Eksploracja zasobów internetowych

Wykład 5

Wstęp do grupowania danych

mgr inż. Maciej Kopczyński

Białystok 2014

Wstęp

Istnieją dwie podstawowe metody klasyfikowania obiektów:

- metoda z nauczycielem,
- metoda bez nauczyciela.

Metoda z nauczycielem wymaga wcześniej przygotowanego zbioru danych sklasyfikowanego przez zewnętrznego eksperta. Na bazie tego zbioru mechanizm się uczy i później operuje na innych danych.

Metoda bez nauczyciela nie wymaga przygotowywania zbioru uczącego. Te metody klasyfikacji zaczynają od razu działanie na danych rzeczywistych.

Wstęp

Przykładem mechanizmu klasyfikacji danych wykorzystującego metodę bez nauczyciela jest grupowanie danych.

Podstawowym celem grupowania danych jest:

- wyszukiwanie wzorców,
- wskazywanie wspólnych cech poszczególnych obiektów w przestrzeniach wielowymiarowych,
- odnajdywanie podobnych obiektów,
- łączenie obiektów w struktury hierarchiczne.

Grupowanie danych

Sformułowanie problemu:

Dany jest zbiór obiektów (rekordów). Problemem jest znalezienie naturalnego pogrupowania obiektów w klasy (klastry, skupienia) lub odnalezienie obiektów o podobnych cechach.

Rozwiązanie problemu:

Zastosowanie procesu grupowania obiektów (rzeczywistych bądź abstrakcyjnych) o podobnych cechach w klasy, nazywane klastrami lub skupieniami. Klasy powinny być jak najbardziej różne od siebie.

Klastry

Pojedynczy klaster może być zdefiniowany jako:

- zbiór obiektów, które są „podobne”,
- zbiór obiektów takich, że odległość pomiędzy dwoma dowolnymi obiektami należącymi do klastra jest mniejsza niż odległość pomiędzy dowolnym obiektem należącym do klastra i dowolnym obiektem nie należącym do tego klastra,
- spójny obszar przestrzeni wielowymiarowej, charakteryzujący się dużą gęstością występowania obiektów.

Grupowanie w sieciach WWW

Możliwe zastosowania mechanizmu grupowania w przypadku dokumentów sieci WWW:

- automatyczne tworzenie *Topic Directories*,
- grupowanie wyników zwracanych przez wyszukiwarki internetowe,
- zwiększenie trafności zwracanych wyników bazujące na stworzonych grupach,
- odkrywanie wzorców i zależności w sieciach WWW.

Rodzaje grupowania danych

Grupowanie danych może być rozpatrywane w czterech kategoriach:

- bazujące na modelu oraz bazujące na podziale (ang. *partitioning*),
- deterministyczne oraz probabilistyczne,
- hierarchiczne oraz płaskie,
- inkrementacyjne oraz całkowite (ang. *batch*).

Każdy z wymienionych wyżej rodzajów grupowania ma zastosowanie zależnie od rozpatrywanej dziedziny problemu. W przypadku sieci WWW najczęściej wykorzystywane jest grupowanie hierarchiczne oraz bazujące na podziale.

Rodzaje grupowania danych

Grupowanie bazujące na modelu polega na zbudowaniu bezpośrednich reprezentacji stworzonych klastrów, zaś grupowanie bazujące na podziale polega na przeglądaniu obiektów każdego klastra.

Grupowanie deterministyczne określa przynależność obiektu do klastra za pomocą wartości boolowskiej, zaś w grupowaniu probabilistycznym przynależność obiektu do grupy jest definiowana poprzez wartość określającą prawdopodobieństwo.

Rodzaje grupowania danych

Grupowanie płaskie dzieli zbiór obiektów na pojedyncze klastry nie posiadające wzajemnych relacji ich wiążących, zaś w przypadku grupowania hierarchicznego tworzone klastry posiadają drzewiastą strukturę.

Grupowanie inkrementacyjne podczas działania algorytmu sprawdza w danym kroku jeden obiekt, zaś grupowanie całkowite podejmuje decyzję na bazie kilku obiektów jednocześnie.

Reprezentacja danych w grupowaniu

Mechanizmy grupowania danych mogą być zastosowane do takich danych, które mogą zostać opisane za pomocą atrybutów (cech). Do każdego atrybutu przypisane są określone zestawy wartości, które atrybut ten może przyjąć.

