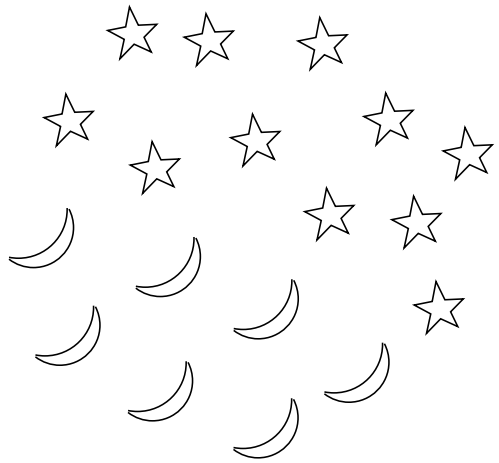
# Sztuczna Inteligencja i Systemy Doradcze

## Uczenie maszynowe - wnioskowanie oparte na podobieństwie

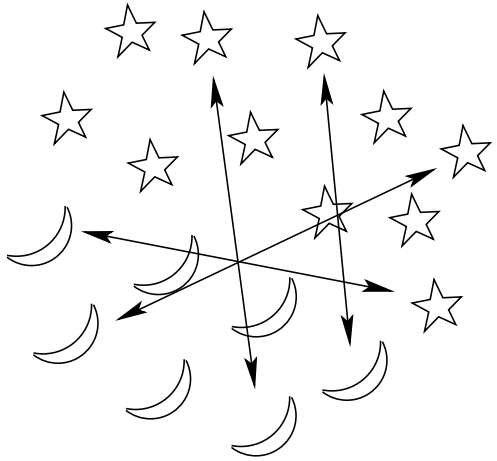
# K najblizszych sasiadow

Główny pomysł: Wszystkie przykłady ze zbioru treningowego $U_{trn}$ są bezpośrednio zapamiętane w bazie przykładów



# K najblizszych sasiadow

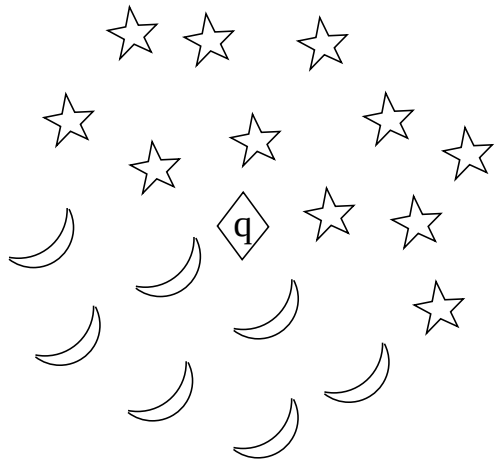
Główny pomysł: Wszystkie przykłady ze zbioru treningowego $U_{trn}$
są bezpośrednio zapamiętane w bazie przykładów



Algorytm uczący indukuje z bazy przykładów miarę odległości $\rho : X^2 \to \mathbf{R}$

# K najblizszych sasiadow

Główny pomysł: Wszystkie przykłady ze zbioru treningowego $U_{trn}$
są bezpośrednio zapamiętane w bazie przykładów

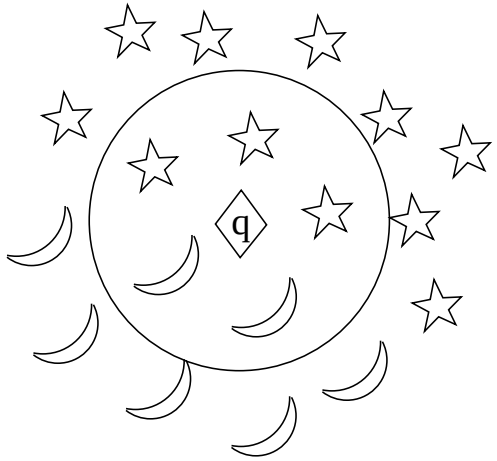



Algorytm uczący indukuje z bazy przykładów miarę odległości $\rho : X^2 \to \mathbf{R}$

Algorytm klasyfikacyjny dla każdego obiektu testowego $q$:

# K najblizszych sasiadow

Główny pomysł: Wszystkie przykłady ze zbioru treningowego $U_{trn}$
są bezpośrednio zapamiętane w bazie przykładów




Algorytm uczący indukuje z bazy przykładów miarę odległości $\rho : X^2 \to \mathbf{R}$

Algorytm klasyfikacyjny dla każdego obiektu testowego $q$:
– wybiera $k$ najbliższych sąsiadów ze zbioru treningowego $U_{trn}$

# K najblizszych sasiadow

Główny pomysł: Wszystkie przykłady ze zbioru treningowego $U_{trn}$
są bezpośrednio zapamiętane w bazie przykładów


Algorytm uczący indukuje z bazy przykładów miarę odległości $\rho : X^2 \to \mathbf{R}$

Algorytm klasyfikacyjny dla każdego obiektu testowego $q$:

