Politechnika Wrocławska

Wydział Informatyki i Zarządzania
kierunek studiów: Informatyka
specjalność: Inżynieria Oprogramowania

Praca dyplomowa - magisterska

Algorytmy ewolucyjne w zadaniu klasteryzacji.

Jarosław Wojtasik

słowa kluczowe:
Klasteryzacja
Obliczenia ewolucyjne
Data mining

krótkie streszczenie:
1 linia
2 linia
3 linia
4 linia
5 linia
6 linia

Promotor:	.....	.....	.....
	imię i nazwisko	*ocena*	*podpis*

*Do celów archiwalnych pracę dyplomową zakwalifikowano do:**
a) kategorii A (akta wieczyste)
b) kategorii BE 50 (po 50 latach podlegające ekspertyzie)
** niepotrzebne skreślić*

Pieczątka instytutu, w którym student wykonywał pracę

Wrocław 2011

Spis treści

Algorytmy ewolucyjne w zadaniu klasteryzacji.

Streszczenie

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Donec sapien ipsum, fermentum eu malesuada ac, malesuada sed magna. Fusce mollis tempor eros, sed feugiat ante auctor ut. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed tincidunt, enim vel pulvinar porttitor, sem velit scelerisque nisi, ut aliquet nulla nunc non justo. Mauris eu ipsum risus, eu iaculis lorem. Ut ut nisl ipsum. Nulla facilisi. Maecenas nec quam justo. Vestibulum id libero lectus, sit amet sodales diam. Maecenas pulvinar, ante vel mollis porta, risus sem molestie purus, facilisis lacinia ipsum ligula ac orci. Ut nec elit tincidunt nisi lobortis bibendum eget sit amet odio.

Fusce rutrum elementum felis quis viverra. Donec blandit mattis consectetur. Vestibulum vitae magna at velit sodales placerat et sed risus. Aenean mollis urna non eros bibendum convallis. Duis in turpis ante, vel viverra urna. Duis in velit tellus. Cras suscipit imperdiet justo, semper pellentesque ligula laoreet a. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nulla sapien urna, gravida non pellentesque volutpat, sagittis porttitor mi. Etiam aliquam risus vel enim venenatis a elementum arcu lacinia. Nam et enim id mauris dictum elementum. Phasellus id porta diam. Nullam ornare odio sagittis ligula ultricies convallis. Integer ac suscipit libero. Fusce sed dolor urna, eu adipiscing metus. Fusce commodo nulla ac diam varius eu adipiscing mi egestas. Donec nibh ante, rutrum sit amet semper a, eleifend quis libero. Vivamus sed posuere nunc.

Abstract

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Donec sapien ipsum, fermentum eu malesuada ac, malesuada sed magna. Fusce mollis tempor eros, sed feugiat ante auctor ut. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed tincidunt, enim vel pulvinar porttitor, sem velit scelerisque nisi, ut aliquet nulla nunc non justo. Mauris eu ipsum risus, eu iaculis lorem. Ut ut nisl ipsum. Nulla facilisi. Maecenas nec quam justo. Vestibulum id libero lectus, sit amet sodales diam. Maecenas pulvinar, ante vel mollis porta, risus sem molestie purus, facilisis lacinia ipsum ligula ac orci. Ut nec elit tincidunt nisi lobortis bibendum eget sit amet odio.

Fusce rutrum elementum felis quis viverra. Donec blandit mattis consectetur. Vestibulum vitae magna at velit sodales placerat et sed risus. Aenean mollis urna non eros bibendum convallis. Duis in turpis ante, vel viverra urna. Duis in velit tellus. Cras suscipit imperdiet justo, semper pellentesque ligula laoreet a. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nulla sapien urna, gravida non pellentesque volutpat, sagittis porttitor mi. Etiam aliquam risus vel enim venenatis a elementum arcu lacinia. Nam et enim id mauris dictum elementum. Phasellus id porta diam. Nullam ornare odio sagittis ligula ultricies convallis. Integer ac suscipit libero. Fusce sed dolor urna, eu adipiscing metus. Fusce commodo nulla ac diam varius eu adipiscing mi egestas. Donec nibh ante, rutrum sit amet semper a, eleifend quis libero. Vivamus sed posuere nunc.

Algorytmy ewolucyjne w zadaniu klasteryzacji.