W przypadku dokumentów sieci WWW opis ten jest zapewniony poprzez reprezentacje wektorowe treści stron. Każdy dokument reprezentuje w przestrzeni wielowymiarowej jeden punkt, którego współrzędne zależą od termów definiujących treść danego dokumentu.

Miara podobieństwa

Każdy rodzaj grupowania wymaga pewnej miary podobieństwa. W przypadku zestawu punktów w danej przestrzeni, najlepiej jest wykorzystać miarę euklidesową.

W przypadku dokumentów sieci WWW opisanych jako wektory z wykorzystaniem współrzędnych modelu *TFIDF*, najbardziej odpowiednie jest użycie podobieństwa cosinusowego:

$$\mathrm{sim}(d_1, d_2) = \frac{d_1 \cdot d_2}{\|d_1\| \, \|d_2\|}$$

gdzie: d_1, d_2 – wektory dokumentów, zaś $\|d_1\|$ – norma L_2 (długość)

Sposoby obliczania podobieństwa

W grupowaniu wyróżnia się cztery podstawowe metody obliczania podobieństwa pomiędzy klastrami:

- podobieństwo pomiędzy centroidami klastrów,
- maksymalne podobieństwo pomiędzy dokumentami w klastrach,
- minimalne podobieństwo pomiędzy dokumentami w klastrach,
- średnie podobieństwo pomiędzy dokumentami w klastrach.

Podane powyżej zależności mają również zastosowanie dla pojedynczych dokumentów, gdyż pojedynczy dokument może być reprezentowany jako klaster jednoelementowy.

Sposoby obliczania podobieństwa

Obliczanie podobieństwa na bazie centroidów klastrów jest definiowane jako:

$$\mathrm{sim}(S_1, S_2) = \mathrm{sim}(c_1, c_2)$$

gdzie centroid *c* klastra *S* jest obliczany jako:

$$c = \frac{1}{|S|} \sum_{d \in S} d$$

Obliczanie maksymalnego podobieństwa dokumentów w klastrach:

$$\mathrm{sim}(S_1, S_2) = max_{d_1 \in S_1, d_2 \in S_2} \mathrm{sim}(d_1, d_2)$$

Ten rodzaj grupowania jest również określany jako metoda najbliższego sąsiada (ang. *nearest-neighbor clustering*).

Sposoby obliczania podobieństwa

Obliczanie minimalnego podobieństwa dokumentów w klastrach:

$$\mathrm{sim}(S_1, S_2) = min_{d_1 \in S_1, d_2 \in S_2} \mathrm{sim}(d_1, d_2)$$

Ten rodzaj grupowania jest również określany jako metoda najdalszego sąsiada (ang. *farthest-neighbor clustering*).

Obliczanie średniego podobieństwa dokumentów w klastrach:

$$\mathrm{sim}(S_1, S_2) = \frac{1}{|S_1||S_2|} \sum_{d_1 \in S_1, d_2 \in S_2} \mathrm{sim}(d_1, d_2)$$

