PRACE NAUKOWE UNIWERSYTETU EKONOMICZNEGO WE WROCŁAWIU
Nr 47 2009
TAKSONOMIA 16
Klasyfikacja i analiza danych – teoria i zastosowania

**Jerzy Korzeniewski**
Uniwersytet Łódzki

# BADANIE EFEKTYWNOŚCI WYBRANYCH METOD GRUPOWANIA DANYCH NA ZBIORACH DANYCH ZE ŚWIATA REALNEGO

## 1. Wstęp

W publikacji [Guha, Rastogi, Shim 1998] autorzy algorytmu CURE zaproponowali algorytm przewyższający wszystkie dotychczas znane algorytmy grupowania danych pod względem szybkości pracy, wrażliwości na stopień rozmycia skupień i zdolności wykrywania skupień o kształcie niekoniecznie normalnym. Algorytm został porównany z takimi metodami, jak CLARA, CLARANS i BIRCH. Porównanie zostało przeprowadzone na kilku sztucznie wygenerowanych zbiorach danych z dwuwymiarowej przestrzeni euklidesowej. Podobnie [Karypis i in. 1999] zaproponowali algorytm CHAMELEON, który okazał się doskonały na sztucznych zbiorach danych z przestrzeni dwuwymiarowej. Wydaje się, że warto byłoby zbadać efektywność wymienionych nowych metod, tj. CURE i CHAMELEON, na zbiorach danych ze świata realnego. Niniejsza praca ma być próbą takiego badania.

## 2. Krótki przegląd metod grupowania obserwacji

Metody grupowania obserwacji można, z grubsza rzecz biorąc, podzielić na partycjonujące oraz hierarchiczne. Można również dokonać nieco dokładniejszego podziału na następujące grupy:

- metody partycjonujące (*partitioning methods*), np. *k*-średnich, rozmytych *c*-średnich;
- metody hierarchiczne (*hierarchical methods*), np. aglomeracja średniego łączenia, całkowitego połączenia, Warda itp.;
- metody gęstościowe (*density based methods*), np. algorytmy DBSCAN, OPTICS;
- metody oparte na siatce (*grid based methods*), np. algorytm STING;
- metody modelowe (*model based methods*), np. metody oparte na sieciach neuronowych;

- metody typu *ensemble* (*ensemble methods*) wykorzystujące wyniki wielu różnych metod grupowania.

W ostatnich latach największe uznanie zdobyły metody łączące cechy kilku, na ogół dwóch, rodzajów metod. Na przykład algorytmy BIRCH i CURE łączą cechy grupowania hierarchicznego i relokacyjnego, algorytmy CLIQUE i Wave-Cluster (por. [Kaufman, Rousseeuw 1990]) łączą cechy metod gęstościowych i opartych na siatce komórek. Algorytm CHAMELEON miał łączyć dobre cechy algorytmów CURE oraz ROCK i nie mieć ich słabości.

W celu ustanowienia punktu odniesienia dla badanych algorytmów CURE oraz CHAMELEON wybrano dwie metody, starsze o kilka lat od dwóch wspomnianych, które można określić mianem tradycyjnych. Są to algorytmy CLARA oraz CLARANS. Podstawą działania tych algorytmów jest algorytm PAM [Kauffman, Rousseeuw 1990]. Algorytmy CLARA i CLARANS mają dość dobrą opinię w kwestii zastosowań do grupowania obserwacji ze zbiorów danych ze świata realnego.

## 3. Charakterystyka metod wybranych do badania

Algorytmy CLARA i CLARANS są oparte na algorytmie PAM (*Partitioning Around Medoids*), którego ideą jest przypisanie każdej obserwacji do skupienia reprezentowanego przez medoidę najbliższą (ze wszystkich skupień) tej obserwacji. Algorytm rozpoczyna od losowego wyboru $k$ medoid ($k$ jest liczbą skupień). Oznaczmy wybrane medoidy symbolami $O_i \quad i = 1,...,k$, zaś obserwacje, które nie są medoidami, symbolami $O_h$. Algorytm działa iteracyjnie według następującego schematu.

