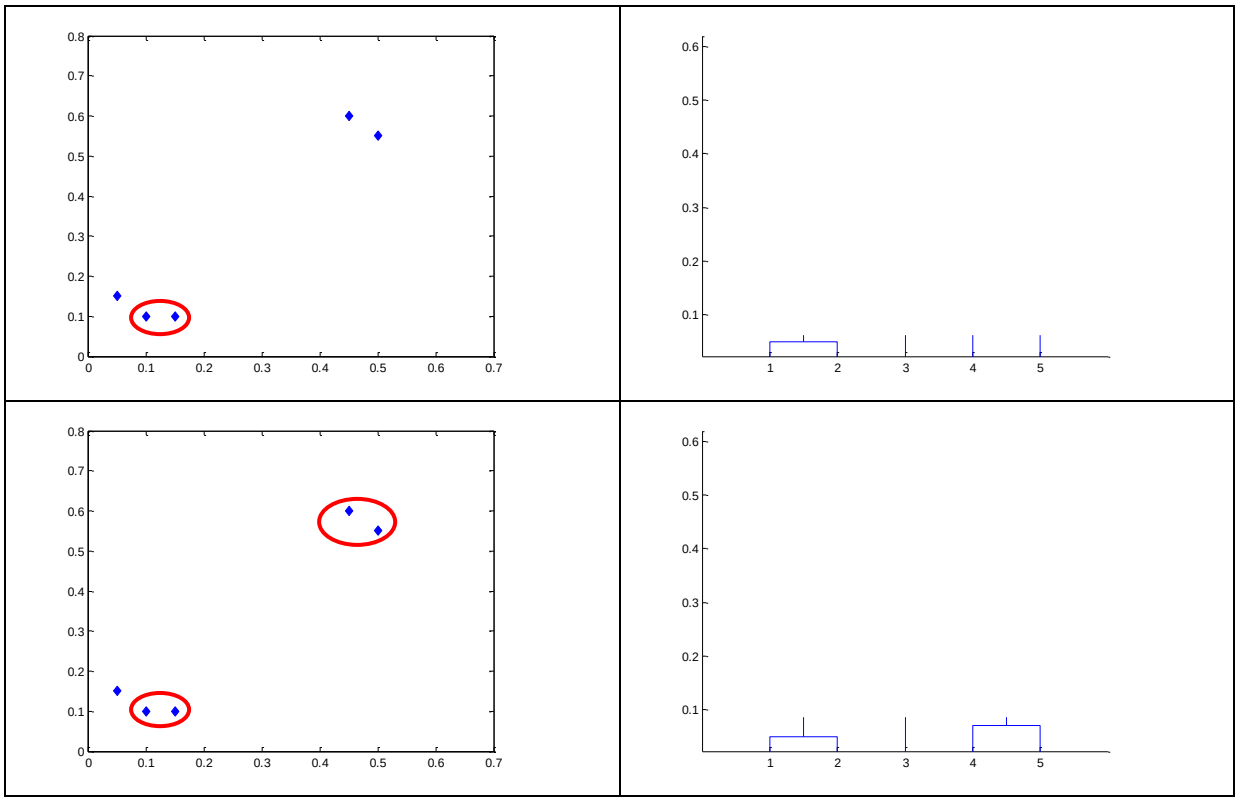
Grupowanie cz. II – Grupowanie hierarchiczne