Grupowanie hierarchiczne

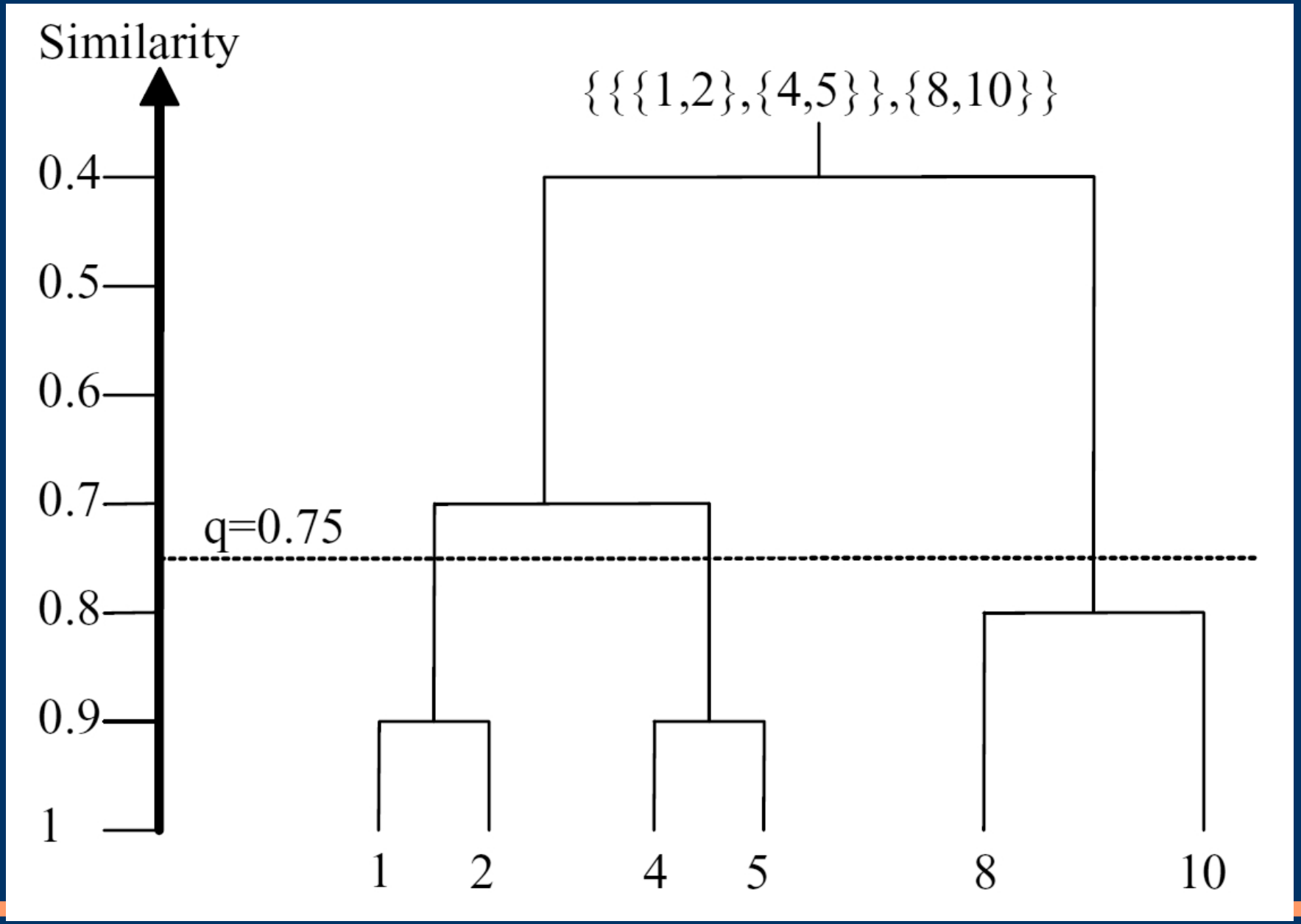
Algorytm *Hierarchical Agglomarative Clustering* jest przykładem algorytmu hierarchicznego. Wynikiem działania algorytmów hierarchicznych jest drzewiasta struktura klastrów nazywana dendrogramem.

Na szczycie dendrogramu znajduje się korzeń (klaster zawierający wszystkie pozostałe klastry), zaś na dole umieszczone są liście reprezentujące jednoelementowe klastry dokumentów.

Kolejny slajd prezentuje dendrogram dla grupowania zbioru liczb {1, 2, 4, 5, 8, 10}. Miara podobieństwa jest zdefiniowana jako:

$$\mathrm{sim}(d_{1,} d_2) = \frac{10 - |d_2 - d_1|}{10}$$

Grupowanie hierarchiczne



Grupowanie hierarchiczne

Istnieją dwa podejścia do grupowania hierarchicznego:

- scalające (*agglomerative*) – polegające na tworzeniu w każdym kroku coraz większych klastrów poczynając od klastrów jednoelemetowych,
- dzielące (*divisible*) – polegające na tworzeniu w każdym kroku coraz mniejszych klastrów poczynając od jednego dużego klastra zawierającego wszystkie obiekty.

Grupowanie hierarchiczne

Podobieństwo pomiędzy klastrami zmniejsza się w miarę przechodzenia w górę dendrogramu. Tak więc w pewnym momencie trzeba zakończyć proces grupowania.

Jakie parametry stopu można przyjąć aby zakończyć algorytm?