1 Wstęp

W niniejszej pracy rozpatrzymy problem klasteryzacji i spojrzymy na niego z perspektywy kilku wybranych artykułów. Skupimy się przede wszystkim na zastosowaniu algorytmów ewolucyjnych. Zaproponujemy także rozwiązanie i sposób jego implementacji.

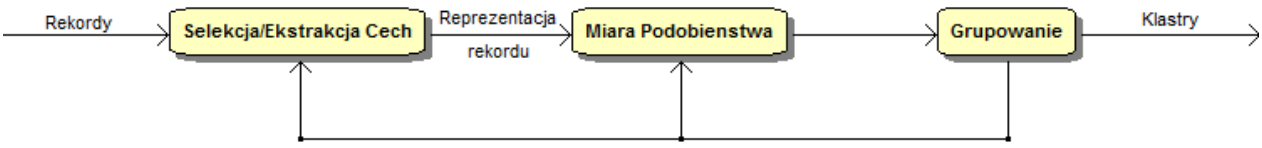
2 Motywacja

[Co to jest klasteryzacja, do czego i poco to się stosuje]

3 Komponenty

[Podstawowe komponenty zadania/algorytmu klasteryzacji]
Według podziału z [1] w zadaniu klasteryzacji można wyszczególnić następujące komponenty:

1. Reprezentacja rekordu. W tym także ekstrakcja cech i/lub ich selekcja.
2. Definicja miary podobieństwa odpowiednio do domeny zbioru danych.
3. Klasteryzacja lub grupowanie
4. Abstrakcja danych (jeśli potrzebne)
5. Ocena produktu (jeśli potrzebne)



rys 3-1 Etapy w klasteryzacji

- **Selekcja Cech** – identyfikacja i dobór cech z oryginalnego zbioru opisującego pojedynczy rekord, myślą o efektywności klasteryzacji.
- **Ekstrakcja Cech** – użycie jednej lub więcej metod transformacji, aby z zestawu wejściowego cech, wyprodukować nowe, użyteczniejsze cechy.
- **Miara Podobieństwa –** z pośród wielu stosowanych [2; 1; 3] najprostszą i najpowszechniej stosowaną jest miara Euklidesowa. Można także stosować miarę konceptualną [4]
- **Grupowanie –** ten krok może zostać wykonany na wiele sposobów. Wynik wyjściowy może być w postaci twardej (podział danych na grupy), lub rozmytej (każdy rekord ma wyznaczony poziom przynależności do każdego klastra).
- **Abtrakcja danych -** proces ekstrakcji prostej i zwartej reprezentacji zbioru danych. Celem jest prostota dalszej maszynowej analizy, lub czytelność i zrozumiałość dla ludzkiego odbiorcy. Zwykle w kontekście klasteryzacji, zwarty opis klastra w postaci prototypu lub centroida [3].
- **Walidacja klastrów –** ocena przebiegu klasteryzacji. Zwykle w tym celu stosuje się odpowiednie kryterium optymalności [5].

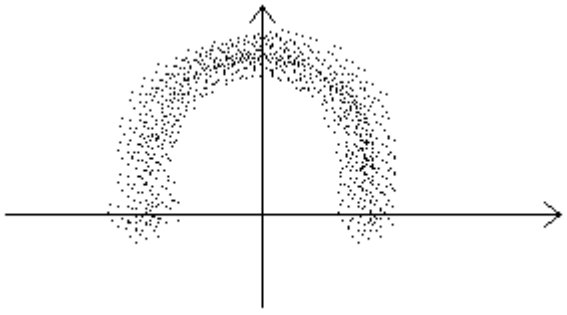
Algorytmy ewolucyjne w zadaniu klasteryzacji.

4 Definicje

- *Rekord* (lub *wektor cech*, *obiekt* lub *punkt*) **x** jest pojedynczym elementem danych, użytym w algorytmie klasteryzacji. Zwykle jest wektorem d miar: **x** = (x_1, …, x_d).
- Indywidualne komponenty x_i rekordu **x** zwane są *cechami* (lub *atrybutami*).
- *d* to *wymiarowość* rekordu lub przestrzeni rekordów.
- *Zbiór rekordów* oznaczony jako $\mathcal{X}$ = {$\mathbf{x}_1$, …, $\mathbf{x}_n$}. W wielu przypadkach, zbiór który ma podlegać klasteryzacji jest postrzegany jako macierz $n \times d$ rekordów.
- *Klasa*
- *Twarde* techniki klasteryzacji każdemu rekordowi x_i przypisują etykietę klasy l_i Identyfikując jego klasę. Zbiorem wszystkich etykiet dla zbioru rekordów $\mathcal{X}$ jest $\mathcal{L}$ = {l_i, …, l_n}, z l_i {1, …, k} $l_i \in \{1, \dots, k\}$, gdzie *k* jest liczbą klastrów.
- Procedury *rozmytej* klasteryzacji przypisują każdemu rekordowi stopień przynależności f_{ij} do danego klastra wyjściowego *j*
- *Miękkie* techniki klasteryzacji przypisują każdemu rekordowi $\mathbf{x}_i$ stan przynależności l_{ij} do danego *j*-tego klastra. Przy czym $l_{ij} \in \{0, 1\}$, gdzie 0- nie należy, 1-należy do klastra.
- *Miara odległości* jest miarą określającą podobieństwo rekordów w przestrzeni ich atrybutów

