ALGORYTMICZNA I STATYSTYCZNA ANALIZA DANYCH

8/01/2015

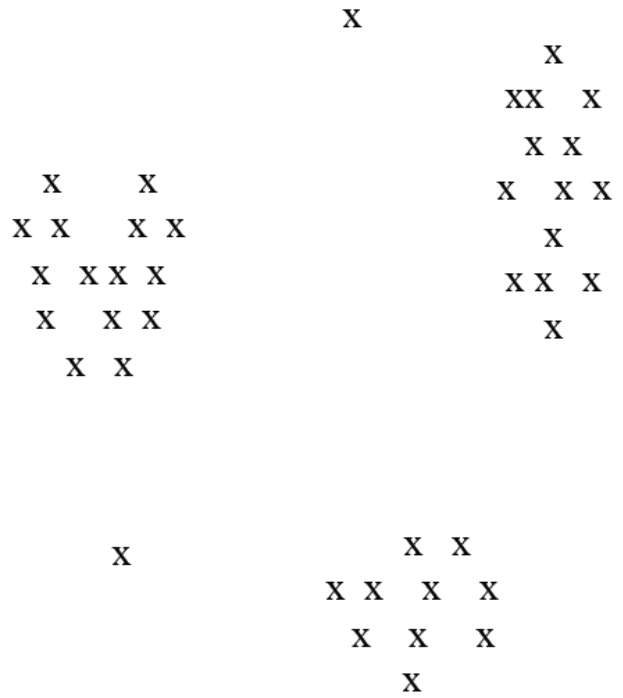
WFAiS UJ, Informatyka Stosowana
II stopień studiów

Eksploracja danych

Algorytmy klastujące

Problem

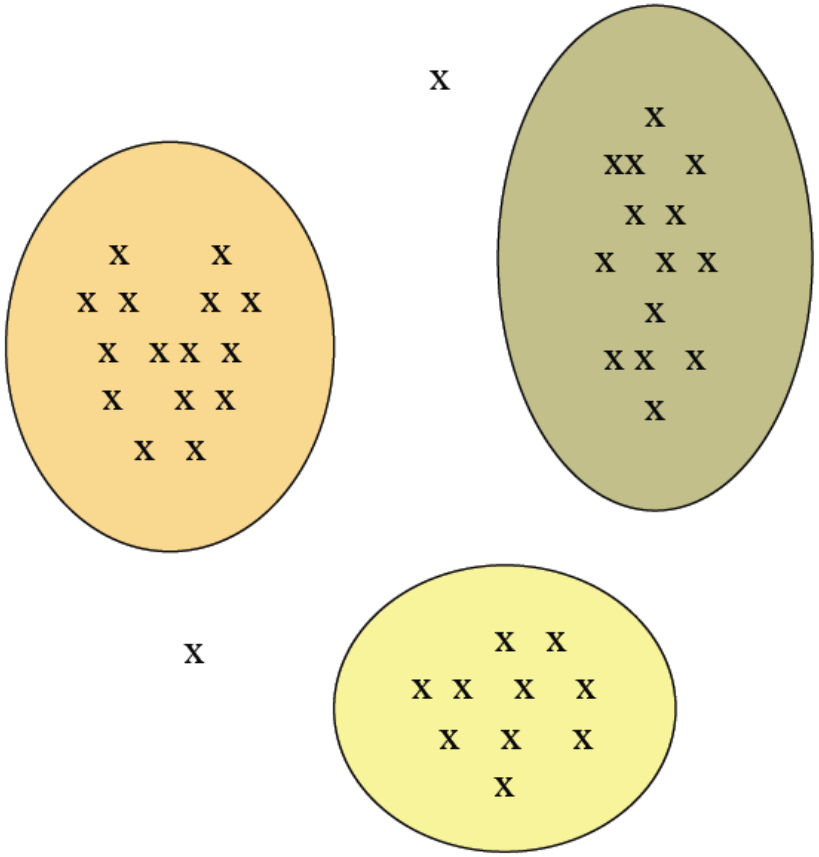
- Mając daną chmurę punktów chcielibyśmy zrozumieć ich strukturę



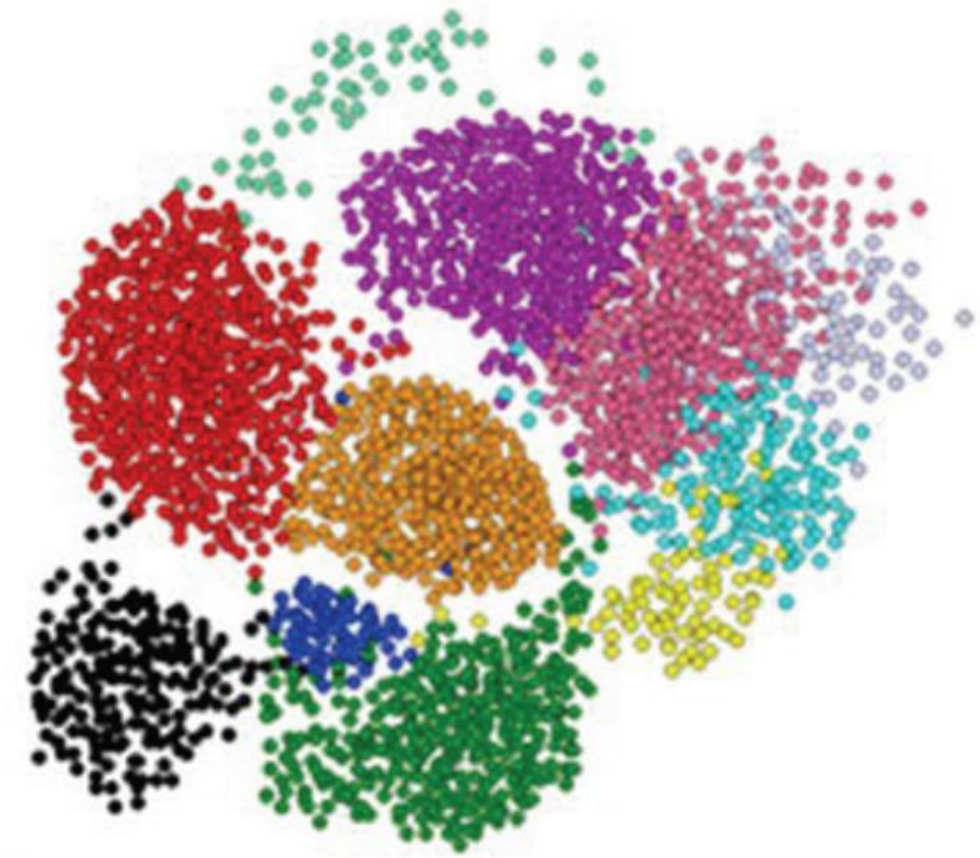
Bardziej formalnie

- Mając dany zbiór punktów pomiarowych oraz podaną miarę odległości pomiędzy punktami pogrupuj dane w klastry
  - Punkty należące do tego samego klastra powinny być ‚podobne” do siebie
  - Punkty należące do dwóch różnych klastrów powinny się istotnie od siebie różnić
- Zazwyczaj:
  - Punkty są w przestrzeni wielowymiarowej
  - Podobieństwo jest zdefiniowane jako miara odległóości pomiędzy punktami
    - Euklidesowa, Cosinus kąta, etc…

Przykład klastrów



Problem jest nietrywialny



Gdzie jest trudność

- Klastrowanie w dwóch wymiarach jest na ogół łatwe
- Klastrowanie małej ilości danych jest na ogół łatwe
- Wiele praktycznych zastosowań dotyczy problemów nie 2 ale 10 lub 10000 wymiarowych.
- W dużej ilości wymiarów trudność polega na fakcie że większość danych jest w „tej samej odległości”.

Przykład: klastowanie obiektów widocznych na niebie: SkyCat

- Katalog 2 bilionów „obiektów” reprezentuje obiekty przez ich częstości emisji promieniowania w 7 zakresach
- Klastrowanie powinno grupować obiekty tego samego typu, np. galaktyki, gwiazdy, kwazary, etc.