1. Obliczyć koszty $TC_{ih}$ zamiany medoidy $O_i$ na obserwację $O_h$, która nie jest medoidą dla wszystkich par $O_i \; O_h$.

2. Wybrać tę parę $O_i \; O_h$, która ma najmniejszy koszt $TC_{ih}$. Jeśli ten najmniejszy koszt jest ujemny, to zamienić $O_i$ na $O_h$ i wrócić do punktu 1.

3. Jeśli najmniejszy znaleziony koszt jest nieujemny, to przypisać każdą obserwację do najbliższej medoidy i zakończyć działanie.

Koszt jest rozumiany w sensie dążenia do zminimalizowania sumy odległości wszystkich obserwacji od medoid reprezentujących skupienia. W związku z tym koszt obliczamy, sumując zmiany odległości wynikające z zamiany $O_i$ na $O_h$ po wszystkich obserwacjach $O_j$, które nie są medoidami, czyli

$$TC_{ih} = \sum_j C_{ijh} \; . \tag{1}$$

Koszt $C_{ijh}$ jest składnikiem kosztu całkowitego wynikającym z zamiany, ale tylko w odniesieniu do obserwacji $O_j$. Mogą wystąpić cztery przypadki i w zależności od nich obliczamy koszt $C_{ijh}$ w następujący sposób.

Przypadek 1. Załóżmy, że $O_j$ przed zamianą należy do skupienia reprezentowanego przez $O_i$ oraz że $O_j$ jest bardziej podobne do $O_{j,2}$ niż $O_h$ tzn. $d(O_j, O_h) \geq d(O_j, O_{j,2})$, gdzie $O_{j,2}$ oznacza drugą medoidę najbardziej podobną do $O_j$. Wówczas, gdy $O_i$ zamienimy na $O_h$, $O_j$ będzie należało do skupienia reprezentowanego przez $O_{j,2}$. Wobec tego

$$C_{ijh} = d(O_j, O_{j,2}) - d(O_j, O_i). \tag{2}$$

Przypadek 2. Załóżmy, że $O_j$ przed zamianą również należy do skupienia reprezentowanego przez $O_i$ oraz że tym razem $d(O_j, O_h) < d(O_j, O_{j,2})$. Wówczas

$$C_{ijh} = d(O_j, O_h) - d(O_j, O_i). \tag{3}$$

Przypadek 3. Załóżmy, że $O_j$ przed zamianą należy do skupienia reprezentowanego przez $O_{j,2}$ oraz że $O_j$ jest bardziej podobne do $O_{j,2}$ niż $O_h$. Wówczas, gdy $O_i$ zamienimy na $O_h$, $O_j$ będzie należało do tego samego skupienia reprezentowanego przez $O_{j,2}$. Wobec tego

$$C_{ijh} = 0. \tag{4}$$

Przypadek 4. Załóżmy, że $O_j$ przed zamianą należy do skupienia reprezentowanego przez $O_{j,2}$ oraz że $O_j$ jest mniej podobne do $O_{j,2}$ niż $O_h$. Wówczas, gdy $O_i$ zamienimy na $O_h$, $O_j$ będzie należało do skupienia reprezentowanego przez $O_{j,2}$. Wobec tego

$$C_{ijh} = d(O_j, O_h) - d(O_j, O_{j,2}). \tag{5}$$

Tak zdefiniowany algorytm działa dobrze dla małych zbiorów, np. o liczbie obserwacji mniejszej od 100. Gdy zbiory są większe, wówczas ci sami autorzy [Kauffman, Rousseeuw 1990] opracowali algorytm CLARA, który zasadniczo rzecz biorąc, jest algorytmem PAM zastosowanym do prób wylosowanych ze zbioru danych. Działa on według następującego schematu.

