Algorytmy grupowania w R

Indukcyjne metody analizy danych

Maksym Telepchuk

Kwiecień 2020

1 Algorytmy klasteringu(grupowania)

Klastering jest procesem grupowania obiektów, rzeczywistych bądź abstrakcyjnych, w klasy, nazywane klastrami lub skupieniami, zgodnie z przyjęta funkcją podobieństwa.

W algorytmach klasteringu optymalizacyjno-iteracyjnych tworzony jest początkowy podział obiektów (zbiór klastrów k), a następnie, stosując technikę iteracyjnej realokacji obiektów pomiędzy klastrami, podział ten jest modyfikowany w taki sposób, aby uzyskać poprawę podziału zbioru obiektów na klastry.

W algorytmach optymalizacyjno-iteracyjnych jest bardzo istotnym zdefiniowanie ilości klastrów k oraz są one zależne od początkowej inicjalizacji. W praktyce, algorytm grupowania jest uruchamiany kilkakrotnie, dla różnych podziałów początkowych, a następnie, najlepszy z uzyskanych podziałów jest przyjmowany jako wynik procesu grupowania.

Tutaj zostały porównane dwa algorytmy : algorytmy **k-means** oraz **PAM**. Te algorytmy działają na wartościach numerycznych, więc nie jest potrzebne wykorzystywanie dyskretyzacji.

1.1 k-means

Podstawowa idea stojąca za algorytmem *k-means* jest to wybranie takiego podziału C na klastry $\{C_1, .., C_k\}$, który minimalizuje sumę węwnątrzklastrowego odchylenia ($wc(C_j)$) dla klastra j, które jest zdefiniowane wzorem:

$$wc(C_j) = \sum_{x_i \in C_j} d(x_i, \mu_j)^2$$

gdzie d jest zdefiniowaną miarą odległości (np. euklidesowa), μ_j jest średnią klastra i jest obliczana ze wzoru

$$\mu_j = \frac{1}{|C_j|} \sum x \quad , x \in C_j$$

Algorytm zajmuje się zminimalizowaniem sumy tych odchyleń

$$wc(C) = \sum_{j=1}^{k} wc(C_j) = \sum_{j=1}^{k} \sum_{x_i \in C_j}$$

Algorytm ma następujący opis:

1. Określ liczbę klastrów k, które mają zostać utworzone (parametr wejściowy).
2. Wybierz losowo k obiektów z zestawu danych jako początkowe środki klastrów.
3. Przypisz każdą obserwację do najbliższego środka na podstawie odległości euklidesowej między obiektem a środkiem.
4. Dla każdego z klastrów k zaktualizuj środek klastra, obliczając nowe średnie wartości wszystkich punktów danych w klastrze. Nowy środek μ_j jest wektorem długości p zawierającym średnie wszystkich atrybutów dla obserwacji w klastrze j (p jest liczbą atrybutów).
5. Powtarzaj kroki 3 i 4, aż przypisania klastrów przestaną się zmieniać lub osiągnięta zostanie maksymalna liczba iteracji. Domyślnie oprogramowanie R używa wartości 10 jako wartości domyślnej maksymalnej liczby iteracji.

   Implementacja algorytmu z pakietu *stats* dodatkowo umożliwia podanie maksymalnej liczby iteracji oraz ilości przeprowadzanych grupowań, z których zostanie wybrana najlepsza.

1.2 PAM

Algorytm *k-medoids*, w przeciwieństwu do poprzedniego algorytmu, przyjmuje środek klastra, który należy do tego klastra. Taki element jest nazywany medoidem. Oznacza to, że algorytm jest mniej wrażliwy na szum i wartości odstające, w porównaniu do *k-means*, ponieważ położenie medoidu nie wiele zależy od „outlierów" ze względu na to, że medoid musi być jednym z obiektów klastra. W kolejnych iteracjach medoidy klastrów są zamieniane obiektami nie będącymi medoidami, jeżeli ta operacja poprawi wynik grupowania (w tym przypadku funkcją kryterialną jest średnia odległość obiektów od środka klastra).