Przykład: klastrowanie CD's

- Intuicyjnie: muzyka może być klasyfikowana w kilka kategorii i kupujący na ogół preferują pewne kategorie.
  - Na czym polegają te kategorie?
- A może klastrować CD's ze względu na kategorię osób które je kupują?
- Podobne CD's mają podobną grupę kupujących i odwrotnie

Przykład: klastrowanie dokumentów

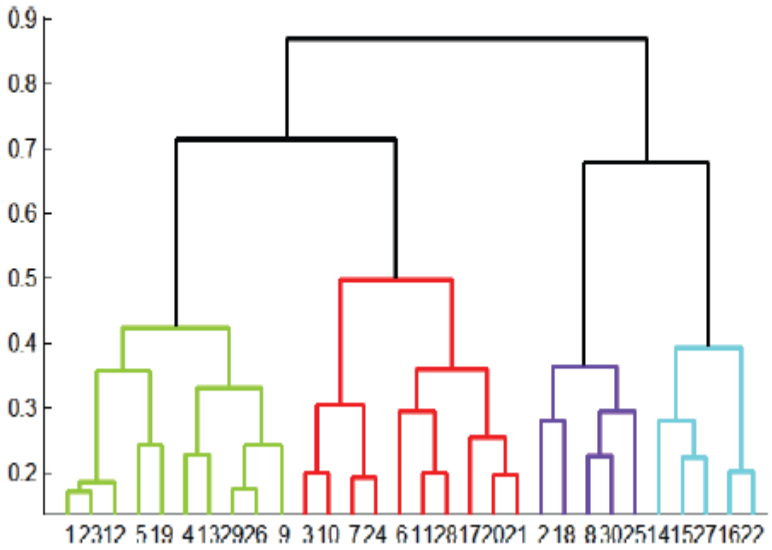
- Problem: pogrupuj razem dokumenty na ten sam temat.
- Dokumenty z podobnym zestawem słów prawdopodobnie są na ten sam temat.
- A może inaczej sformułować: „temat” to podobny zestaw słów który występuje w wielu dokumentach. Więc może należy grupować słowa w klastry i te klastry będą definiować tematy?

Miara odległości

- Dokument: zbiór słów
  - „Jaccard” odległość: podzbiór tych samych elementów
- Dokument: punkt w przestrzeni słów, $(x_1,x_2,\ldots,x_n)$, gdzie $x_i = 1$ jeżeli dane słowo występuje.
  - Euklidesowa odległość
- Dokument: vector w przestrzeni słów $(x_1,x_2,\ldots,x_n)$.
  - Cosinus odległość: iloczyn skalarny unormowanych wektorów

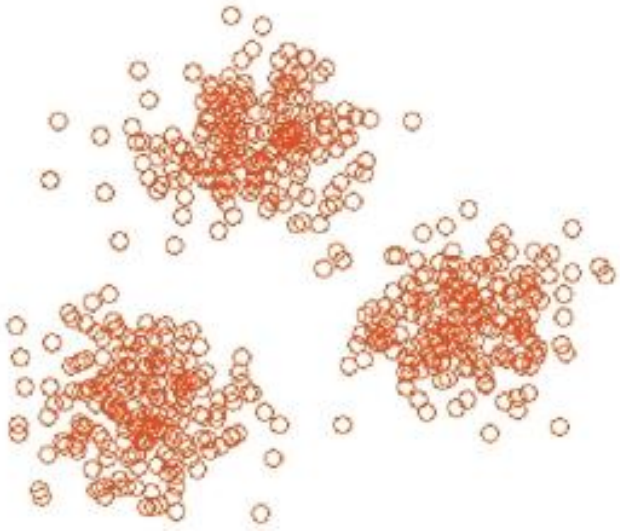
Metody klastrowania

- **Hierarchiczne:**
  - Bottom-up
    - Początkowo każdy punkt jest klastrem
    - Łączymy dwa klastry o najmniejsze odległości w jeden klaster
    - Powtarzamy iteracyjnie
  - Top-down
    - Startujemy z jednego wielkiego klastra
    - Dzielimy na dwa klastry jeżeli spełnione pewne warunki
    - Powtarzamy iteracyjnie aż osiągnięty inny warunek



Metody klastrowania

- **K-means:**
  - Zakładamy na ile klastrów chcemy pogrupować dane.
  - Wybieramy początkowe centra klastrów.
  - Przeglądamy listę punktów, przypisuje do najbliższego klastra.
  - Powtarzamy iteracyjnie po każdej iteracji poprawiając położenie centerów klastrów.

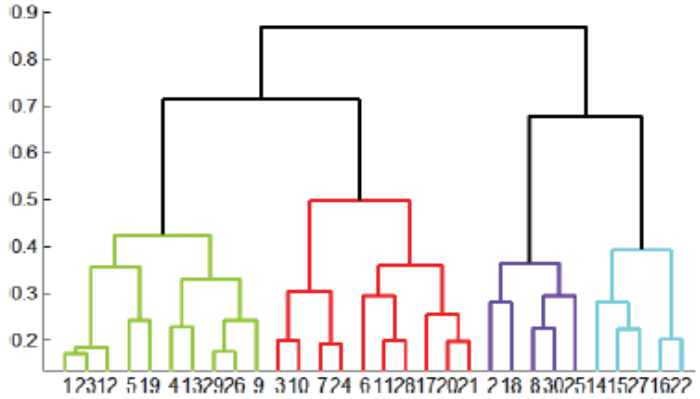


Klastrowanie hierarchiczne

Bottom-up klastrowanie

Klastrowanie hierarchiczne

- Podstawowa operacja: iteracyjnie powtarzaj sklejanie dwóch klastrów w jeden.

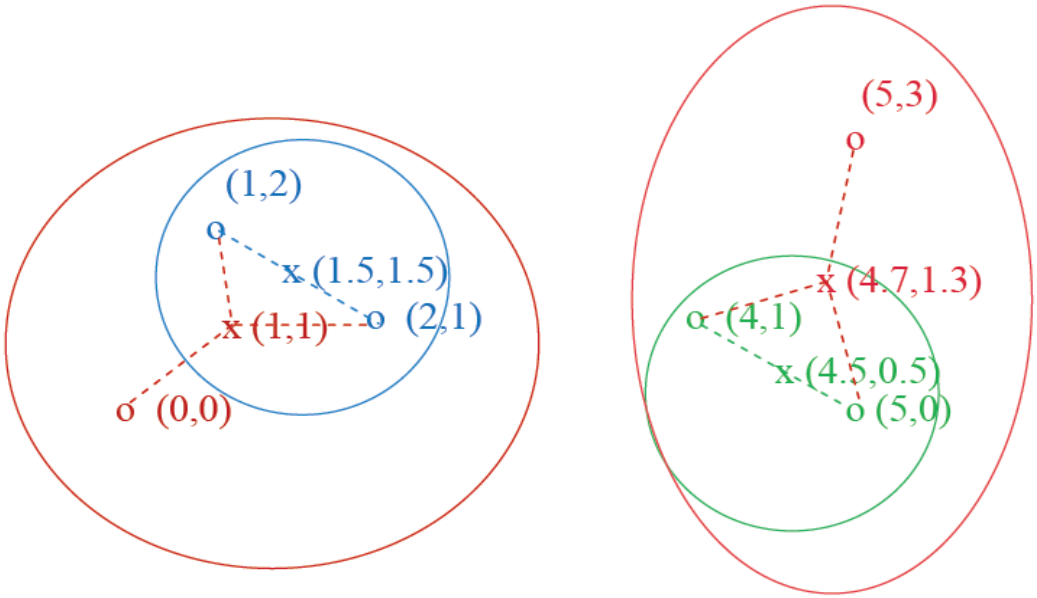


  - Podstawowe pytania:
    - Jak reprezentować klaster który zawiera więcej niż jeden punkt?
    - W jaki sposób zdefiniować „dwa najbliższe” klastry?
    - Kiedy zatrzymać procedurę sklejającą?