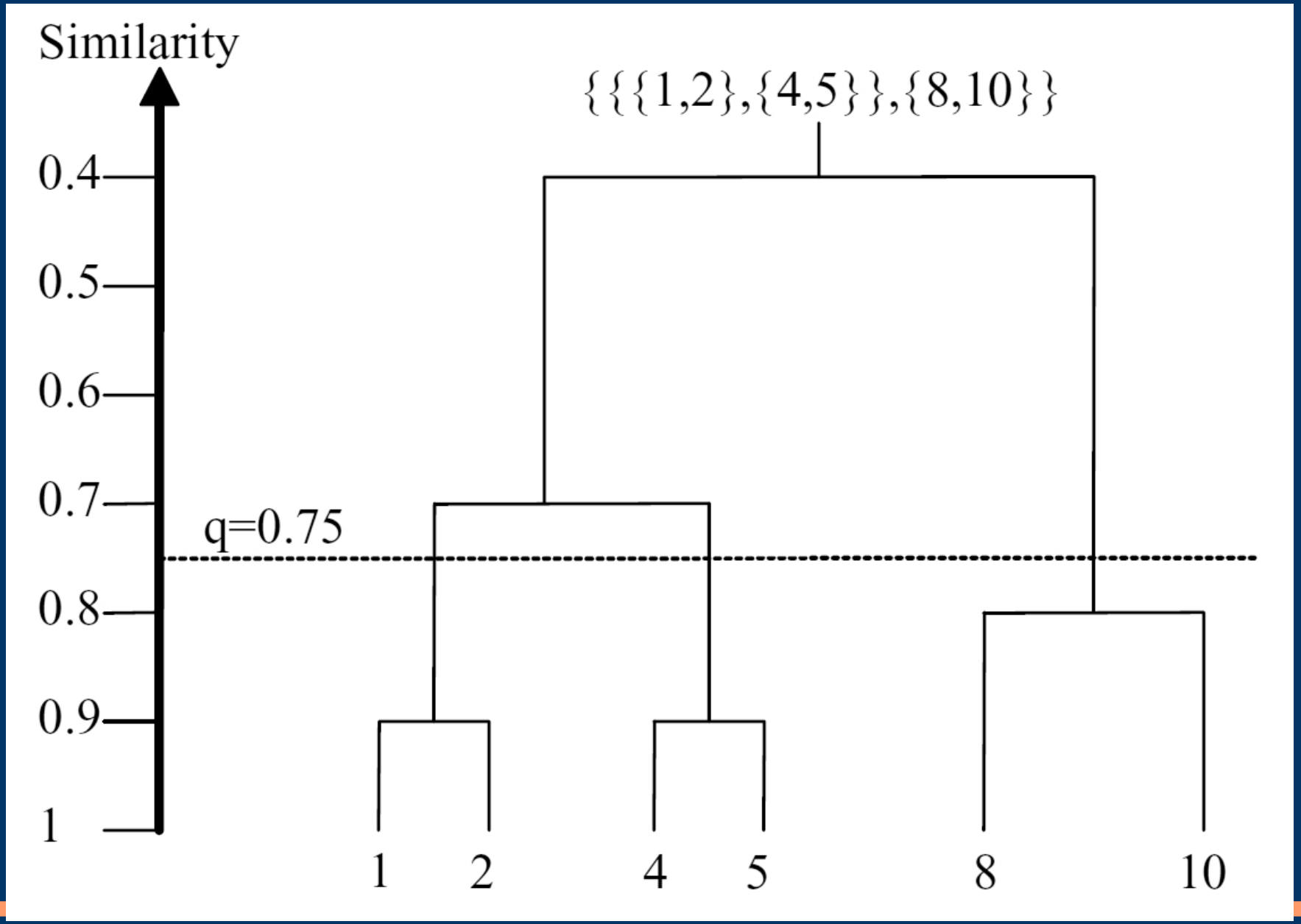
Algorytm można zatrzymać w momencie, gdy stworzona zostanie odpowiednia ilość klastrów (parametr k) lub w momencie, gdy podobieństwo pomiędzy klastrami spadnie poniżej pewnego poziomu (parametr q).

Grupowanie hierarchiczne

Działanie algorytmu *Hierarchical Agglomarative Clustering* można przestawić w kilku krokach:

1. Zainicjalizuj *G* jako zbiór klastrów jednoelementowych.
2. Jeśli $|G| \leq k$, to zakończ algorytm.
3. Znajdź takie dwa klastry S_i, S_j, że $(i, j) = arg\,max_{(i,j)}\,\mathrm{sim}(S_i, S_j)$
4. Jeśli $\mathrm{sim}(S_i, S_j) < q$, to zakończ algorytm.
5. Usuń ze zbioru *G* klastry S_1 oraz S_2.
6. Powiększ zbiór *G* o nowy klaster zawierający S_1 oraz S_2.
7. Przejdź do kroku 2.