Popularną realizacją algorytmu *k-medoids* jest algorytm *PAM (and. Partitioning Around Medoids)*.

Struktura algorytmu wygląda następująco:

1. Inicjalizacja: „zachłannie” wybierz k punktów jako medoidy, aby zminimalizować koszt.

2. Powiąż każdy punkt danych z najbliższym medoidem.

3. Dopóki koszt konfiguracji zmniejsza się:

    1. Dla każdego medoidy m i dla każdego niemedoidalnego punktu danych o:

        1. Rozważ zamianę m i o i oblicz koszt zmiany.
        2. Jeśli koszt zmiany jest obecnie najlepszy, zapamiętaj tę kombinację m i o.

    2. Wykonaj najlepszą zamianę m_{best} i o_{best}, jeśli zmniejsza to funkcję kosztu. W przeciwnym razie algorytm zostanie zakończony.

2 Miary klasteryzacji

Do analizy wyników klastrowania zostały użyte miary DBI, Rand index, Dunn i Purity. DBI i Dunn ilustrują jakość klastrowania bez znanego rzeczywistego grupowania. Natomiast Rand index i Purity porównują wyniki klastrowania do rzeczywistych wartości grup.

1. DBI określa się wzorem

$$C = \frac{1}{K} \sum_{j=k}^{K} \max_{k=k'} \left(\frac{\delta_k + \delta_{k'}}{\Delta_{kk'}} \right)$$

    gdzie K - ilość klastrów, δ_k jest średnią odległością punktów od środka klastra k, $\Delta_{kk'}$ - odległość pomiędzy środkami klastrów k i k'

    Bardzo niski - daleke od siebie klastry, które mają niski „rozrzut”. Bardzo wysoki - bliskie klastry, które mają wysoki rozrzut.

2. Dunn określa się wzorem

$$C = \frac{d_{min}}{D_{max}}$$

    gdzie d_{min} - minimalna odległość pomiędzy dwoma punktami, należącymi do różnych klastrów, D_{max} - największy „diametr” z wyliczonych klastrów (odległość pomiędzy najbardziej oddalonymi punktami wewnątrz klastra).

    Niski - duże klastry blisko siebie, wysoki - małe klastry daleko od siebie.

3. Rand Index określa się wzorem

$$C = \frac{yy + nn}{\binom{N}{2}}$$

gdzie yy - ilość par należących do jednego klastra w obu porównywalnych podziałach, nn - ilość par należących do różnych klastrów w obu porównywalnych podziałach, N - ilość obiektów.

Niski - wyniki klastrowania istotnie się różnią, wysoki - wyniki klastrowania są podobne.

4. Puritry określa się wzorem

$$C = \frac{1}{N} \sum_k \max_j |w_k \cap c_j|$$

gdzie w_k jest wyliczonym klastrem, c_j - rzeczywistą klasą

Wysoki - klastrowanie jest bliskim do rzeczywistych klas. Niski - wyniki klastrowania nie odpowiadają rzeczywistym klasom.

3 Wyniki

W badaniu wydajności oba algorytmy zostały uruchomione dla klastrów od 2 do 15. Wszystkie atrybuty zostały znormalizowane. Dla k-means symulowano 30 przebiegów dla każdego k. Na wykresach są pokazane wartości miar dla każdego k. Linią przerywającą pokazano liczbę klas w zbiorze.