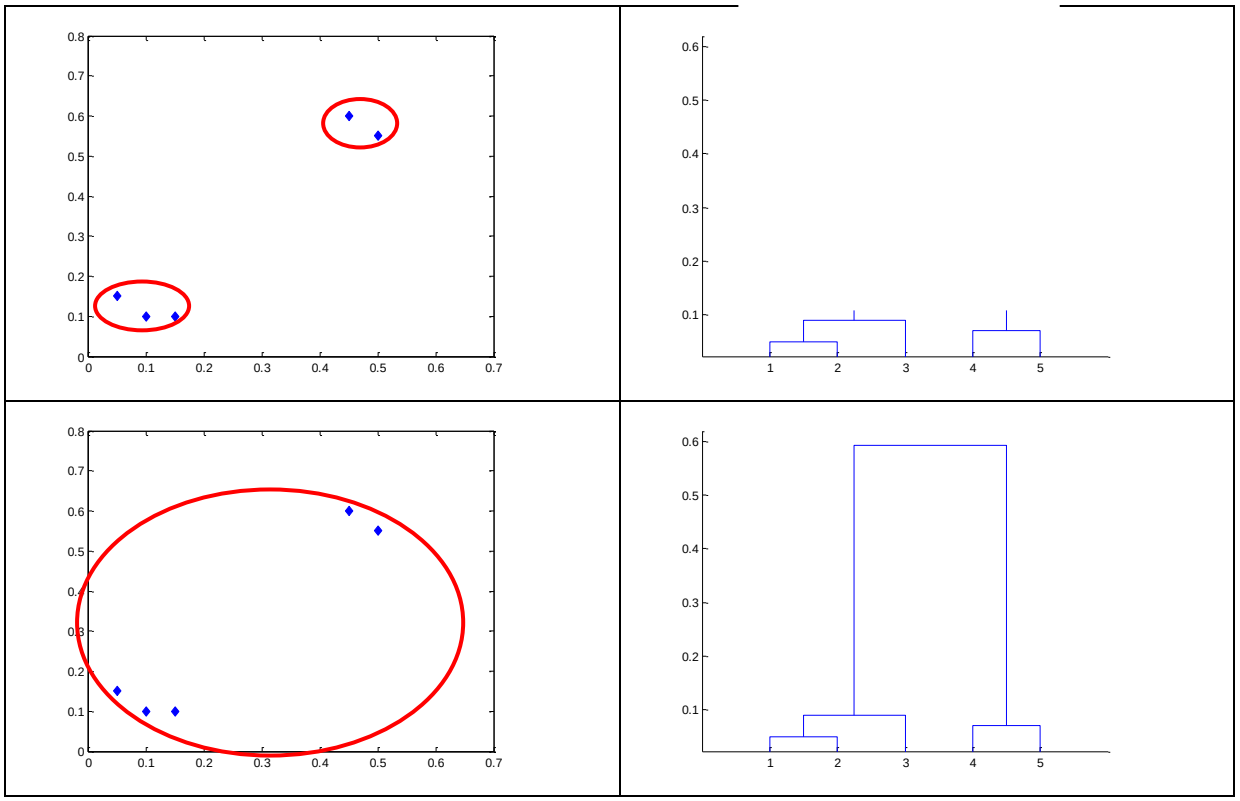
Podstawy teoretyczne

Grupowanie hierarchiczne jest odmienną grupą metod grupowania danych w stosunku do metod bazujących na minimalizacji skalarnego współczynnika jakości jak np. algorytm k-means. Metoda ta bazuje na budowie grafu w postaci drzewa. Algorytm ten jest typem algorytmów z dołu do góry „buttom – top" gdzie zakłada się, iż każdy wektor stanowi oddzielny klaster, a następnie łączy się małe klastry w coraz to większe. Proces łączenia realizowany jest na zasadzie poszukiwania klasterów leżących najbliżej siebie i zastępowania ich nowym większym klastrem, stanowiącym połączenie dwóch poprzednich. Proces ten stopniowo postępuje aż do chwili, w której zostanie osiągnięta właściwa liczba klastrów (określona przez użytkownika) lub do momentu gdy wszystkie wektory znajdują się w jednym klastrze.

Charakterystyczną cechą tego algorytmu jest momzliwość reprezentacji struktury klasteryzacji w postaci drzewa dendrogramu. Dendrogram na osi x posiada etykiety punktów, natomiast na osi y podobieństwo między grupami. Taka reprezentacja wyników klasteryzacji daje szerego możliwości jak np. ocenę liczby klastrów (jeśli wcześniej liczba ta jest nie znana), możliwość analizy pojawienia się wektorów odstających itp.

Przykład procesu grupowania hierarchicznego:





Proces klasteryzacji w tym algorytmie można podzielić na trzy etapy:

1. Liczenie odległości
2. Budowa drzewa na podstawie odległości
3. Przycięcie drzewa na określonym poziomie

Gdzie w drugim etapie jednym z parametrów jest określenie sposobu liczenia podobieństwa pomiędzy poszczególnymi grupami.

Typowymi parametrami są tutaj:

- Uśredniona wartość odległości pomiędzy wektorami w grupach

$$d(r,s) = \frac{1}{n_r n_s} \sum_{i=1}^{n_r} \sum_{j=1}^{n_s} dist(x_{ri}, x_{sj})$$

- Wyszukiwanie dwóch najodleglejszych obiektów w klastrach

$$d(r,s) = max(dist(x_{ri}, x_{sj})), i \in (1, \ldots, n_r), j \in (1, \ldots, n_s)$$

- Wyszukiwanie dwóch najbardziej podobnych obiektów w klastrach

$$d(r,s) = min(dist(x_{ri}, x_{sj})), i \in (i, \ldots, n_r), j \in (1, \ldots, n_s)$$