**Algorytm CLARA**

1. Powtórzyć pięciokrotnie następujące kroki.
2. Wylosować próbę o liczebności $40 + 2k$ i zastosować algorytm PAM do znalezienia $k$ medoid w tej próbie.
3. Przypisać każdą obserwację do najbliższej jej medoidy.
4. Znaleźć średnią odległość obserwacji od ich medoid.
5. Zmienić przypisanie obserwacji na nowe (w następnej z pięciu iteracji), jeśli średnia odległość z punktu 4 jest mniejsza od najmniejszej z dotychczasowych.

Oczywiście tak działający algorytm nie jest jednoznaczny, pozostawia pewien chaos przypisań obserwacji do skupień wynikający z częściowej losowości, ale spisuje się dość dobrze (por. [Ng, Han 1994]).

Poprawioną wersją CLARY jest algorytm CLARANS, którego idea polega na tym, że zaczynając podobnie od dowolnie wybranych medoid, w każdym kroku wymienia jedną z medoid na dowolnie wybraną obserwację (oczywiście, gdy koszt takiego grupowania jest niższy). Takiej ewentualnej zamiany próbuje dokonać wiele razy (*maxneighbor*) i kończy na znalezieniu lokalnego minimum kosztu. Takie lokalne przeszukiwania powtarza kilka razy (*numlocal*). Dokładniej działanie algorytmu można ująć w następującym schemacie.

**Algorytm CLARANS**

1. Połóż $i = 1$ oraz $mincost = \infty$.
2. Wybierz dowolnie $k$ medoid.
3. Połóż $j = 1$.
4. Dowolnie wybraną medoidę zamień na dowolnie wybraną obserwację niebędącą medoidą. Oblicz różnicę kosztów zgodnie ze wzorem (1).
5. Jeśli nowy koszt jest niższy, to wróć do punktu 3, zachowując nowy wybór.
6. Jeśli nowy koszt jest wyższy, to zwiększ $j$ o 1 i idź do punktu 4, zachowując stary wybór, pod warunkiem że $j < maxneighbor$.
7. Gdy $j > maxneighbor$, porównaj koszt aktualnego wyboru medoid z *mincost*. Gdy koszt aktualnego wyboru jest mniejszy, połóż *mincost* równe nowemu kosztowi.
8. Zwiększ $i$ o 1 i jeśli $i <= numlocal$, idź do punktu 2. W przeciwnym razie przypisz obserwacje do najbliższej medoidy i zakończ działanie.

Autorzy tego algorytmu [Ng, Han 1994] polecają losować obserwacje do ewentualnej zamiany nie więcej niż $maxneighbor = \max\{250;\ 1{,}25\%k(n\text{-}k)\}$ razy ($n$ − liczba obserwacji w zbiorze danych, $k$ – liczba skupień), zaś wyszukiwanie lokalnego minimum kosztu nie więcej niż $numlocal = 2$ razy. Taką wersję algorytmu zastosowano w niniejszej pracy.

**Algorytm CURE**

Ogólnie rzecz biorąc, algorytm ten [Guha, Rastogi R., Shim 1998] najłatwiej wyjaśnić, porównując go do algorytmu aglomeracyjnego najbliższego połączenia. Różnice są następujące:

- algorytm CURE przy ustalaniu dwóch najbliższych skupień w każdym kroku aglomeracyjnym bierze pod uwagę tylko ustaloną liczbę *c* reprezentantów każdego skupienia;
- reprezentanci każdego skupienia są „ściągnięci" (*shrunk*) w kierunku średniej ze wszystkich reprezentantów o czynnik $\alpha \in (0,1)$ w następującym sensie:

$$x_{shrunk} = x + \alpha(\bar{x} - x). \tag{6}$$

Wybieranie reprezentantów każdego skupienia ma za zadanie skrócić czas pracy algorytmu, zaś ściąganie obserwacji „do środka" ma za zadanie uodpornić algorytm na wadę łańcucha często psującą algorytmy aglomeracyjne. Reprezentanci są tak wybierani, by dobrze reprezentowali całe skupienie. Zastosowano prostą metodę wyboru *c* dobrze rozproszonych obserwacji. Można je wybierać sekwencyjnie w następujący sposób: pierwszą obserwację wybieramy dowolnie, drugą – tak, by była z tego samego skupienia i jak najbardziej oddalona od pierwszej, trzecią tak, by była z tego samego skupienia i by była jak najbardziej oddalona od najbliższej z dwóch pierwszych itd. Przy łączeniu dwóch skupień, z których każde jest już reprezentowane przez *c* obserwacji, można procedurę nieco uprościć, wybierając *c* dobrze rozproszonych obserwacji spośród już określonych 2*c* obserwacji. Autorzy twierdzą, że algorytm powinien dać dobre rezultaty na każdym zbiorze dla $c = 10$ oraz $\alpha = 0{,}5$. Do w miarę szybkiego działania tego algorytmu konieczne jest wykorzystanie drzewa *k-d,* ponieważ trzeba wielokrotnie przeszukiwać zbiór danych w celu wyszukiwania najbliższych sąsiadów.

**Algorytm CHAMELEON**

Ideę tego algorytmu najłatwiej przedstawić, odwołując się do teorii grafów [Karypis, Eui-Hong, Kumar 1999]. Najpierw dla całego zbioru danych należy skonstruować graf *k*-najbliższego sąsiada. Wierzchołkami grafu są obserwacje, zaś krawędziami − odległości pomiędzy obserwacjami. Następnie należy podzielić graf na „małe wstępne skupienia". W kolejnym kroku algorytm aglomeruje małe skupienia, posługując się ciekawą miarą podobieństwa, która ma za zadanie uwzględniać zarówno odległość między skupieniami, jak i spójność pary skupień. Ostateczną miarą podobieństwa pomiędzy skupieniami $C_i$ i $C_j$ jest wyrażenie

$$RI(C_i, C_j) \cdot RC^2(C_i, C_j), \tag{7}$$

gdzie $RI(C_i, C_j)$ jest miarą spójności obu skupień (*relative inter-connectivity*) zaś $RC(C_i, C_j)$ jest miarą odległości pomiędzy skupieniami (*relative closeness*). Obie miary obliczane są z następujących wzorów:

$$RI\left(C_i, C_j\right) = \frac{\left|EC_{\{C_i,C_j\}}\right|}{\frac{\left|EC_{C_i}\right| + \left|EC_{C_j}\right|}{2}}, \tag{8}$$

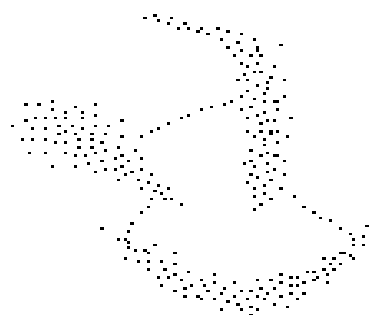
$$RC\left(C_i, C_j\right) = \frac{\overline{S}_{EC_{\{C_i,C_j\}}}}{\frac{\left|C_i\right|}{\left|C_i\right| + \left|C_j\right|}\overline{S}_{EC_{C_i}} + \frac{\left|C_j\right|}{\left|C_i\right| + \left|C_j\right|}\overline{S}_{EC_{C_j}}}, \tag{9}$$

gdzie $EC_{\{C_i,C_j\}}$ oznacza sumę wag (*edge cut*) krawędzi łączących wierzchołki (obserwacje) skupienia $C_i$ z wierzchołkami skupienia $C_j$, $EC_{C_i}$ oznacza sumę wag krawędzi dzielących graf *i*-tego skupienia na dwie w przybliżeniu równe części, $\overline{S}_{EC_{\{C_i,C_j\}}}$ oznacza średnią wagę krawędzi łączących wierzchołki skupienia $C_i$ z wierzchołkami skupienia $C_j$, zaś $\overline{S}_{EC_{C_i}}$ jest średnią wagą krawędzi dzielących graf *i*-tego skupienia na dwie w przybliżeniu równe części.