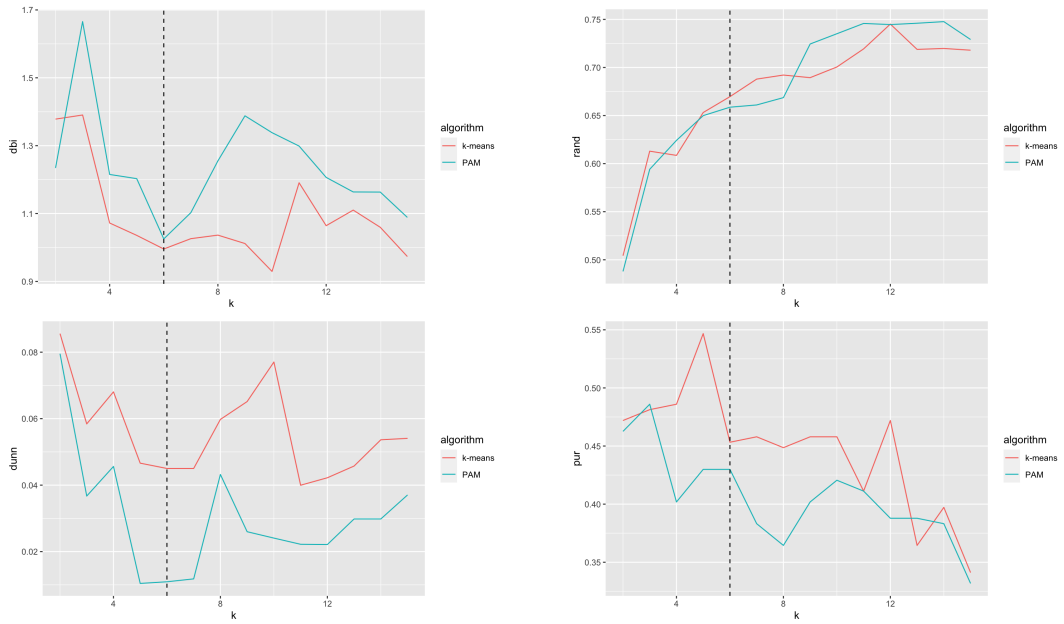
3.1 Glass

Dla zbioru danych **glass** wartość DBI różnią się w zależności od wyboru algorytmu. Algorytm *k-means* wykazuje lepsze rozdzielenie według tej miary. Porównując wartości tej miary dla obu algorytmów można dojść do wniosku, że wybór $k = 6$ (rzeczywista ilość klas) da względnie najniższe wyniki dla obu algorytmów (ok. 1). Wartość 1 oznacza, że „rozrzut klastrów" jest porównywalny do odległości pomiędzy klastrami.

Miara Dunn jest jest stosunkowo niska dla $k = 6$ w obu przypadkach. Tak niska wartość wykazuje, że klastry są o wiele większe od odległości pomiędzy nimi.

Miara Rand jest podobna w obu algorytmach oraz nie wskazuje na dobre wyniki klastrowania.

Miara Purity jest zdecydowanie lepsza dla algortymu k-means, ale nadal jest dosyć niska (ok. 45% w przypadku $k = 6$).



Rysunek 1: Clustering on glass dataset

Ze wszystkich miar wynika, że na podstawie grupowania tego zbioru danych ciężkim jest wykrycie klasy rzeczywistych. Oznacza to, że obiekty tego zbioru źle się grupują według swoich atrybutów. k-means w przypadku tego zbioru danych zachowuje się lepiej od PAM we wszystkich miarach.