5 Reprezentacja rekordu

Brak konkretnych przewodników na temat poprawnej reprezentacji rekordów i doboru cech użytych dla specyficznych sytuacji. Na ogół proces generowania rekordów nie podlega bezpośredniej kontroli. Rolą użytkownika jest gromadzenie faktów i przypuszczeń, ewentualnie dokonać wyboru i ekstrakcji cech. Na pewno dokładna analiza dostępnych cech i dostępnych przekształceń (nawet tych prostych) może dać znacznie lepsze wyniki. Może to zadecydować czy otrzymamy proste i dobrze zrozumiałe klastry, czy skomplikowaną strukturę, której prawdziwą naturę trudno odgadnąć.



rys 5-1 . ukazuje prosty przykład. Punkty, w tej przestrzeni 2D cech, są zorganizowane w kształt mniej więcej równo oddalony od jednego punktu środka. Jeśli wybrać współrzędne Kartezjańskie, większość algorytmów klastrowania prawdopodobnie podzieli tą figurę na dwa lub więcej klastrów. Jeśli by jedna użyć reprezentacji koordynatów polarnych, uzyskanie pojedynczego klastra ma większe prawdopodobieństwo.

Rekord może mierzyć tak fizyczny obiekt (np. krzesło) jak i abstrakcyjny (np. styl pisania). Jak wspomniano powyżej, każdy obiekt reprezentowany jest, jako wektor, gdzie każdy wymiar jest pojedynczą cechą. Cechy te można podzielić na ilościowe i jakościowe. Dla przykładu, jeśli waga

i kolor były by dwiema użytymi cechami, wtedy (20, black) jest reprezentacją czarnego obiektu o wadze 20 jednostek. Cechy można podzielić na następujące typy.

(1) Cechy ilościowe
   A. Wartości ciągłe (np. waga)
   B. Wartości dyskretne (np. liczba komputerów)
   C. Wartości przedziału (np. czas przebiegu wydarzenia)
(2) Cechy jakościowe
   A. Nominalne lub nieposortowane (np. kolor)
   B. Porządkowy (np. ranga wojskowa, temperatura („gorąco", „zimno"), czy intensywność głośności („cicho", „głośno")

Można także używać cech ustrukturyzowanych, reprezentowanych przez drzewo, gdzie rodzic jest generalizacją swoich dzieci. Dla przykładu, węzeł rodzic może być generalizacją dla dzieci oznaczonych, jako „samochody", „autobusy", „ciężarówki" i „motory". Dalej, węzeł „samochody" może być generalizacją samochodów typu „Toyota", „Ford", „Mercedes"

Ważne jest, aby wyizolować tylko te najbardziej wartościowe, opisowe i dyskryminujące cechy z zestawu wejściowego, a następnie indywidualnie poddać kolejnym przetwarzaniom analizy. Techniki wyboru cech wyznaczają podzbiory istniejących cech. Natomiast techniki ekstrakcji obliczają nowe cechy z już istniejących. W obu przypadkach celem jest polepszenie jakości klasyfikacji i/lub efektywności obliczeniowej. Dobór cech to bardzo obszerny temat, jednakże w klasteryzacji często zmuszeni jesteśmy na dokonywanie wyborów „na wyczucie" a nie rzadko jest to proces prób i błędów, gdzie wybierane są różne zestawy cech obiektów i wybór jest oceniany po zakończeniu procesu klasteryzacji.