Grupowanie hierarchiczne



Grupowanie hierarchiczne

Grupowanie hierarchiczne dokumentów sieci WWW wymaga przeliczania wartości podobieństwa pomiędzy poszczególnymi dokumentami w klastrach.

W jaki sposób można zoptymalizować szybkość działania algorytmu?

Aby grupowanie dokumentów działało szybciej można stworzyć macierz wzajemnego podobieństwa dokumentów (przeliczyć podobieństwa pomiędzy wszystkimi zindeksowanymi dokumentami). Koszt pamięciowy i czasowy: n^2.

Przykład

Na kolejnym slajdzie pokazany zostanie przykład grupowania zbioru dokumentów WWW ze strony uczelni *CCSU*.

Grupowanie zostało przeprowadzone dla parametru *k* równego 1 oraz dwóch wartości parametru *q*: $q = 0$ oraz $q = 0{,}04$.

Grupowanie wykorzystywało algorytm najbliższego sąsiada jako metodę obliczania podobieństwa pomiędzy dwoma klastrami.

Parametr *q*:

q = 0

Średnie podobieństwo międzyklastrowe:

0,4257

```
1 [0.0224143]
  2 [0.0308927]
    3 [0.0368782]
      4 [0.0556825]
        5 [0.129523]
          Art
          Theatre
        Geography
      6 [0.0858613]
        7 [0.148599]
          Chemistry
          Music
        8 [0.23571]
          Computer
          Political
    9 [0.0937594]
        10 [0.176625]
          Communication
          Economics
        Justice
  11 [0.0554991]
    12 [0.0662345]
        13 [0.0864619]
          14 [0.177997]
            History
            Philosophy
          15 [0.186299]
            English
            Languages
        16 [0.122659]
          Anthropology
          Sociology
    17 [0.0952722]
      18 [0.163493]
        19 [0.245171]
          Biology
          Mathematics
      Psychology
    Physics
```

```
1 []
  2 [0.0554991]
    3 [0.0662345]
      4 [0.0864619]
        5 [0.177997]
          History
          Philosophy
        6 [0.186299]
          English
          Languages
      7 [0.122659]
        Anthropology
        Sociology
    8 [0.0952722]
      9 [0.163493]
          10 [0.245171]
            Biology
            Mathematics
        Psychology
      Physics
  11 [0.0556825]
      12 [0.129523]
        Art
        Theatre
    Geography
  13 [0.0858613]
      14 [0.148599]
        Chemistry
        Music
      15 [0.23571]
        Computer
        Political
    16 [0.0937594]
      17 [0.176625]
        Communication
        Economics
      Justice
```

Parametr *q*:

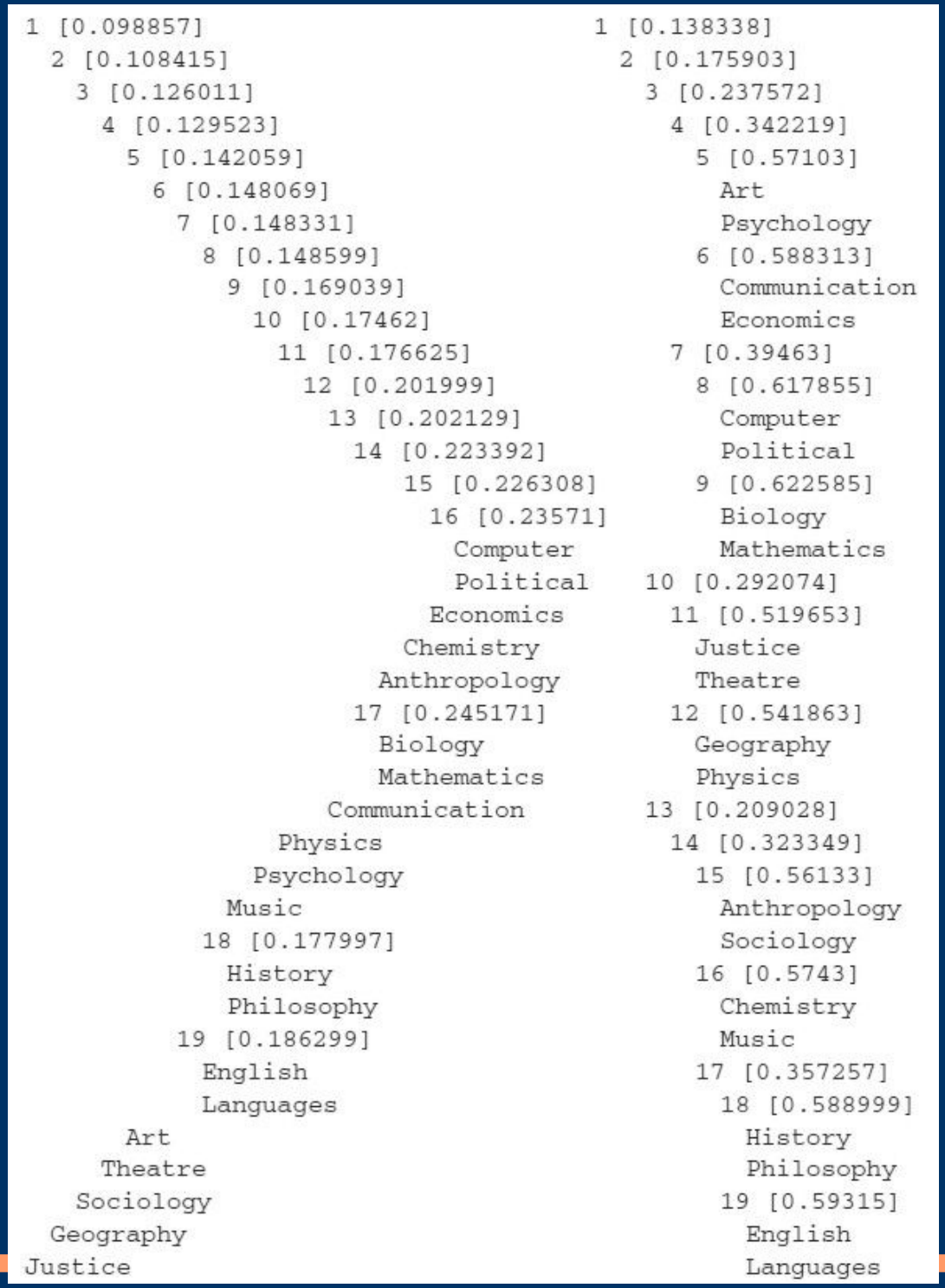
q = 0,04

Średnie podobieństwo międzyklastrowe:

0,4516

Metoda najdalszego sąsiada.

Średnie podobieństwo międzyklastrowe: 0,304475



Metoda średniego podobieństwa.

Średnie podobieństwo międzyklastrowe: 0,434181

Grupowanie hierarchiczne

Metoda najdalszego sąsiada jest skuteczna w przypadku, gdy zbiór obiektów jest skupiony oraz ma wyraźne krawędzie.

Metoda najbliższego sąsiada dobrze radzi sobie w przypadku, gdy grupy mają nieregularne kształty, jednak jest czuła na obiekty typu *outliers* (strony zgodne ze słowami kluczowymi, jednak nieistotne pod względem treści).

Jaki układ obiektów może powodować problemy dla metody najbliższego sąsiada?

Są to obiekty leżące pomiędzy dwoma dobrze odseparowanymi klastrami. Obiekty te mogą tworzyć mosty pomiędzy klastrami.

Algorytm k-średnich

Podział jednego dużego klastra na wiele mniejszych jest dużo prostszy niż scalanie pojedynczych klastrów, ponieważ w tym przypadku nie trzeba badać wszystkich możliwych kandydatów do przydziału do nowego klastra.

Jedynym wymogiem jest znajomość ilości klastrów, które mają powstać w wyniku działania algorytmu.

Idea algorytmu *k-średnich* polega na wykorzystaniu centroidów do reprezentacji danego klastra oraz podziale dużego klastra za pomocą obliczonych centroidów.

Algorytm k-średnich

Schemat działania algorytmu *k-średnich* można przedstawić jako:

1. Wybierz *k* dokumentów będących centroidami.
2. Przyporządkuj dokumenty do centroidów bazując na podobieństwie.
3. Przelicz ponownie centroidy dla każdego klastra.
4. Jeśli centroidy się nie zmieniają, zakończ algorytm.
5. Przejdź do kroku 2.

Najważniejszy w algorytmie jest krok nr 2, gdyż przenoszenie dokumentów pomiędzy klastrami pozwala zwiększać wartość podobieństwa międzyklastrowego.