Tak sformułowany algorytm jest dość ogólny i pozostawia dużą dowolność w wyborze jego konkretnej wersji. Po pierwsze należy wybrać parametr *k* przy konstruowaniu grafu *k*-najbliższego sąsiada. Nie ma co do tego żadnych konkretnych reguł, powinna to być liczba mała w stosunku do liczebności zbioru danych. We wszystkich przykładach analizowanych w tej pracy przyjęto *k* = 5. Po drugie należy określić rozmiar małych wstępnych skupień, na które dzielony jest graf całego zbioru danych. Autorzy polecają dowolną wartość z przedziału od 1 do 5% liczebności zbioru danych. W tej pracy przyjęto dla wszystkich zbiorów wartość równą 3%**n*. Jednak największa dowolność istnieje przy wyborze metody podziału grafu na małe skupienia lub na dwie w przybliżeniu równe części (stosujemy tę sama metodę, przy czym w pierwszym przypadku stosujemy ją wielokrotnie). Istnieje wiele metod, które wykonują zadanie bisekcji grafu – większość z nich w dość dużym stopniu obarczona jest losowością [Karypis, Kumar 1998]. W tej pracy zastosowano metodę bisekcji polecaną przez autorów (por. [Karypis, Kumar 1998]), polegającą na tym, że kilkadziesiąt (przyjęto 20) razy losowo wybieramy połowę grafu i zapamiętujemy wybór, który dał najmniejszą wagę interesujących nas krawędzi (tj. tych, które mają swe końce w dwóch różnych połowach grafu). Następnie wymieniamy pojedyncze obserwacje, przenosząc je z jednej połowy grafu do drugiej (w następnym kroku odwrotnie), wybierając obserwacje tak, by dawały jak największą zmianę zmniejszającą wagę krawędzi mających swe końce w dwóch różnych połowach grafu. Wymianę kontynuujemy dopóty, dopóki waga krawędzi przestanie się zmniejszać.

## 4. Eksperyment badawczy

Na wstępie w celu wizualnej weryfikacji poprawności działania programów zastosowano wszystkie cztery algorytmy (CLARA, CLARANS, CURE, CHAMELEON) do znalezienia skupień zbioru przedstawionego na rys. 1. Algorytmy CLARA i CLARANS spisały się w miarę poprawnie, urywając kilka punktów z końca jednego skupienia i przypisując je do innego skupienia. Podobnie spisał się algorytm CURE, myląc się w przypadku kilkunastu punktów. Natomiast algorytm CHAMELEON działał bezbłędnie do czasu utworzenia około 8-9 skupień. Później okazało się, że faworyzuje łączenie skupień większych albo przyłączanie skupienia mniejszego do większego. Takie zachowanie wynika ze wzoru (9). Ważona średnia spójność skupień z mianownika tego wzoru daje większe znaczenie spójności skupienia większego.



Rys. 1. Przykładowy zbiór złożony z 281 punktów z przestrzenie dwuwymiarowej
Źródło: opracowanie własne.

W celu ocenienia porównywanych metod na realnych zbiorach danych wybrano do badania kilka zbiorów danych, których obserwacje zostały pogrupowane w zadaną liczbę skupień w sposób, do którego można mieć zaufanie, ponieważ został zaakceptowany przez specjalistów z danej dziedziny. Zgodność tych grupowań z grupowaniami uzyskanymi przez badane algorytmy stanowiła kryterium oceny algorytmów. Popularny zbiór *Iris* z bazy danych UCI Machine Learning Repository składający się ze 150 obserwacji opisanych przez 4 zmienne powinien zostać podzielony na 3 skupienia − każde powinno zawierać po 50 obserwacji. Zbiór *Invest* z pracy [Atkinson, Riani, Cerioli 2004] składający się ze 103 obserwacji (funduszy finansowych) opisanych przez 3 zmienne powinien zostać podzielony na 2 skupienia: pierwsze zawierające 56 obserwacji oraz drugie skupienie zawierające 47 obserwacji. Zbiór *Diabet* z pracy [Atkinson, Riani, Cerioli 2004] składający się ze 145 obserwacji (chorych pacjentów) opisanych przez 3 zmienne powinien zostać podzielony na 3 skupienia: pierwsze zawierające 76 obserwacji, drugie składające się z 36 obserwacji i trzecie zawierające 33 obserwacje.