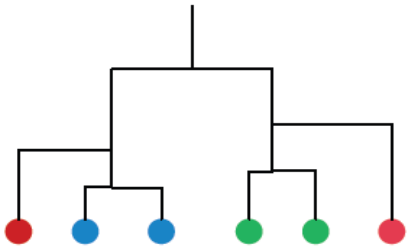
Metryka Euklidesowa

- Jak reprezentować klaster który zawiera więcej niż jeden punkt?
  - Reprezentujemy każdy klaster przez położenie jego środka ciężkości wg. tej metryki. Nazywamy go „centroid”, czyli centrum klastra nie jest jednym z punktów danych.
- W jaki sposób zdefiniować „dwa najbliższe” klastry?
  - Mierzymy odległość pomiędzy centrami klastrów, wybieramy najbliższe.

Przykład:



Data:
o … data point
x … centroid

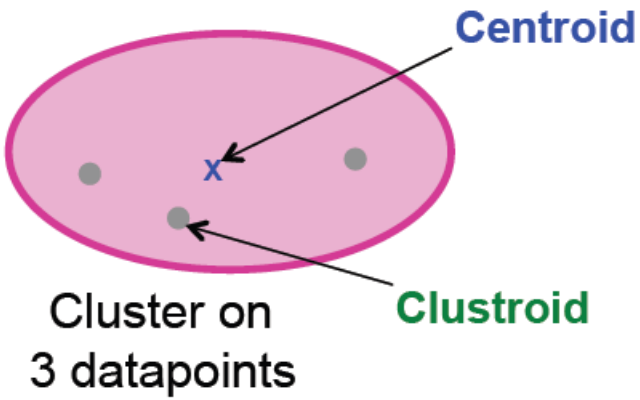


Dendrogram

Metryka nie-euklidesowa

- Jedyne położenia o których możemy mówić to punkty danych. Nie istnieje pojęcie „średniej”.
  - Klaster reprezentowany przez jego punkt będący najbliżej wszystkich innych punktów. Najbliżej wg. zadanej metryki. Nazywamy go „klastroid”.

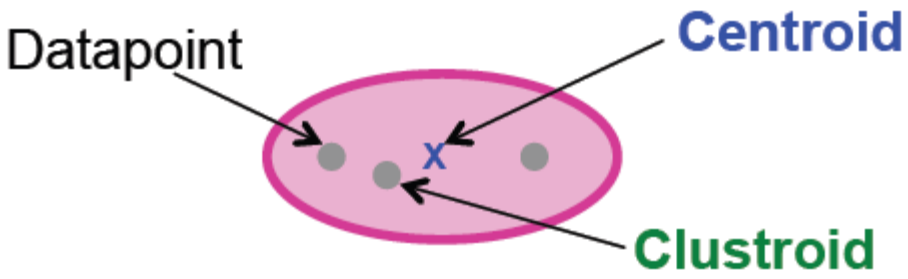
Metryka nie-euklidesowa



Centroid is the avg. of all (data)points in the cluster. This means centroid is an "artificial" point.

Clustroid is an **existing** (data)point that is "closest" to all other points in the cluster.

Metryka nie-euklidesowa



- Klastroid: punkt najbliższy do wszystkich innych.
- Co to znaczy „najbliższy”?
  - Najmniejsza maksymalna odległość od wszystkich innych
  - Najmniejsza suma odległości od wszystkich innych
  - Najmniejsza suma kwadrat ow odległości od wszystkich innych.

Kiedy zakończyć klastrowanie

- Jeżeli oczekujesz że dane powinny się grupować w k –klastrów zakończ jeżeli pogrupujesz w k-klastrów
- Zatrzymaj jeżeli kolejne klastrowanie doprowadza do klastrów o gorszej jakości:
  - np. średnica (maksymalna odległość) większa niż wymagana granica
  - Np. promień klastra większy niż wymagana granica
  - Np. potęga promienia klastra większa niż wymagana granica.

Implementacja

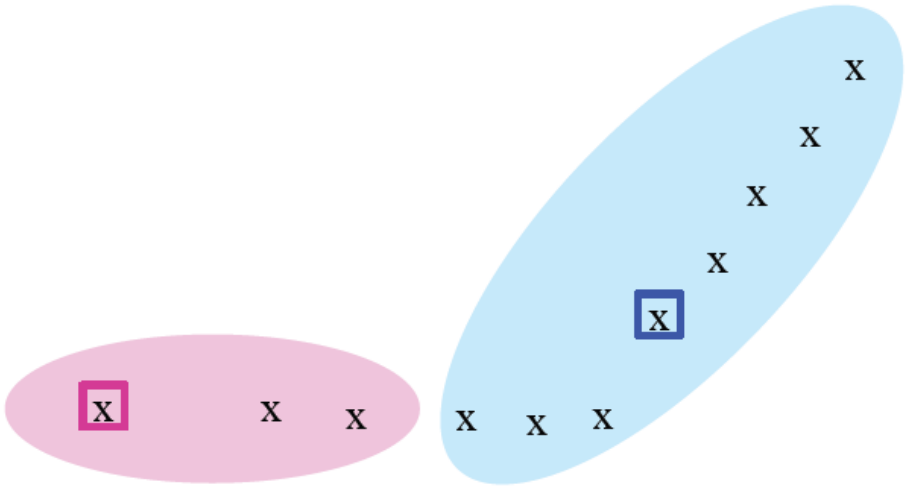
- Naiwna implementacja: w każdym kroku obliczaj odległość każdej pary klastrów a potem sklejaj: złożoność $O(N^3)$ dla każdej iteracji
- Optymalna implementacja z wykorzystaniem kolejki priorytetowej: $O(N^2 \log N)$
  - Dalej złożoność zbyt duża dla dużych zbiorów danych lub danych nie mieszczących się w pamięci.

Klastrowanie k-means

K-mean algorytm

- Zakłada metrykę euklidesową
- W pierwszym kroku wybieramy liczbę k-klastrów
  - Wybieramy k- punktów danych będących reprezentami tych klastrów.
  - Na razie załóżmy że wybieramy je losowo.
- Przeglądamy listę punktów danych i przyporządkowujemy do klastrów
- Updatujemy położenie centrów klastrów na podstawie punktów przypisanych do klastrów
- Powtarzamy operację przeglądania listy punktów, przypisujemy do najbliższych klastrów, punkty mogą migrować pomiędzy klastrami.
- Powtarzamy dopóki punkty nie przestają migrować