Algorytm k-średnich

Algorytm *k-średnich* stara się znaleźć ekstremum funkcji celu, która zdefiniowana jest następującą zależnością:

$$J = \sum_{i=1}^{k} \sum_{d_t \in D_i} \mathrm{sim}(c_i, d_t)$$

gdzie: *c* – centroid zbioru dokumentów *D*,

Jaki rodzaj ekstremum stara się znaleźć algorytm *k-średnich*?

Algorytm k-średnich stara się odnaleźć maksimum podanej funkcji celu.

Algorytm k-średnich

Działanie algorytmu *k-średnich* zawsze znajduje maksimum funkcji celu, jednak nigdy nie ma pewności, że jest to maksimum globalne.

Od czego zależy to, czy algorytm znajdzie maksimum lokalne lub maksimum globalne?

Zależy to od początkowo wybranego zbioru centroidów.

W jaki sposób można sobie radzić z tym problemem?

Wykonać algorytm kilkukrotnie i wybrać spośród rozwiązań te, które ma największą wartość podobieństwa pomiędzy klastrami.

Algorytm k-średnich - przykład

Przykład wykonania algorytmu *k-średnich* dla zbioru dokumentów WWW opisujących stronę uczelni *CCSU*, pokazuje, że niezależnie od wybranej wartości współczynnika *k* działanie algorytmu w większości przypadków kończy się po wykonaniu dwóch iteracji. Podczas grupowania brane były pod uwagę wszystkie atrybuty (wszystkie 671 termów).

Takie zachowanie jest normalne dla algorytmu w przypadku danych, które nie tworzą wyraźnych grup.

Algorytm k-średnich

W celu poprawy jakości wyników zwracanych przez algorytm *k-średnich* w odniesieniu do dokumentów sieci WWW należy ograniczyć rozpatrywane termy do takich, które najlepiej reprezentują wszystkie dokumenty należące do zbioru. W tym celu można wykorzystać np. technikę entropii, zwracającą najbardziej optymalny zestaw cech.

Jak można stworzyć zestaw optymalnych termów zgodnych z wybraną tematyką?

Można zwrócić zestaw wyników bazujący na słowach kluczowych z wykorzystaniem wyszukiwarki, a następnie zastosować entropię.

Algorytm k-średnich – przykład

Poprzez wykorzystanie techniki entropii ilość istotnych termów opisujących zbiór dokumentów została zmniejszona do 6. Wybrane zostały termy: *history*, *science*, *research*, *offers*, *students* oraz *hall*.

Algorytm *k-średnich* został uruchomiony z parametrem *k* równym 2.

Wybór centroidów został dokonany na dwa sposoby:

- najbardziej podobnych dokumentów: *Computer Science* i *Chemistry*, (wartość podobieństwa: 0,995461) – przykład 1,
- najbardziej różnych dokumentów: *Economics* i *Art* (wartość podobieństwa: 0, wektory ortogonalne) – przykład 2.

Algorytm k-średnich – przykład

	history	*science*	*research*	*offers*	*students*	*hall*
Anthropology	0	0.537	0.477	0	0.673	0.177
Art	0	0	0	0.961	0.195	0.196
Biology	0	0.347	0.924	0	0.111	0.112
Chemistry	0	0.975	0	0	0.155	0.158
Communication	0	0	0	0.780	0.626	0
Computer Science	0	0.989	0	0	0.130	0.067
Criminal Justice	0	0	0	0	1	0
Economics	0	0	1	0	0	0
English	0	0	0	0.980	0	0.199
Geography	0	0.849	0	0	0.528	0
History	0.991	0	0	0.135	0	0
Mathematics	0	0.616	0.549	0.490	0.198	0.201
Modern Languages	0	0	0	0.928	0	0.373
Music	0.970	0	0	0	0.170	0.172
Philosophy	0.741	0	0	0.658	0	0.136
Physics	0	0	0.894	0	0.315	0.318
Political Science	0	0.933	0.348	0	0.062	0.063
Psychology	0	0	0.852	0.387	0.313	0.162
Sociology	0	0	0.639	0.570	0.459	0.237
Theatre	0	0	0	0	0.967	0.254