6 Miara bliskości

Jako, że podobieństwo jest podstawą definicji klastra, pomiar podobieństwa dwóch rekordów z tej samej przestrzeni cech, jest bardzo ważne dla większości procedur klasteryzacji. Z powodu różnorodności typów wartości cech i ich skalowania, miara (lub miary) podobieństwa, muszą być dobierane bardzo ostrożnie. Najbardziej powszechną praktyką jest obliczanie *miary niepodobieństwa* pomiędzy dwoma rekordami używając miary odległości zdefiniowanej na przestrzeni cech. Najczęściej w przypadku cech o wartościach ciągłych stosowana jest odległość Euklidesowa.

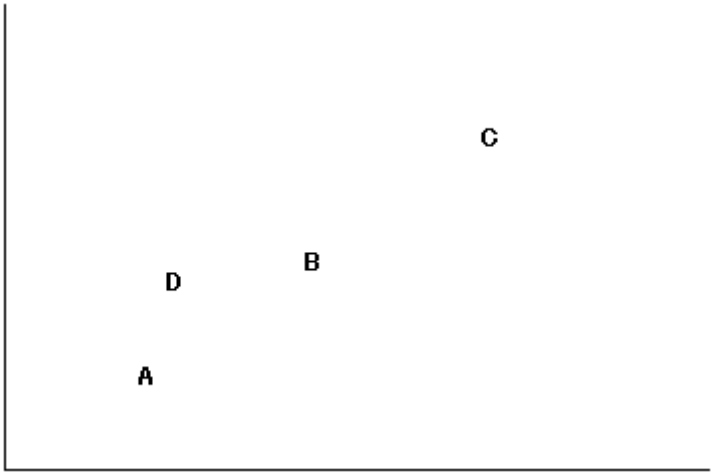
$$d(\mathbf{x}_i, \mathbf{x}_j) = \left(\sum_{k=1}^{d} (\mathbf{x}_{i,k} - \mathbf{x}_{j,k})^p \right)^{1/p} = \|\mathbf{x}_i - \mathbf{x}_j\|_p$$

Odległość Euklidesowa jest intuicyjnie używane do obliczeń odległości obiektów w dwu i trzy wymiarowych przestrzeniach. Sprawuje się doskonale tak z wyizolowanymi jak i złożonymi klastrami. Wadą używania tej miary jest fakt, że szybko cechy o mniejszej skali stracą na wartości zdominowane przez cechy o większej skali. Rozwiązaniem tego problemu jest normalizacja używanych wartości lub stosowanie wag.

Obliczanie odległości pomiędzy rekordami, których część lub wszystkie cechy nie są wartościami ciągłymi, staje się o wiele trudniejsze. W ekstremalnych przypadkach wartość

odległości jest określana binarnie. I tutaj jednak istnieje wiele prac, które starają się rozwiązać ten problem.

Rekordy mogą być także reprezentowane używając stringów czy struktur drzewiastych. Wymaga to wtedy syntaktycznego lub statystycznego podejścia do miary odległości. Ale są one zdecydowanie mniej przydatne.



rys 6-1 – D i B są bardziej podobne niż B i C

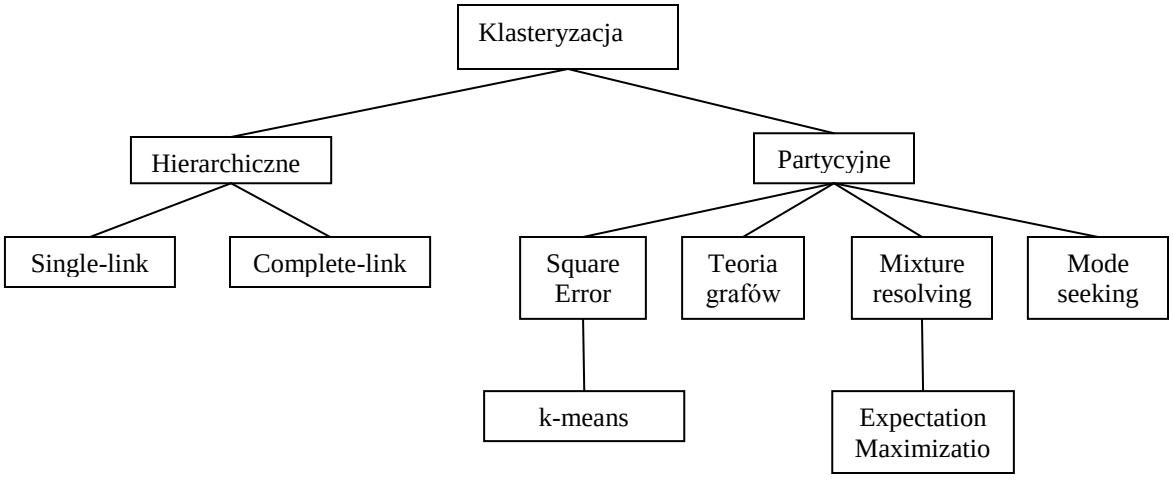
Są także miary odległości, który pod uwagę biorą otoczenie lub sąsiednie punkty, zwanych kontekstem. [6; 7; 4]. Najważniejszym przykładem tych metod jest mutual neighbour distance (MND) zaproponowane przez [6].