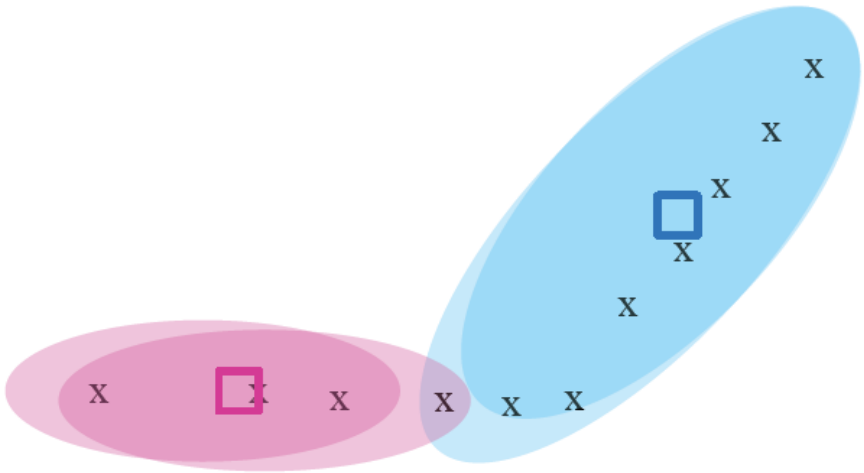
Przykład: k = 2



x … data point
□ … centroid

Round 1

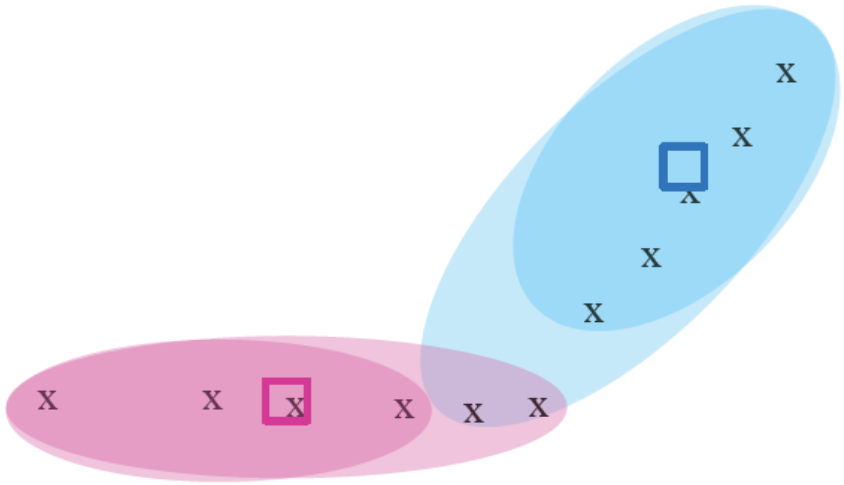
Przykład: przyporządkuj punkty



x … data point
□ … centroid

Round 2

Przykład: popraw centra



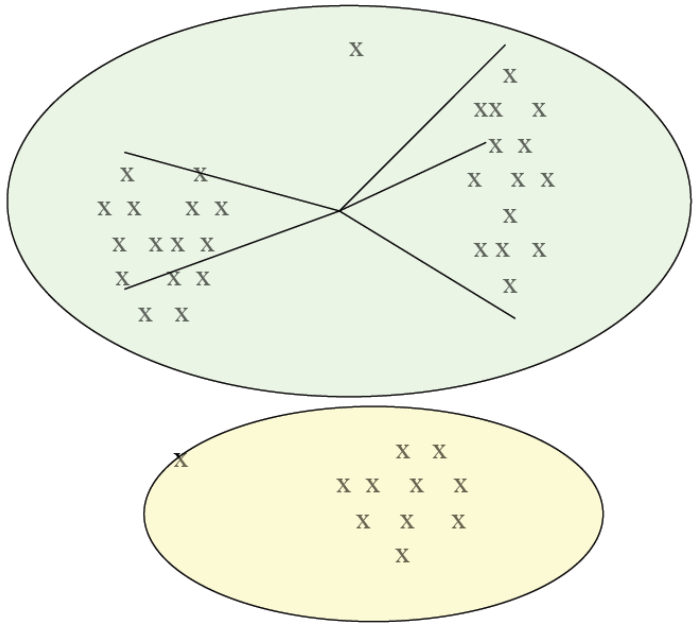
x … data point
□ … centroid

Round 3

W jaki sposób wybrać k

- Spróbuj kilka różnych wartości, badaj jaka jest zmiana średniej odległości jak zwiększasz k

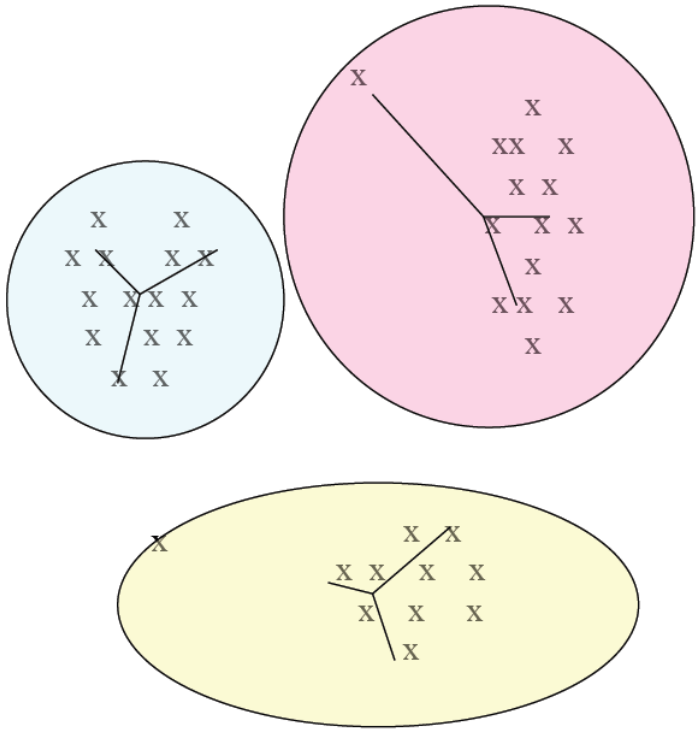
Too few;
many long
distances
to centroid.



W jaki sposób wybrać k

- Spróbuj kilka różnych wartości, badaj jaka jest zmiana średniej odległości jak zwiększasz k

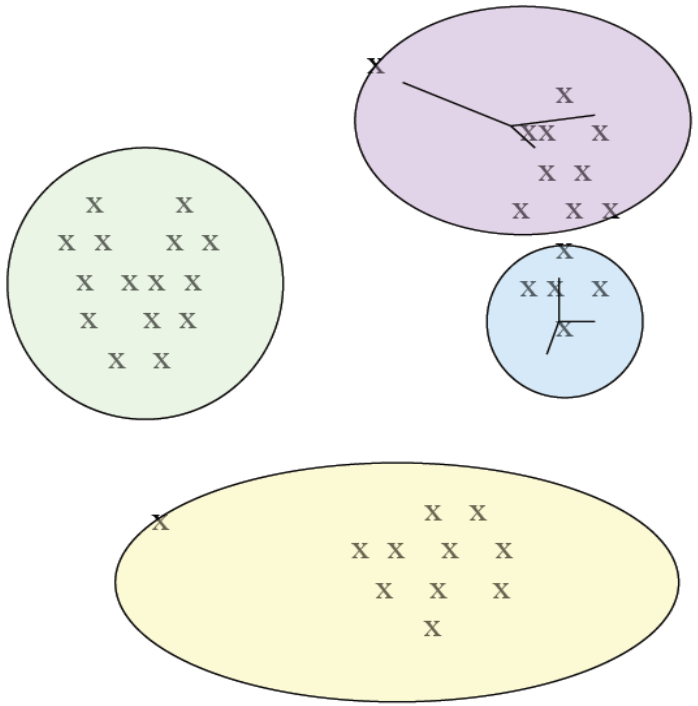
Just right;
distances
rather short.



W jaki sposób wybrać k

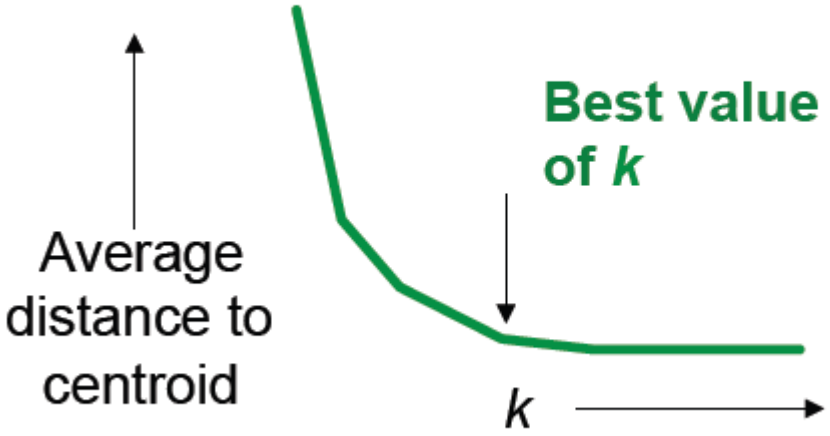
- Spróbuj kilka różnych wartości, badaj jaka jest zmiana średniej odległości jak zwiększasz k

Too many;
little improvement
in average
distance.



W jaki sposób wybrać k

- Spróbuj kilka różnych wartości, badaj jaka jest zmiana średniej odległości jak zwiększasz k



W jaki sposób wybieramy początkowe punkty?