Algorytm k-średnich – przykład 1

Iteration	Cluster *A*	Cluster *B*	Criterion Function J
1	{Computer Science, Political Science}	{Anthropology, Art, Biology, Chemistry, Communication, Criminal Justice, Economics, English, Geography, History, Mathematics, Modern Languages, Music, Philosophy, Physics, Psychology, Sociology, Theatre}	1.93554 (cluster *A*) + 4.54975 (cluster *B*) = 6.48529
2	{Chemistry, Computer Science, Geography, Political Science}	{Anthropology, Art, Biology, Communication, Criminal Justice, Economics, English, History, Mathematics, Modern Languages, Music, Philosophy, Physics, Psychology, Sociology, Theatre}	3.82736 (cluster *A*) + 10.073 (cluster *B*) = 13.9003
3	{Anthropology, Chemistry, Computer Science, Geography, Political Science}	{Art, Biology, Communication, Criminal Justice, Economics, English, History, Mathematics, Modern Languages, Music, Philosophy, Physics, Psychology, Sociology, Theatre}	4.60125 (cluster *A*) + 9.51446 (cluster *B*) = 14.1157

Algorytm k-średnich – przykład 2

Iteration	Cluster *A*	Cluster *B*	Criterion Function J
1	{Anthropology, Biology, Economics, Mathematics, Physics, Political Science, Psychology}	{Art, Chemistry, Communication, Computer Science, Criminal Justice, English, Geography, History, Modern Languages, Music, Philosophy, Sociology, Theatre}	5.04527 (cluster *A*) + 5.99025 (cluster *B*) = 11.0355
2	{Anthropology, Biology, Computer Science, Economics, Mathematics, Physics, Political Science, Psychology, Sociology}	{Art, Chemistry, Communication, Criminal Justice, English, Geography, History, Modern Languages, Music, Philosophy, Theatre}	7.23827 (cluster *A*) + 6.70864 (cluster *B*) = 13.9469
3	{Anthropology, Biology, Chemistry, Computer Science, Economics, Geography, Mathematics, Physics, Political Science, Psychology, Sociology}	{Art, Communication, Criminal Justice, English, History, Modern Languages, Music, Philosophy, Theatre}	8.53381 (cluster *A*) + 6.12743 (cluster *B*) = 14.6612

Algorytm k-średnich – przykład

Przykład nr 1 pokazuje niewłaściwy dobór centroidów początkowych, gdyż w ostatecznym kroku stworzone zostały dwa nierówne klastry: pierwszy zawierający niewielką ilość dokumentów oraz drugi, składający się z wielu rozproszonych dokumentów, z których wiele jest ortogonalnych względem siebie. Początkowe centroidy trafiły do jednego klastra.

Przykład 2 pokazuje lepsze wyniki, gdyż klastry są bardziej równomierne i składają się z bardziej podobnych dokumentów – klaster *A* zawiera nauki ścisłe, zaś klaster *B* składa się z nauk humanistycznych.

Algorytm k-średnich

Algorytm *k-średnich* jest często wykorzystywany w grupowaniu danych ze względu na jego prostotę oraz szybkość działania, jednak wymaga zdefiniowania dobrze odseparowanych centroidów.

W jaki sposób w sieciach WWW można otrzymać dobre wyniki przy zastosowaniu algorytmu *k-średnich*?

Wymagane jest ręczne zdefiniowane ortogonalnych względem siebie centroidów i następnie wykonanie algorytmu.

Grupowanie – miary podobieństwa

W przypadku sieci WWW można wykorzystać dodatkowe miary podobieństwa pomiędzy stronami WWW na bazie połączeń pomiędzy poszczególnymi dokumentami:

- długość najkrótszej ścieżki pomiędzy dwoma dokumentami d_1 i d_2,
- ilość stron wskazujących zarówno na d_1 jak i d_2,
- ilość stron, na które wskazują jednocześnie linki z d_1 jak i d_2.

Powszechnie stosowanym rozwiązaniem jest wykorzystanie średniej podobieństw liczonych na bazie miary cosinusowej (termów) oraz miary obliczanej za pomocą linków.

Podsmowanie

Algorytmy grupowania danych mogą być wykorzystywane w procesie tworzenia *Topic Directories*.

Najlepsze wyniki uzyskiwane są dla dokumentów tworzących dobrze odseparowane grupy. Równie ważny jest wybór takich termów, które niosą największą ilość informacji.

Istotny jest również dobór odpowiednich miar metrycznych oraz sposobu obliczania podobieństwa pomiędzy dwoma klastrami.

Dziękuję za uwagę!