$$MND(\mathbf{x}_i, \mathbf{x}_j) = NN(\mathbf{x}_i, \mathbf{x}_j) + NN(\mathbf{x}_j, \mathbf{x}_i)$$

Gdzie NN($\mathbf{x}_i$, $\mathbf{x}_j$) jest numerem sąsiada $\mathbf{x}_j$ w odniesieniu do punktu $\mathbf{x}_i$. Tzn. funkcja ta określa, którym z kolei najbliższym sąsiadem $\mathbf{x}_i$ jest $\mathbf{x}_j$. Na rys 6-1 –rys 6-1 A i D są ja bliższymi sąsiadami. NN(A,D) = NN(D,A) = 1. Zatem MND(A,D) = 2. Jednakże MND(C,B) = 4. Miara ta nie jest metryczna, może ulec zmianie, gdy tylko pojawią się nowe obiekty w sąsiedztwie i nadal potrzebuje standardowej funkcji odległości. Ale z powodzeniem została zaimplementowana w aplikacjach takich jak [8].

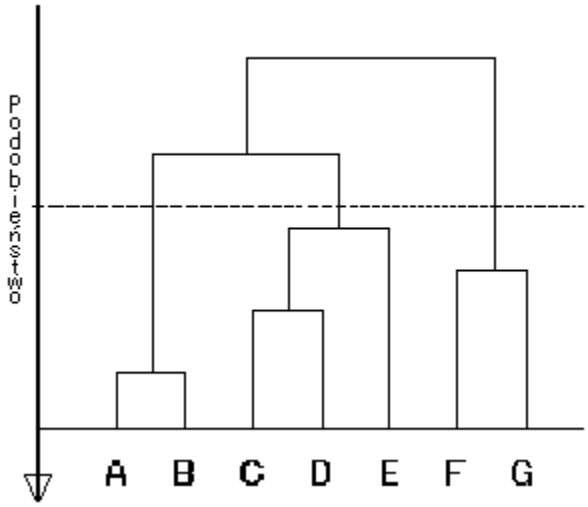
7 Techniki klasteryzacji

Na podstawie dyskusji w [1] można wprowadzić następujący podział technik klasteryzacji:



7.1 Hierarchiczne algorytmy klasteryzacji

Zadaniem algorytmów hierarchicznych jest wygenerowanie drzewa reprezentującego zagnieżdżone grupowanie rekordów. Tak jak to przedstawiono na rys 7-1. Następnie drzewo to może zostać przecięte na dowolnym poziomie dając nam różne rozwiązania klasteryzacji tego samego zbioru danych.



rys 7-1 Drzewo uzyskane za pomocą algorytmu single-link

Większość algorytmów hierarchicznych to warianty single-link [9; 1], complete-link [10] oraz minimum-variance [11; 12]. Z tych najpopularniejszymi algorytmami są single-link oraz complete-link, a różnią się jedynie charakterystyką „podobieństwa" pomiędzy klastrami. W single-link jest to minimum z odległości wszystkich par rekordów z tych klastrów (jeden rekord z jednego klastra, drugi z drugiego). Przypadku complete-link jest to maksimum. W obu przypadkach klastry są scalane używając kryterium minimalnej odległości.

7.2 Algorytmy partycyjne

Zamiast całej struktury klastrów, algorytmy partycyjne uzyskują pojedyncze partycje danych. Zyskują one przewagę przy dużych zbiorach, gdzie tworzenie drzew obliczeniowo jest bardzo wymagające. Jednakże największym problemem algorytmów partycyjnych jest dobór odpowiedniej liczby klastrów. Ponieważ ich liczna na ogół jest wymagana jeszcze przed rozpoczęciem algorytmu. Istnieje wiele propozycji aby znaleźć najlepsze rozwiązanie na ten problem [13]. Ale w praktyce najlepszy wynik uzyskuje się poprzez wykonanie algorytmu kilka razy i wybranie najlepsze rezultatu.