- Na podstawie podzbioru danych:
  - Wybierz podzbiór danych, przeprowadź klastrowanie hierarchiczne aby otrzymać k-klastrów.
  - Wybierz dla każdego punkt najbliższe do środka klastra
  - Użyj tych punktów jako punktów startowych
- Najbardziej dalekie:
  - Wybierz pierwszy punkt losowo
  - Jako następny wybierz najbardziej od niego odległy
  - Powtarzaj zawsze wybierając najbardziej odległy od już wybranych punktów, aż wybierzesz k-punktów

Złożoność obliczeniowa

- Za każdym razem przeglądamy pełna listę punktów aby przypisać punkt do jednego z k-centrów
- Każda iteracja dla N punktów i k-centrów to O(N k)
- Liczna wymaganych iteracji może być bardzo duża.

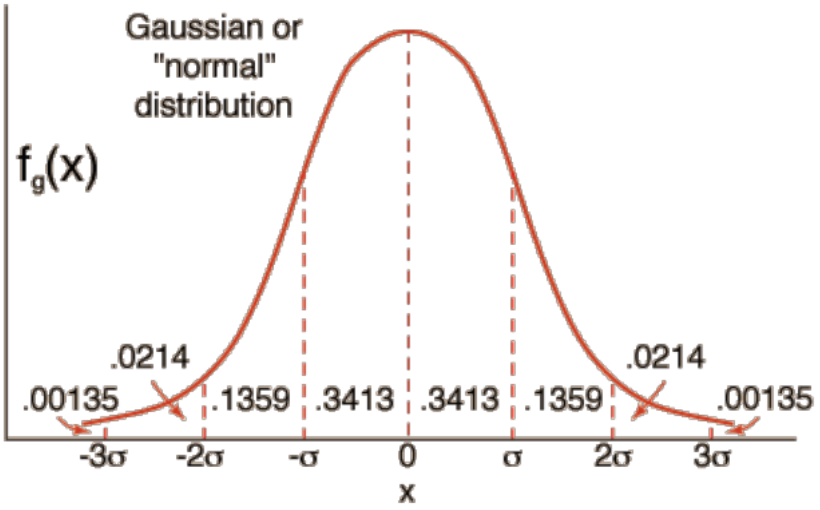
Czy moglibyśmy ten algorytm zrealizować tylko raz przeglądając dane?

Klastrowanie

Algorytm BRF (Bradley- Fayyard-Reina)

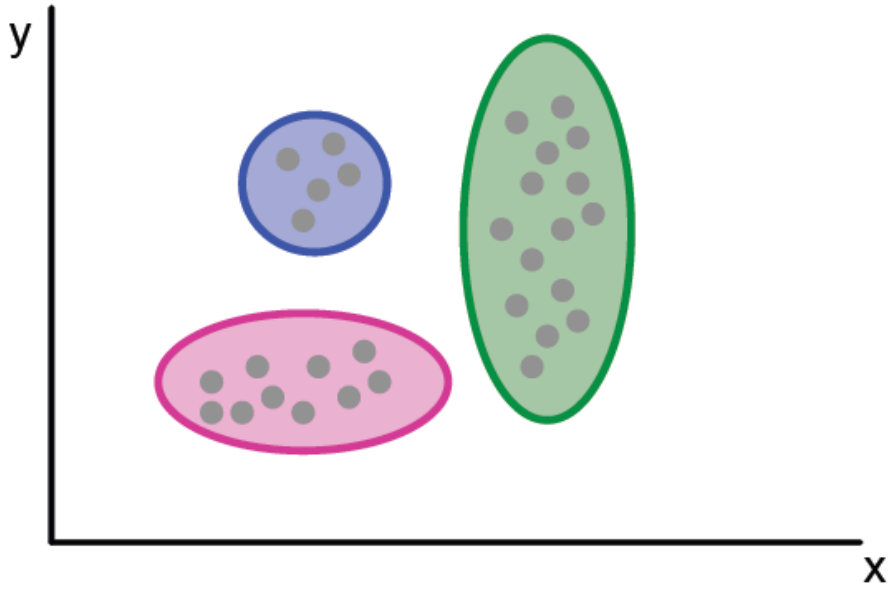
Algorytm BRF

- To jest wersja algorytmu k-means dla bardzo dużych zbiorów danych.
- Zakłada euklidesowa metrykę.
- Zakłada że dane w klastrze mają rozkład normalny względem centrum klastra i każdego wymiaru

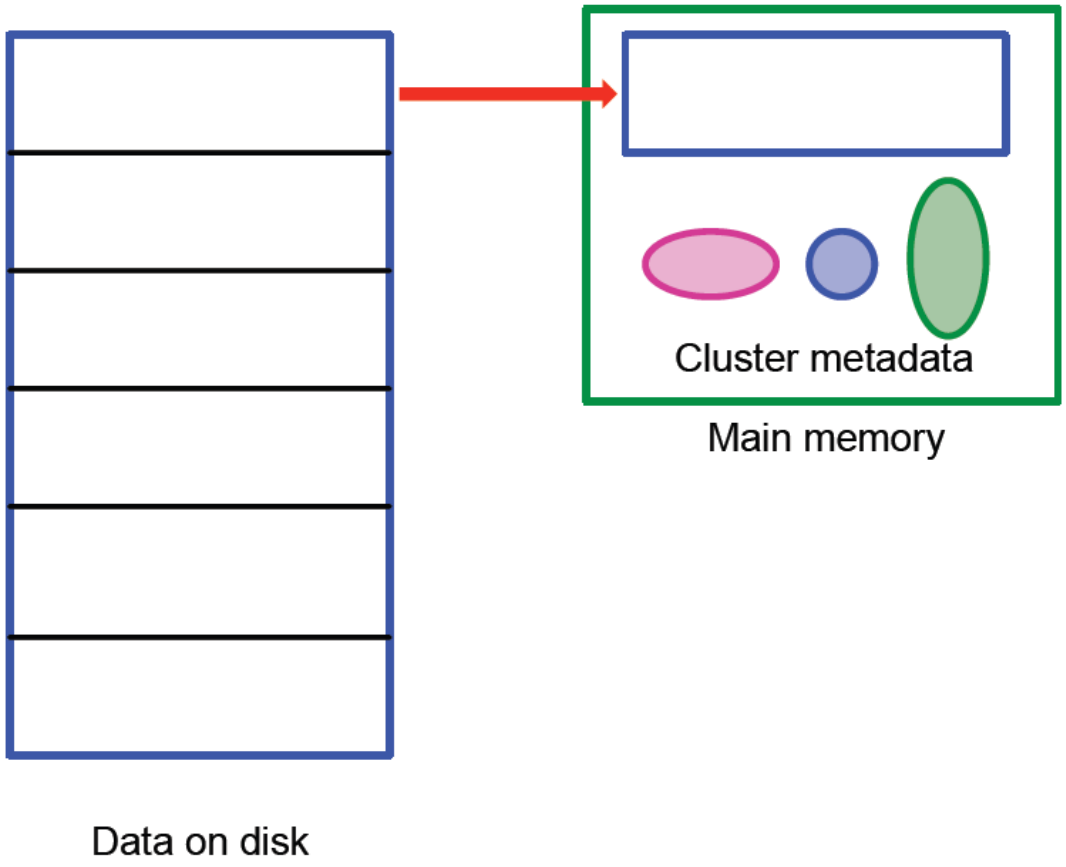


BFR klastry

- Z założenia o normalności rozkładu wynika że klastry wygladają jak elipsy o osiach równoległych do kierunków wymiarów.