- Wyznaczenie centroidy dla każdej grupy I liczenie odległości pomiędzy centroidami

$$d(r,s) = \left\| \bar{x}_r - \bar{x}_s \right\|_2 \text{ gdzie } \bar{x}_r = \frac{1}{n_r} \sum_{i=1}^{n} x_{ri}$$

- Wyznaczenie centroidy dla każdej grupy I liczenie odległości pomiędzy centroidami, jednak centoida wyznaczana jest jako średnia centroida z już istniejących centroid (tylko odległośc Euklidesa, duża szybkość)
  $d(r,s) = \|\tilde{x}_r - \tilde{x}_s\|_2$ gdzie $\tilde{x}_r$ i $\tilde{x}_s$ są centroidami klasterów *r* i *s* powstałymi z dwóch centroid *p* i *q* na podstawie których powstał dany klaster $\tilde{x}_r = \frac{1}{2}(\tilde{x}_p + \tilde{x}_q)$

Cechą algorytmu grupowania hierarchicznego jest możliwość wpływu na kształt powstałych grup poprzez wybór odpowiedniego typu łączenia grup. Cechy tej nie posiadają inne algorytmy grupowania, w szczególności bazujące na centroidach jak algorytm k-średnich czy VQ .

W Matlabie

Do klasteryzacji w oparciu o metodę hierarchiczną służy w Matlabie funkcja *clusterdata(),* która w rzeczywistości wywołuje funkcje:

- *D = pdist(dane)* – funkcja licząca odległości miedzy wektorami
- *Z = linkage(D,'typ_łączenia_klasterów')* – funkcja tworząca strukturę drzewiastą, jako argumenty wywołania przyjmuje ona wektor odległości oraz sposób liczenia odległości pomiędzy grupami opisany powyżej. Argument ten przekazywany jest w postaci napisu:
  - 'average' - Uśredniona wartość odległości pomiędzy wektorami w grupach
  - 'complete' - Wyszukiwanie dwóch najodleglejszych obiektów w klastrach
  - 'single' - Wyszukiwanie dwóch najbardziej podobnych obiektów w klastrach
  - 'centroid' - Wyznaczenie centroidy dla każdej grupy I liczenie odległości pomiędzy centroidami
  - 'median' - Wyznaczenie centroidy dla każdej grupy I liczenie odległości pomiędzy centroidami, jednak centoida wyznaczana jest jako średnia centroida z już istniejących centroid (tylko odległośc Euklidesa, duża szybkość)

- *cluster*(Z,'Sposób_przycięcia',parametr)– funkcja przycinająca strukturę drzew i zwracająca wynik klasteryzacji (przynależność danego wektora do danego klastera). Funkcja umozliwia wykorzystanie różnych metod określających liczbę powstałych grup. Najprostszą i najczęściej stosowaną jest 'maxclust' umozliwiająca ręczne zdefiniowanie liczby grup.
- *dendrogram* – funkcja rysująca dendrogram na podstawie wyników funkcji *linkage*

linkage – jak wspomniano funkcja służy budowie struktury drzewiastej,

Przykładowy proces klasteryzacji za pomocą metody hierarchicznej powinien wiec wyglądać:

```
D = load('.....');
d  = pdist(D);
L = linkage(d,'typ_łączenia_grup');
C = cluster(L,'maxclust',3);
```

Zadania

1. Zbadaj wpływ metody łączenia klasterów (linkage) na wyniki klasteryzacji, kształt dendrogramu oraz postać klasterów, badania przeprowadź na zbiorze iris34
    a. Dokonaj klasteryzacji na dwa klastry przy metodzie łączenia typu single, complete oraz centroid
       Przeanalizuj wyniki i zastanów się nad ich interpretacją
    b. Dokonaj klasteryzacji z podziałem na 3 klastry
       Skomentuj wyniki w porównaniu do wyników z algorytmu k-śrenich
    c. Ustaw liczbę klastrów na 5 (‘maxclust’) i uruchom algorytm kilkukrotnie, czy wyniki różnią się od siebie?
2. Dokonaj klasteryzacji zbioru spiral500 zmieniając typu funkcji linkującej. (klasteryzacja na 2 grupy). Czy udało ci się uzyskać klastry odpowiadające prawidłowym etykietom?, jeśli tak to przy jakiej konfiguracji
3. Uruchom skrypt scriptTime. Służy on pomiarowi czasu klasteryzacji w funkcji rozmiaru zbioru poddanego klasteryzacji. W kolejnych iteracjach pętli for zmieniamy liczbę danych poddanych klasteryzacji począwszy od 100 wektorów kończąc na 1000 obiektach poddanych klasteryzacji . Skomentuj uzyskane wyniki