7.2.1 Squared error

Najbardziej intuicyjna i najczęściej stosowana partycyjna technika klasteryzacji, która dobrze radzi sobie z klastrami wyizolowanymi i zwartymi. Jest to nic innego jak łączna suma odległości rekordów do centroidów klastrów, do których przynależą. Squared error samo w sobie nie jest technika ale kryterium oceniającym jakość rozwiązania. Dla klasteryzacji L zbioru rekordów H (zawierającego K klastrów) obliczamy za pomocą wzoru [14]:

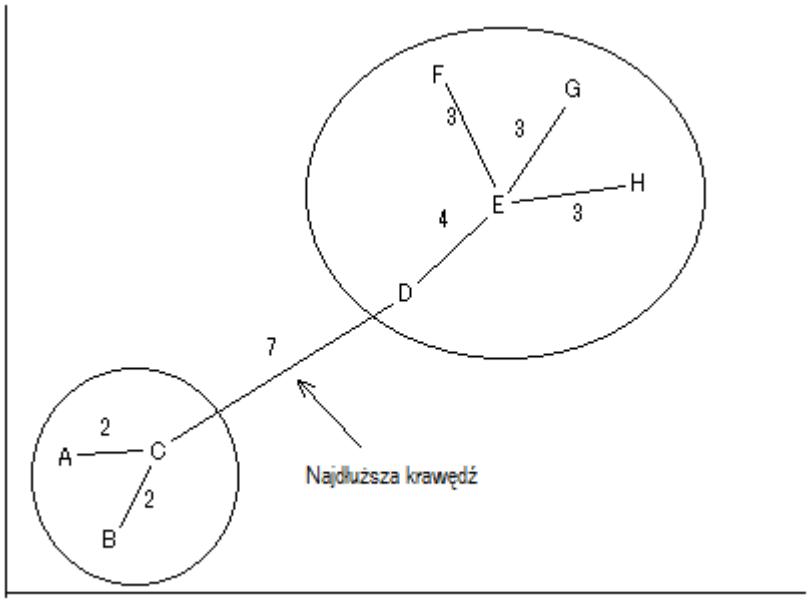
$$e^2(L,H) = \sum_{j=1}^{K} \sum_{i=1}^{n_j} \left\| x_i^{(j)} - c_j \right\|^2 \tag{7-1}$$

Gdzie $x_i^{(j)}$ jest i-tym rekordem należącym do j-tego klastra, a c_j jest centroidem tego klastra. Najprostszym i bardzo szeroko stosowanym algorytmem, korzystającym z tego kryterium, jest k-mens [15]. Rozpoczyna się on z losowym, startowym zestawem partycji, a następnie realokuje rekordy przypisując je do innych klastrów, bazując na podobieństwie rekordu do centroida, dopóki nie zostanie spełnione kryterium (np. nie było kolejnych przesunięć rekordów pomiędzy partycjami lub kryterium Squared-error przestało się znacząco zmniejszać). K-means zyskał swoją popularność dzięki łatwości implementacji oraz złożoności czasowej (O(n), gdzie n to liczba rekordów). Ma on jednak swoje słabe strony. Jest on wrażliwy na dobór startowych partycji i może ostatecznie wyniknąć lokalnym minimum. Powstało wiele wariantów [2] tego algorytmu. Niektóre wstępnie próbują dokonać wyboru partycji startowych, aby zwiększyć swoją szansę w odnalezieniu globalnego minimum. Innym wariantem jest łączenie lub dzielenie wynikowych partycji. Jeśli różnorodność wewnątrz klastra jest zbyt duża (przekracza odp. wskaźnik), jest on dzielony na dwie osobne partycje. Jeśli natomiast centroidy dwóch klastrów są odpowiednio bliskie ich partycje są łączone. Technika ta pozwala na uzyskanie optymalnego zestawu klastrów zaczynając od całkiem innego zbioru startowego. Technika ta jest stosowana w ISODATA [16]. Inne warianty k-means tworzone są poprzez zmianę funkcji kryterium. Np. algorytmy dynamiczne klasteryzacji (które dopuszczają reprezentacje klastra inną niż centroid) zaproponowane w [17; 18; 19].