BFR algorytm



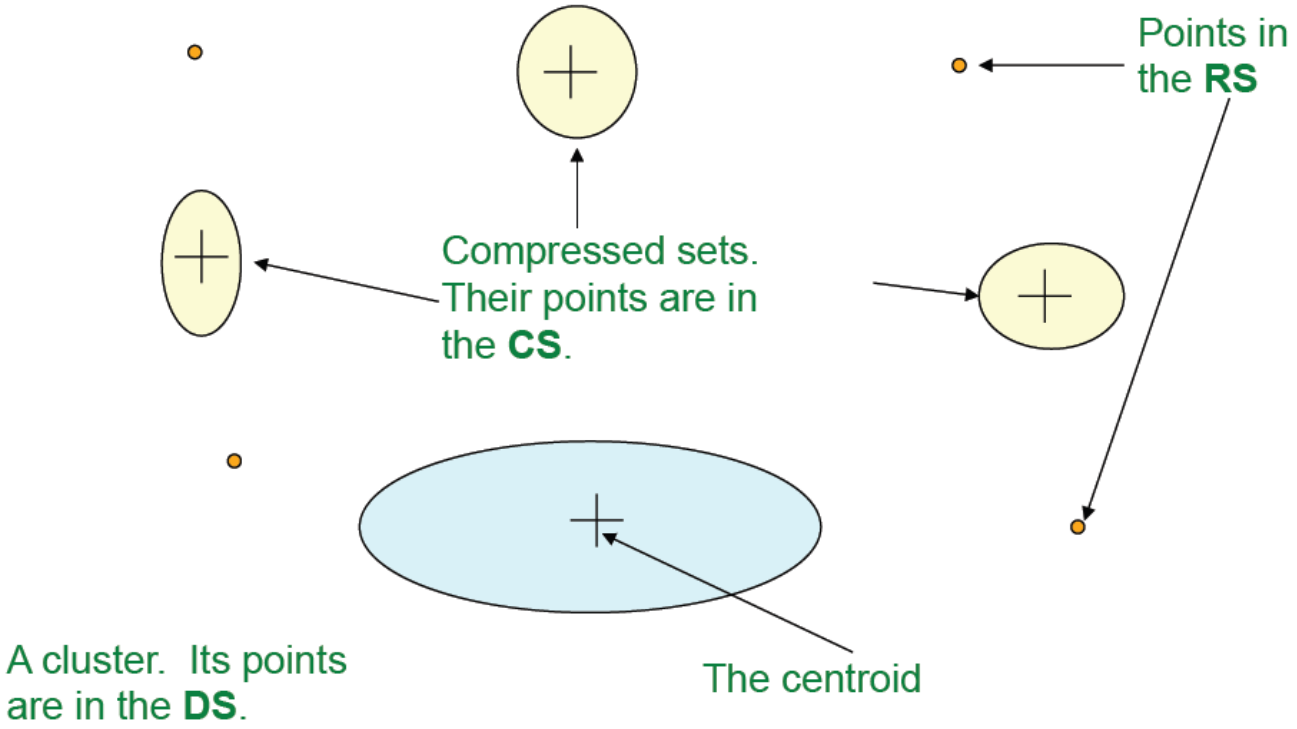
BFR algorytm

- Punkty danych przeglądane są tylko raz, na raz w pamięci tylko podzbiór danych
- Informacja o większości z punktów z każdej partii przechowywana w postaci kilku sumarycznych wielkości statystycznych.
- Na początku, z pierwszej partii punktów wybieramy k-centrów wg. jednej z poznanych metod.

Trzy klasy punktów

- Odrzucone punkty (discard set DS): punkty będące blisko centrów klastrów tak że wystarczy zapisanie informacji sumarycznej
- Punkty w mini-klastrach (compression setCS): punkty które są blisko siebie ale ni blisko żadnego z centrów, zapisujemy tylko informację sumaryczną
- Punkty izolowane (retained set RS): przechowujemy izolowane punkty do następnego etapu.

Trzy klasy punktów



Discard set (DS): Close enough to a centroid to be summarized
Compression set (CS): Summarized, but not assigned to a cluster
Retained set (RS): Isolated points

Sumaryczna informacja

- Dla każdego klastra (DS) i każdego mini-klastra (CS) przechowywana jest informacja sumaryczna:
  - Liczba punktów N
  - Wektor d-wymiarowy SUM, każda współrzędna to suma odległości punktów klastra od centrum w danym wymiarze
  - Wektor d-wymiarowy SUMSQ, każda współrzędna to suma kwadratów odległości punktów klastra od centrum w danym wymiarze

Sumaryczna informacja

- 2d + 1 wartości reprezentuje każdy klaster i mini-klaster
- Średnia w każdym wymiarze (centroid) może być przeliczona jako SUM_i/N
- Wariancja w kaśdym wymiarze może być przeliczona jako $(SUMSQ_i/N) – (SUM_i/N)2$

Procesowanie porcji danych

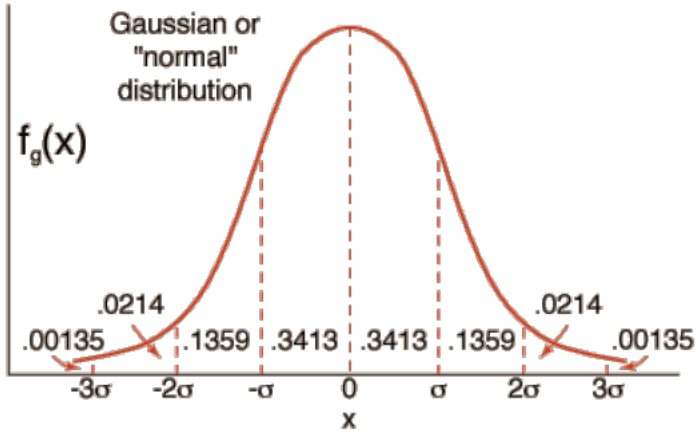
- Sprawdź czy punkt jest „wystarczajaco blisko” do DS lub CS klastra, wybierz najbliższy, dodaj do sumarycznej informacji a następnie usuń punkt z pamięci.
- Jeżeli punkt nie był wystarczająco blisko sprawdź czy możesz utworzyć nowy CS klaster przeglądając RS punkty. Jeżeli nie zapamiętaj ten punkt jako nowy RS punkt.
- Po ostatniej iteracji posklejaj wszystkie CS i RS do najbliższych DS. Ostatecznie utworzone zostanie tylko k klastrów.

Mahalanobis odległość

- Co to znaczy że punkt jest „wystarczająco blisko”?
  - Klaster C ma centroid w $(c_1, c_2, \ldots, c_d)$ i odchylenie standardowe $(\sigma_1, \sigma_2, \ldots, \sigma_d)$
  - Rozważany punkt P = $(x_1, x_2, \ldots, x_d)$
  - Znormalizowana odległość: $y_i = (x_i - c_i) / \sigma_i$
- MD punktu P od klastra C: $\sqrt{\sum_{i=1}^{d} y_i^2}$

Mahalanobis warunek

- Przypuśćmy że punkt P jest w odległości jednego odchylenia standardowego od centrum w każdym wymiarze.
- Każdy y_i = 1 i wówczas MD = sqrt(d)



68% of points have MD $\leq \sqrt{d}$

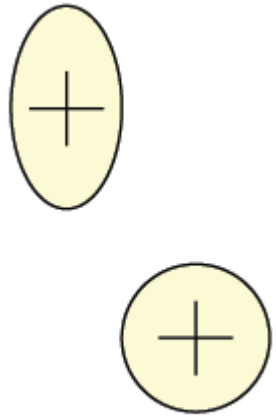
95% of points have MD $\leq 2\sqrt{d}$

99% of points have MD $\leq 3\sqrt{d}$

Akceptuj punkt P do klastra C jeżeli np. wartość MD < 3 sqrt(d)

Kiedy sklejać dwa CS

- Policz wariancję dla sklejonych klastrów, sklej jeżeli wariancja poniżej wartości granicznej.
- Możliwe inne warunki:
  - Gęstość klastra
  - Odległości mogą mieć inną wagę dla każdej współrzędnej
  - Itd..