Wszystkie zmienne z tych zbiorów zostały znormalizowane oddzielnie względem każdej zmiennej przez odjęcie od wartości *j*-tej zmiennej dla *i*-tej obserwacji średniej arytmetycznej *j*-tej zmiennej i podzielenie przez odchylenie średnie tej zmiennej. Taka normalizacja uważana jest za odporniejszą od dzielenia przez odchylenie standardowe (por. [Kaufman, Rousseeuw 1990]).

Tabela 1. Efektywność grupowania obserwacji porównywanych algorytmów

| Nazwa zbioru danych | Odsetek błędnych klasyfikacji | | | |
|---|---|---|---|---|
| | CLARA | CLARANS | CURE | CHAMELEON |
| Iris | 32 | 31 | 35 | 50 |
| Diabet | 27 | 33 | 28 | 34 |
| Invest | 0 | 0 | 0 | 10 |

Źródło: obliczenia własne.

Tabela 1 zawiera odsetki błędnie przypisanych obserwacji. Algorytm CHAMELEON jest dość wrażliwy na losowość, którą jest obarczony, i odsetki błędnych klasyfikacji są średnimi arytmetycznymi z 5 zastosowań do każdego zbioru.

## 5. Wnioski

Algorytm CURE nie okazał się lepszy od swoich starszych konkurentów. Algorytm CHAMELEON poradził sobie w miarę przyzwoicie jedynie w przypadku trzeciego zbioru. W pozostałych dwóch przypadkach uwidoczniła się wada wspomniana przy grupowaniu obserwacji z przykładowego zbioru punktów płaszczyzny. CHAMELEON działał bezbłędnie do momentu utworzenia 7-8 skupień, potem większe skupienia wchłaniały mniejsze.

## Literatura

Atkinson A., Riani M., Cerioli A. (2004), *Exploring multivariate data with the forward search,* Springer-Verlag.

Gordon A.D. (1999), *Classification,* Chapman & Hall.

Guha S., Rastogi R., Shim K. (1998), *CURE: an efficient clustering algorithm for large databases*, Proceedings of ACM SIGMOD International Conference on Management of Data.

Karypis G., Eui-Hong H., Kumar V. (1999), *CHAMELEON: a hierarchical clustering algorithm using dynamic modeling,* IEEE Computer.

Karypis G., Kumar V. (1998), *Multilevel k-way partitioning scheme for irregular graphs*, "Journal of Parallel and Distributed Computing".

Kaufman L., Rousseeuw P.J. (1990), *Finding groups in data: an introduction to cluster analysis,* John Wiley&Sons.

Ng R., Han J. (1994), *Efficient and effective clustering methods for spatial data mining,* Proceedings of the 20th VLDB Conference, Chile.

# INVESTIGATING THE EFFICIENCY OF SELECTED DATA GROUPING METHODS ON REAL WORLD DATA SETS

## Summary

In their paper [Guha et al. 1998], the authors of CURE developed an algorithm that beats all so far known data grouping algorithms with respect to speed, sensitivity to outliers and capacity to find non-normal clusters. The algorithm was compared with CLARA, CLARANS and BIRCH. The comparison was carried out on a couple of artificial data sets from two-dimensional Euclidean space. In a similar way Karypis et al. [Karypis et al. 2000] proposed CHAMELEON and checked its performance on sets from two-dimensional Euclidean space. It seems worthy to investigate the performance of these two new methods on real world data sets. This is the aim of this paper.