7.2.2 Teorie grafów

Najlepiej znanym algorytmem używającym tego podejścia jest algorytm bazujący na skonstruowaniu z danych Minimalnego Drzewa Rozpiętego (eng. Minimal Spanning Tree - MST) [20], a następnie na usuwaniu krawędzi o największej długości w celu sformułowania klastrów. Przykład na rys 7-2 ukazuje MST uzyskane z 8 dwu-wymiarowych punktów. Poprzez zerwania

połączenia CD (krawędź z największą długością Euklidesową), uzyskujemy dwa klastry ({A, B, C} oraz {D, E, F, G, H}). Kolejny podział może zostać dokonany na krawędzi DE itd.



rys 7-2 Przykład użycia minimalnego drzewa rozpiętego o klasteryzacji.

Podejście hierarchiczne także odnosi się do grafów. Klastry Single-link są podgrafami minimalnego drzewa rozpinającego [21; 22]. Klastry complete-link są maksymalnie kompletnymi subgrafami [23] i w pracach [24; 25] uznawane są za definicję klastra. Inne podejście używające grafów w nie-hierarchicznych strukturach nakładających się klastrów prezentuje [26].

7.3 Klasteryzacja rozmyta

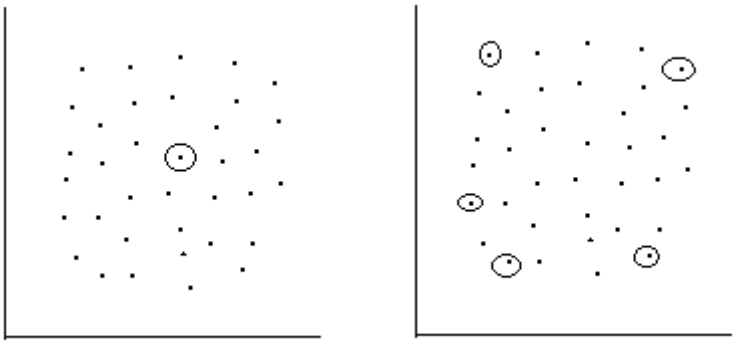
Tradycyjne algorytmy klasteryzacji generują partycję. W każdej partycji, każdy rekord należy do jednej i tylko jednej partycji. Klastry w twardej klasteryzacji są rozłączne. Klasteryzacja rozmyta rozszerza to podejście przypisując rekordy do każdego klastra używając funkcji przynależności [27]. Wynikiem są klastry ale nie partycje. Stosuje się tutaj np. kryterium wyważonego błędu kwadratowego (Squared-error).
Klastrowanie rozmyte pojawiło się w [28]. Natomiast książka [29] jest dobrym źródłem na temat Klastrowania rozmytego. Najpopularniej stosowany jest tutaj algorytm fuzzy c-means (FCM), mimo to, że jest lepszy od twardego k-means w ignorowaniu lokalnych minimum, nadal jednak potrafi osiągnąć taki wynik. Największy problem stanowi tutaj funkcja przynależności i jej określenie. W [29] zaproponowano generalizację tego algorytmu. W [30] zaproponowano c-shell, którego zaletą miało być wykrywanie granic okrągłych i eliptycznych.

7.4 Reprezentacja klastra

W aplikacjach, których zadaniem jest odnalezienie pewnej liczby klas lub klastrów w zbiorze danych, zestaw partycji jest produktem końcowym. W pracach [31; 3; 32] proponuje się następujące podejścia:

- Reprezentacja klastra za pomocą centroida lub zestawu skrajnych punktów.
- Reprezentacja klastra za pomocą węzłów w drzewie klasyfikacji.
- Reprezentacja za pomocą wyrażeń logicznych

Z nich wszystkich najpopularniejszą metodą jest użycie centroida. Doskonale nadaje się, gdy klaster jest zwarty. W innych przypadkach jednak sobie z tym nie radzi. Wtedy użycie kolekcji punktów granicznych może z powodzeniem oddać kształt opisywanego klastra. Wtedy wielkość tej kolekcji jest proporcjonalna do skomplikowalności figury przez nią opisywanej.



rys 7-3 Przykład reprezentacji za pomocą punktów. Centroid po lewej i punkty granicznej po prawej.

7.5 Sztuczne Sieci Neuronowe w klasteryzacji

Sztuczne Sieci Neuronowe [33] znajdują szerokie zastosowanie tak w klasteryzacji jak i klasyfikacji [34; 35].