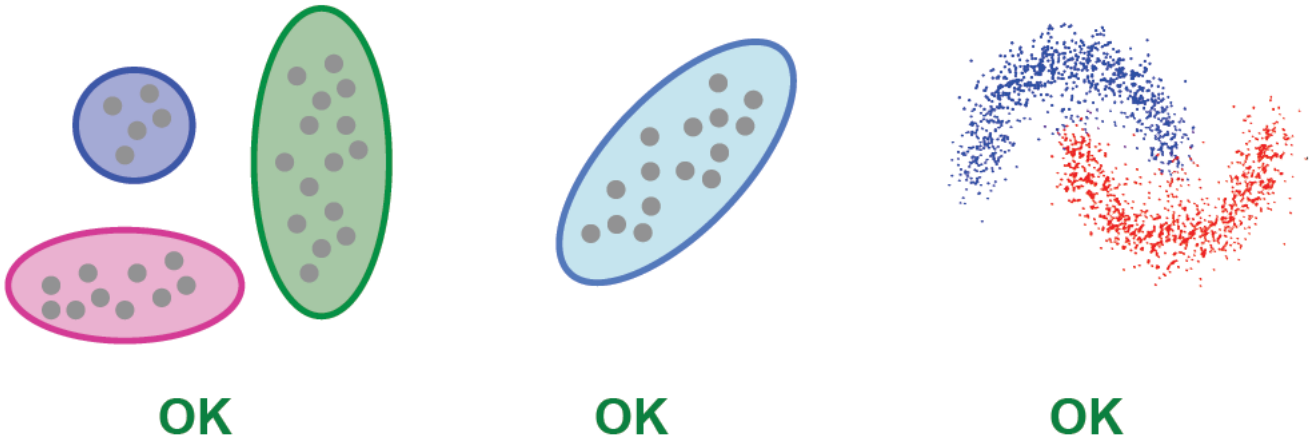
Algorytm CURE

Ograniczenia algorytmu BFR

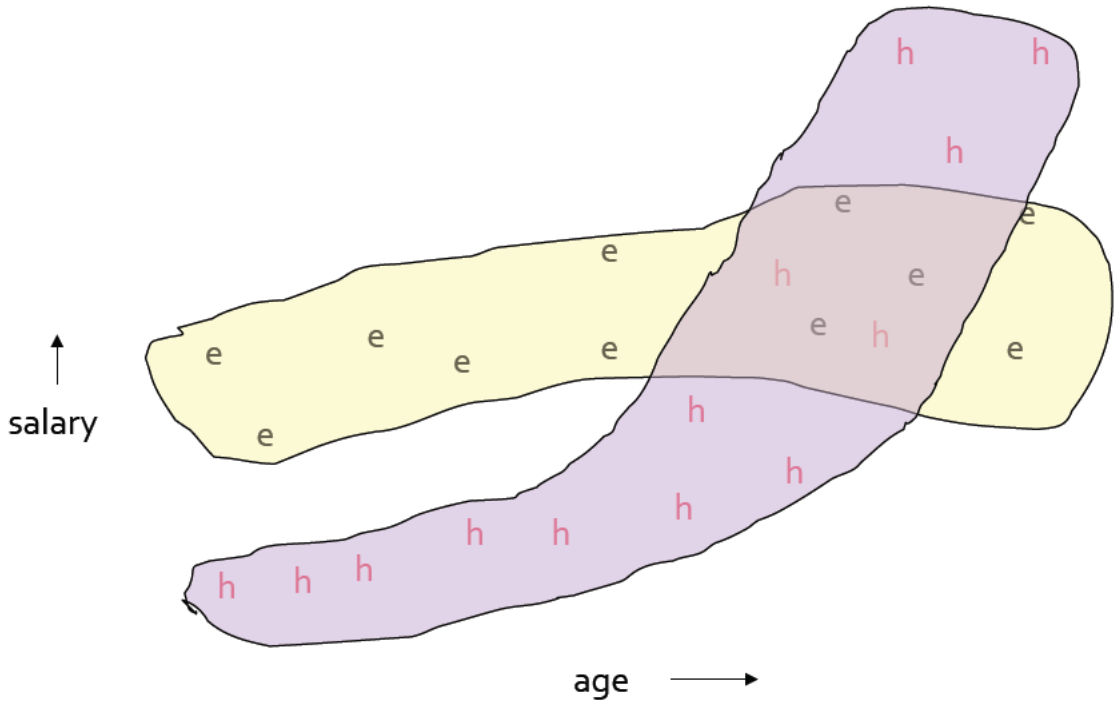
- Silne założenia:
  - Normalny rozkład punktów w każdym wymiarze
  - Osie wzdłuż osi współrzędnych

Algorytm CURE

- CURE (Clustering Usinh Representatives):
  - Zakłada metrykę Euklidesową
  - Dopuszcza klastry różnych kształtów
  - Używa podzbiór punktów do reprezentowania klastra



