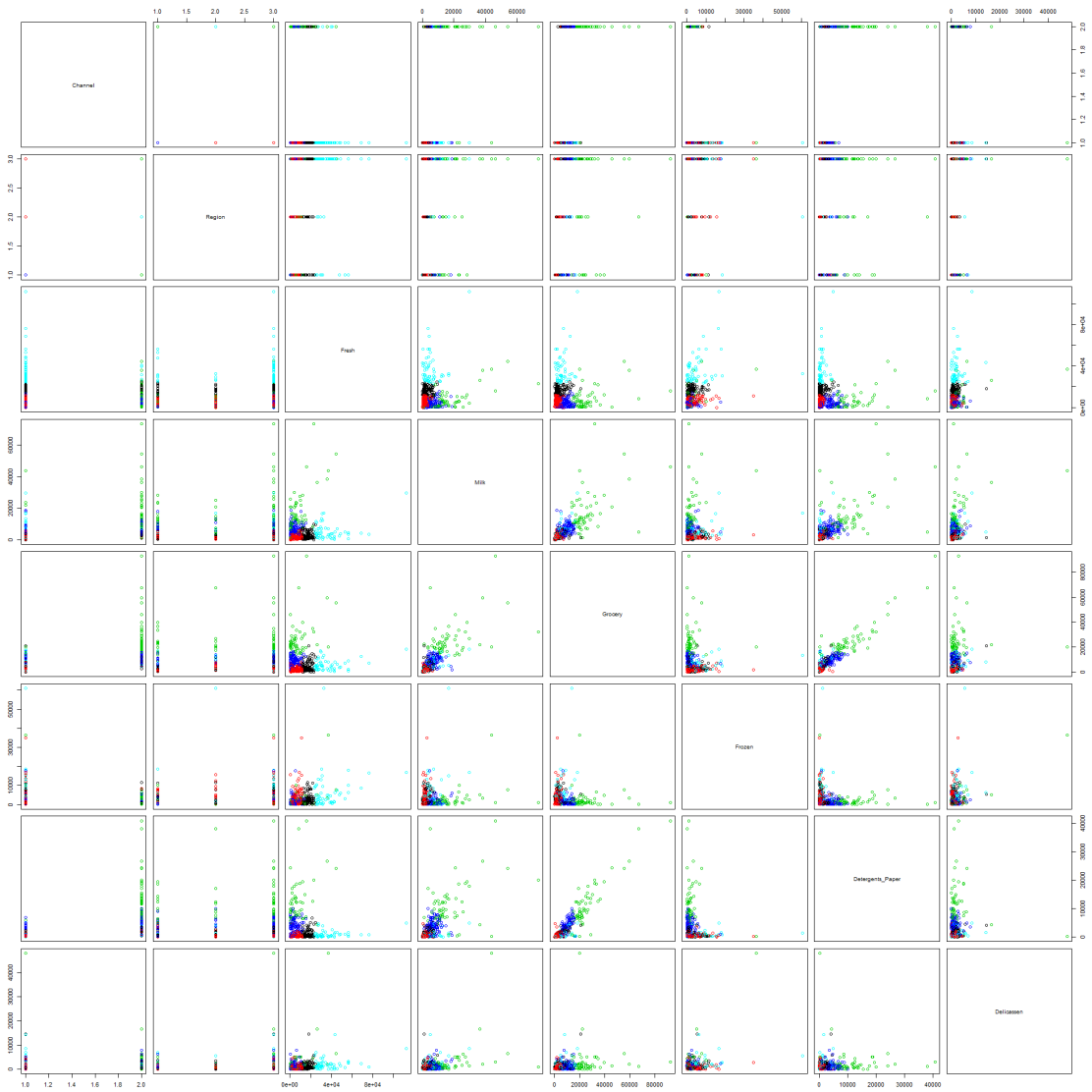
Wartości pierwszego atrybutu są o pare rzędów wielkości wyższe niż pozostałe, w związku z tym metryka euklidesowa uwzględnia praktycznie tylko go - jest to widoczne powyżej, wykresy uwzględniające parametr pierwszy są sensownie pogrupowane, w pozostałych ciężko dopatrzeć się prawidłowości.
Przykładowe oceny jakości grupowania:
Davies Bouldin Index - 0.3883059
Silhouette - 0.7448675
Dunn Index - 0.03023115

2. Dla zbioru danych: https://archive.ics.uci.edu/ml/datasets/banknote+authentication oraz parametrów $n = 5$, $m = 100$, $k = 10$, $max\_iter = 10$, $minkowski\_lvl = 2$



W przeciwieństwie do poprzedniego przypadku widać tutaj ładną separcję przykładów na większości wykresów.
Przykładowe oceny jakości grupowania:
Davies Bouldin Index - 0.9682127
Silhouette - 0.325289
Dunn Index - 0.01217318

3. Dla zbioru danych: https://archive.ics.uci.edu/ml/datasets/wholesale+customers oraz parametrów $n = 5$, $m = 100$, $k = 10$, $max\_iter = 10$, $minkowski\_lvl = 2$



Przykład prezentacji wyników z większą liczbą atrybutów. (Obraz w większej rozdzielczości załączony wraz z dokumentacją).
Przykładowe oceny jakości grupowania:
Davies Bouldin Index - 0.9415711
Silhouette - 0.1948544
Dunn Index - 0.010781