8 Bibliografia:

1. *Algorithms for Clustering Data.* **JAIN, A. K. i DUBES, R. C.** Upper Saddle River, NJ : Prentice-Hall, 1988.
2. *Cluster Analysis for Applications.* **ANDERBERG, M. R.** New York, NY : Academic Press, Inc., 1973.
3. *Clustering Analysis.* **DIDAY, E. i SIMON, J. C.** Secaucus, NJ : Springer-Verlag, 1976, Digital Pattern Recognition, strony 47–94.
4. *Automated construction of calsifications: conceptual clustering versus numerical taxonomy.* **MICHALSKI, R., STEPP, R. E. i DIDAY, E.** brak miejsca : IEEE, 5 Sept 1983, IEEE Trans. Pattern Anal. Mach. Intell., strony 396–409.
5. *Cluster analysis and related issues.* **DUBES, R. C.** River Edge, NJ : World Scientific Publishing Co, 1993, Handbook of Pattern Recognition & Computer Vision, strony 3–32.
6. *Agglomerative clustering using the concept of mutual nearest neighborhood.* **GOWDA, K. C. i G., KRISHNA.** 105–112, 1977, Pattern Recogn.
7. *Clustering using a similarity method based on shared near neighbors.* **JARVIS, R. A. i PATRICK, E. A.** 1025–1034, 1973, IEEE Trans. Comput.
8. *Symbolic clustering using a new dissimilarity measure.* **GOWDA, K. C. AND DIDAY, E.** 368–378, 1992, IEEE Trans. Syst. Man Cybern, Tom 22.
9. *Numerical Taxonomy.* **SNEATH, P. H. A. i SOKAL, R. R.** London : Freeman, 1973.
10. *Step-wise clustering procedures.* **KING, B.** brak miejsca : J. Am. Stat. Assoc., 1967.
11. *Hierarchical grouping to optimize an objective function.* **WARD, J. H. JR.** brak miejsca : J. Am. Stat. Assoc., 1963.
12. *A survey of recent advances in hierarchical clustering algorithms which use cluster centers.* **MURTAGH, F.** brak miejsca : Comput. J., 1984.
13. *How many clusters are best?—an experiment.* **DUBES, R. C.** 1987, Pattern Recogn.
14. *Data clustering: a review.* **Jain, A. K., Murty, M. N. i Flynn, P. J.** 3, brak miejsca : ACM, 1999, Comput. Surveys, Tom 31, strony 264-323.
15. *Some methods for classification and analysis of multivariate observations.* **MCQUEEN, J.** 1967. In Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability.
16. **BALL, G. H. i HALL, D. J.** *ISODATA, a novel method of data analysis and classification.* Tech. Rep.. Stanford University,. Stanford, CA. : brak nazwiska, 1956.
17. *The dynamic cluster method in non-hierarchical clustering.* **DIDAY, E.** 1973, J. Comput. Inf. Sci., Tom 2, strony 61–88.
18. *Clustering criterion and multi-variate normal mixture.* **SYMON, M. J.** Biometrics, Tom 77, strony 35–43.
19. *A self-organizing network for hyperellipsoidal clustering (HEC).* **MAO, J. AND JAIN, A. K.** 1996, IEEE Trans. Neural Netw., Tom 7, strony 16–29.
20. *Graph-theoretical methods for detecting and describing gestalt clusters.* **ZAHN, C. T.** 1971, IEEE Trans. Comput., Tomy C-20 (Apr.),, strony 68–86.
21. *Minimum spanning trees and single-linkage cluster analysis.* **GOWER, J. C. i ROSS, G. J. S.** 1969, Appl. Stat., Tom 18, strony 54–64.
22. *Semantic clustering of index terms.* **GOTLIEB, G. C. i KUMAR, S.** 1968, J. ACM, Tom 15, strony 493–513.

23. *A graphtheoretic approach to goodness-of-fit in complete-link hierarchical clustering.* **BACKER, F. B. i HUBERT, L. J.** 1976, J. Am. Stat. Assoc., Tom 71, strony 870–878.
24. *An analysis of some graph theoretical clustering techniques.* **AUGUSTSON, J. G. i MINKER, J.** 4 Oct 1970, J. ACM, Tom 17, strony 571–588.
25. *A comparison of the stability characteristics of some graph theoretic clustering methods.* **RAGHAVAN, V. V. i YU, C. T.** 1981, IEEE Trans. Pattern Anal. Mach. Intell., Tom 3, strony 393–402.
26. *A stratificational overlapping cluster scheme.* **OZAWA, K.** 1985, Pattern Recogn., Tom 18, strony 279–286.

9 Spis rysunków:

10 Spis wzorów:

Algorytmy ewolucyjne w zadaniu klasteryzacji.

Załączniki do pracy