3.2 Wine

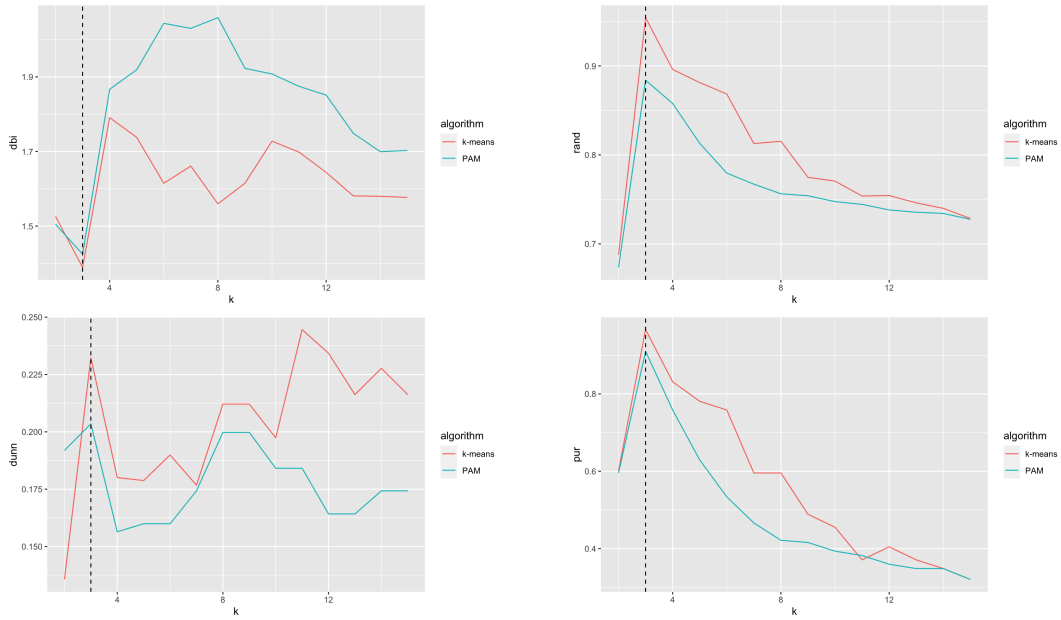
Dla zbioru danych **wine** wartość DBI również różnią się w zależności od wyboru algorytmu. Algorytm *k-means* wykazuje tutaj lepsze rozdzielenie według tej miary. Wybór $k = 3$ (rzeczywista ilość klas) da najniższe wyniki dla obu algorytmów (ok. 1.3). Taka wartość wskazuje na to, że „rozrzut klastrów" jest porównywalny do odległości pomiędzy klastrami.

Miara Dunn jest względnie wysoka dla $k = 3$ w obu przypadkach. Wartość wykazuje, że klastry są większe od odległości pomiędzy nimi.

Miara Rand jest podobna w obu algorytmach. Daje bardzo dobre wyniki dla $k = 3$ (ok. 90%).

Miara Purity też jest wysoka dla $k = 3$ (ok. 85%).

Z tych miar wynika, że na podstawie grupowania tego zbioru danych można dobrze klasyfikować dane ze zbioru. k-means w przypadku tego zbioru danych zachowuje się nieco lepiej od PAM we wszystkich miarach.



Rysunek 2: Clustering on wine dataset

3.3 Seeds

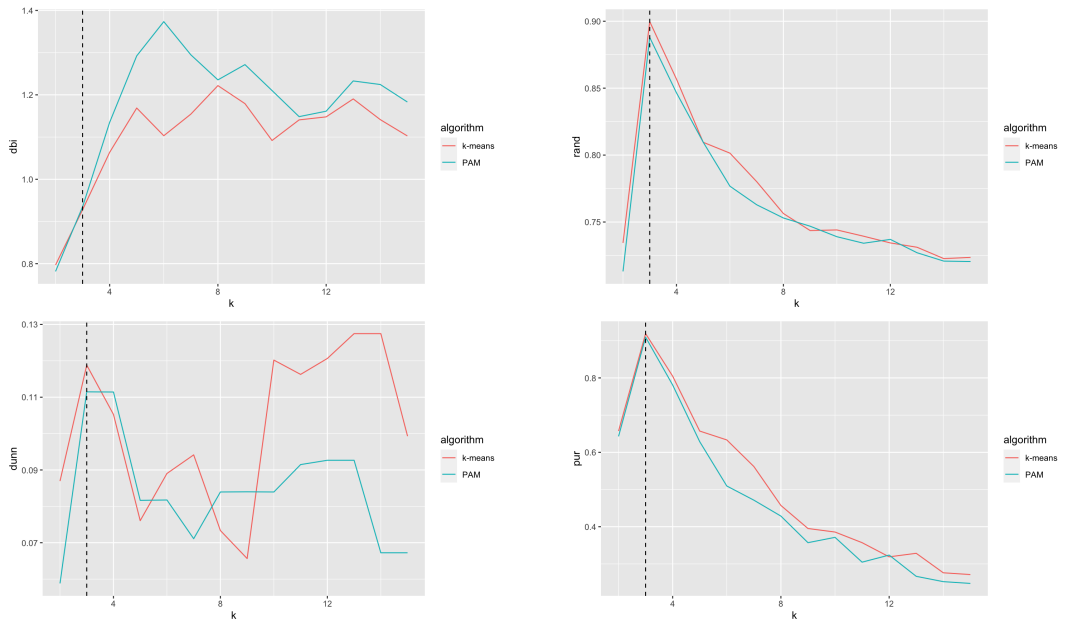
Dla zbioru danych **seeds** (ilość klasy = 3) zachowanie wartości miar są porównywalne do zbioru danych **wine**. Również są obserwowane duże klastry rozłożone blisko siebie.