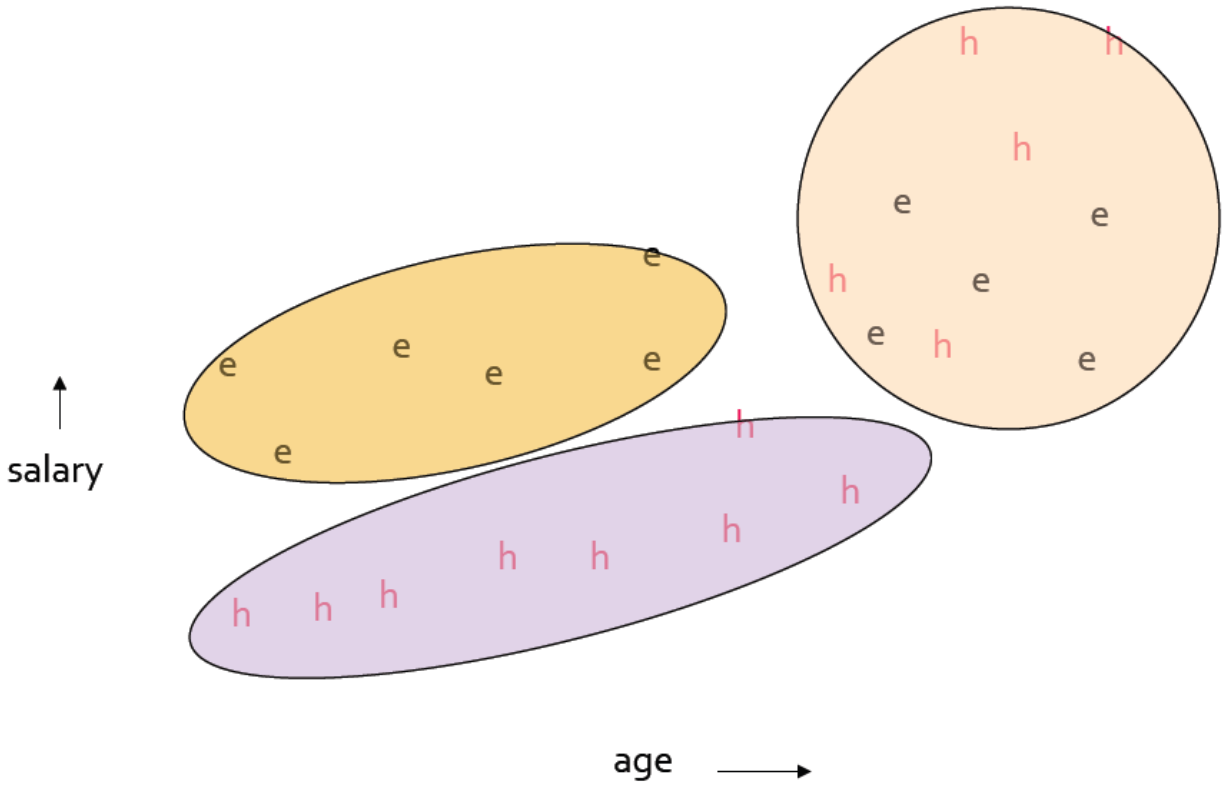
Przykład: płace w Standford



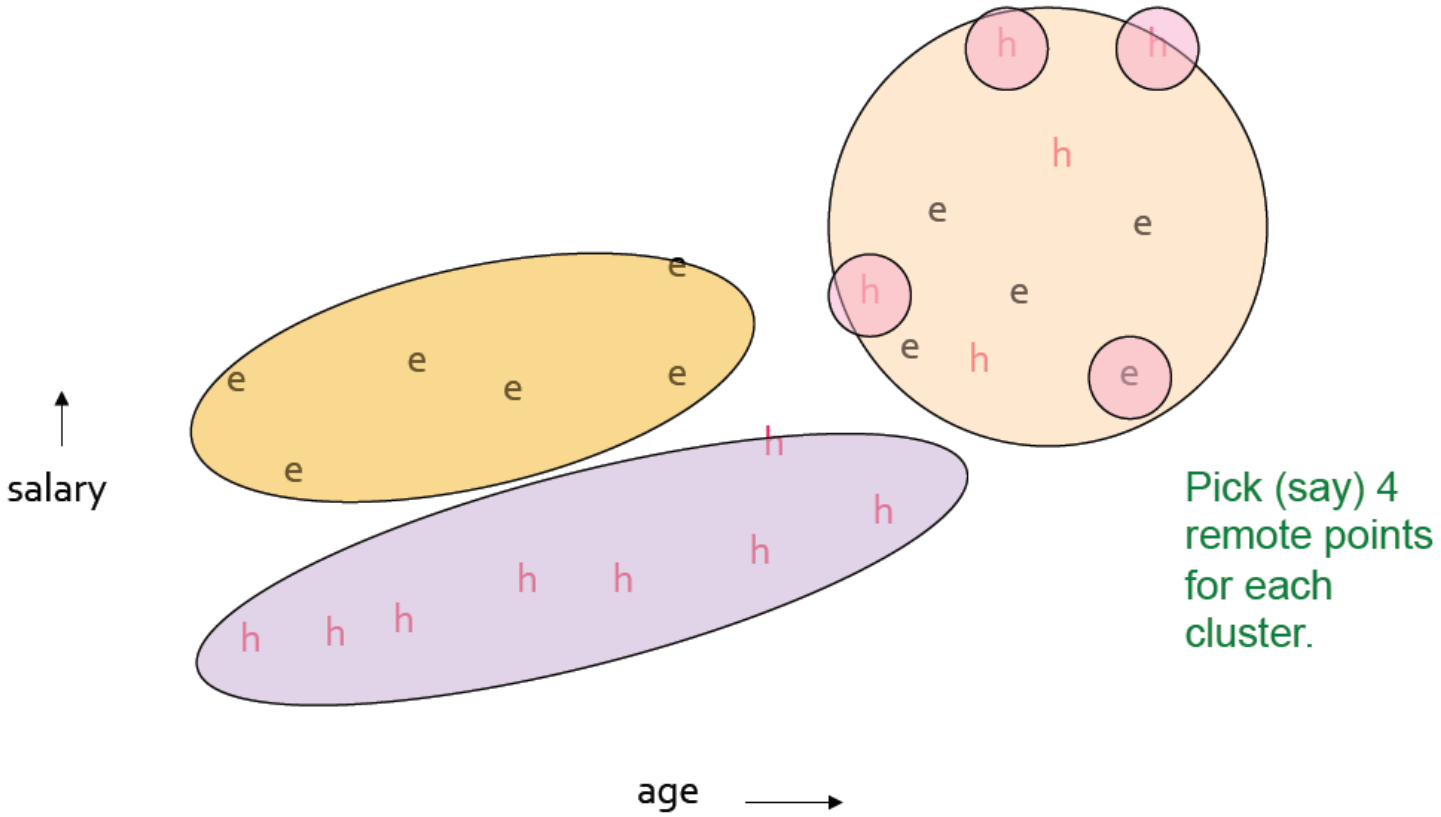
Algorytm CURE

- Pierwsze przejrzenie danych
  - Wybierz podzbiór danych który mieści się w pamięci
  - Przeprowadź klastrowanie hierarchiczne tego podzbioru danych.
  - Wybierz k-punkty reprezentujące klaster (np. k=4), jak najbardziej od siebie odległe
  - Utwórz sztuczne punkty przez przesunięcie wybranych np. o 20% w stronę centrum klastra

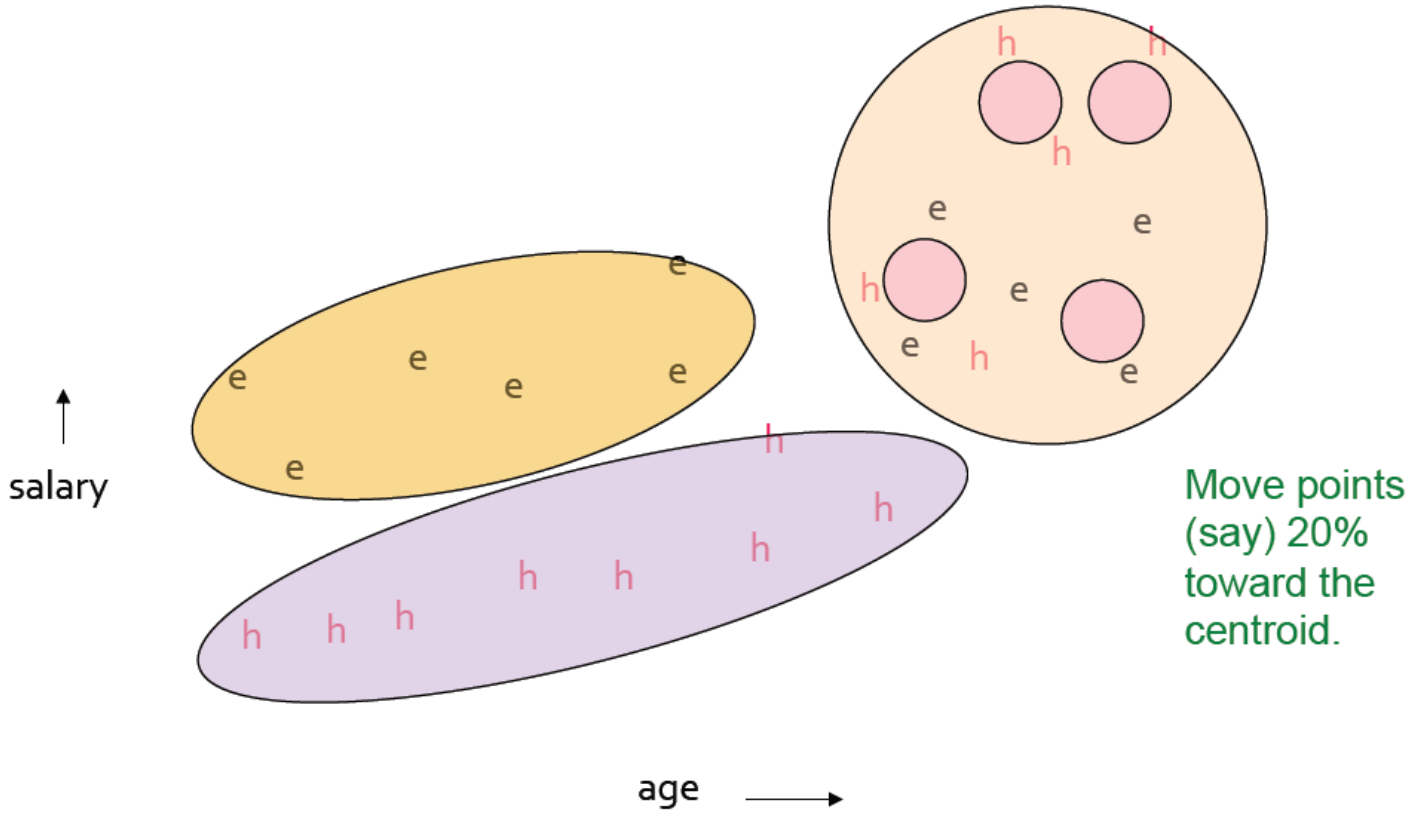
Przykład: początkowe klastry



Przykład: wybór reprezentatywnych punktów



Przykład: wybór reprezentatywnych punktów



Algorytm CURE

- Drugie przejrzenie danych
  - Teraz przejrzyj całość danych
  - Umieść punkt w najbliższym klastrze: do określenia „najbliższy” użyj dla każdego klastra reprezentatywnych punktów

I to już jest koniec!