Miara Purity jest wysoka dla $k = 3$ (ok. 85%).

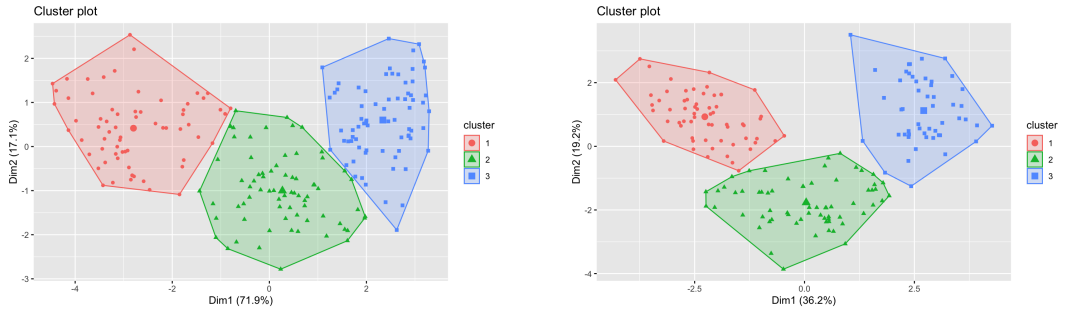
Z tych miar wynika, że na podstawie grupowania tego zbioru danych można dobrze klasyfikować dane ze zbioru. *k-means* w przypadku tego zbioru danych zachowuje się nieco lepiej od PAM we wszystkich miarach.

4 Wizualizacje

Obiekty ze zbiorów danych zostały zredukowane do 2 wymiarów przy użyciu metody PCA. Daje to możliwość wizualizacji klastrów. Na rys.4 znajdują się takie wizualizacje dla algorytmu *k-means* (30 inicjalizacji). Z wizualizacji wynika, że zbiory **wine** i **seeds** są lepiej klastrowane niż zbiór **glass**.

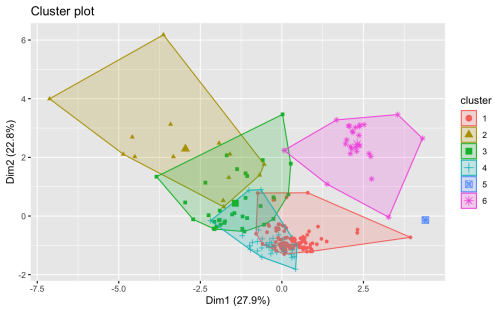


Rysunek 3: Clustering on seeds dataset



(a) seeds

(b) wine



(c) glass

Rysunek 4: Wizualizacja wyników klastrowania

Literatura

[1] http://wazniak.mimuw.edu.pl/images/8/86/ED-4.2-m11-1.01.pdf.

[2] https://www.datanovia.com/en/lessons/k-means-clustering-in-r-algorith-and-practical-examples/.

[3] https://www.datanovia.com/en/lessons/k-medoids-in-r-algorithm-and-practical-examples/.

[4] https://davetang.org/muse/2017/09/21/the-rand-index/.

[5] https://www.rdocumentation.org/packages/funtimes/versions/6.1/topics/purity.