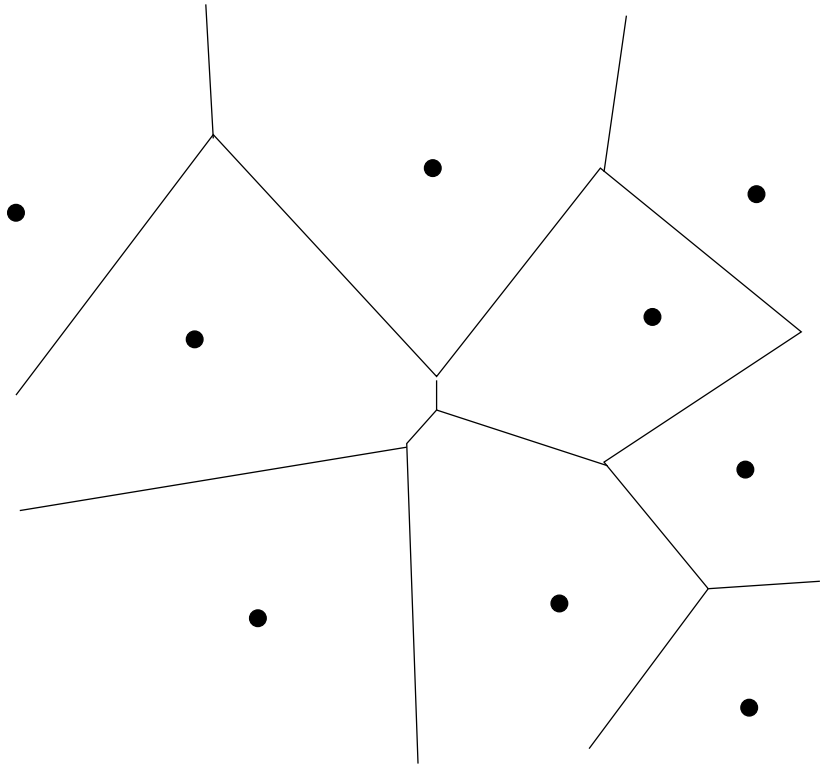
– wybiera $k$ najbliższych sąsiadów ze zbioru treningowego $U_{trn}$

– zwraca decyzję dla $q$ na podstawie decyzji najbliższych sąsiadów

# K najblizszych sasiadow: $k = 1$

Fakt: W przypadku $k = 1$ podział przestrzeni danych
przez zbiór przykładów treningowych tworzy diagram Voronoi

Obiekt testowy $x_q$ jest klasyfikowany z decyzją tego obiektu treningowego, w którego obszar wpada $x_q$:



# K najblizszych sasiadow: $k > 1$

$S(x_q, k)$ — zbiór $k$ najbliższych sąsiadów obiektu $x_q$

Metody głosowania:

♢ jednorodna

$$h(x_q) = \arg\max_{d \in V_d} |\{x \in S(x_q, k) : dec(x) = d\}|$$

♢ ważona odległością

$$h(x_q) = \arg\max_{d \in V_d} \sum_{\substack{x \in S(x_q,k) \\ dec(x)=d}} \frac{1}{\rho(x_q, x)^2}$$

# Miara odleglosci

Miara odległości to pseudometryka $\rho : X^2 \to \mathbf{R}$
definiująca odległość między obiektami w przestrzeni danych $X$
indukowana ze zbioru treningowego $U_{trn}$:

- $\rho(x, y) \geq 0$
- $\rho(x, x) = 0$
- $\rho(x, y) = \rho(y, x)$
- $\rho(x, y) + \rho(y, z) \geq \rho(x, z)$

# Miara odleglosci: rodzaje metryk

$A$ — zbiór atrybutów
$\rho_a$ — miara odległości dla pojedynczego atrybutu $a \in A$

Rodzaje metryk ze względu na łączenie atrybutów:

Miejska Manhattan:

$$\rho(x, y) = \sum_{a \in A} \rho_a(a(x), a(y))$$

Euklidesowa:

$$\rho(x, y) = \left( \sum_{a \in A} \left( \rho_a \left( a(x), a(y) \right) \right)^2 \right)^{\frac{1}{2}}$$

Maksimum:

$$\rho(x, y) = \max_{a \in A} \rho_a(a(x), a(y))$$

# Miara odleglosci: City-Hamming

Dla atrybutu symbolicznego delta Kroneckera:

$$\rho_a = \begin{cases} 0 \text{ jeśli } a(x) = a(y) \\ 1 \text{ wpp.} \end{cases}$$

Dla atrybutu numerycznego różnica wartości:

$$\rho_a(a(x), a(y)) = |a(x) - a(y)|$$

Lepiej stosować normalizowaną różnicę:

$$\rho_a(a(x), a(y)) = \frac{|a(x) - a(y)|}{a_{\max} - a_{\min}} \quad \text{lub}$$

$$\rho_a(a(x), a(y)) = \frac{|a(x) - a(y)|}{2\sigma(a)}$$

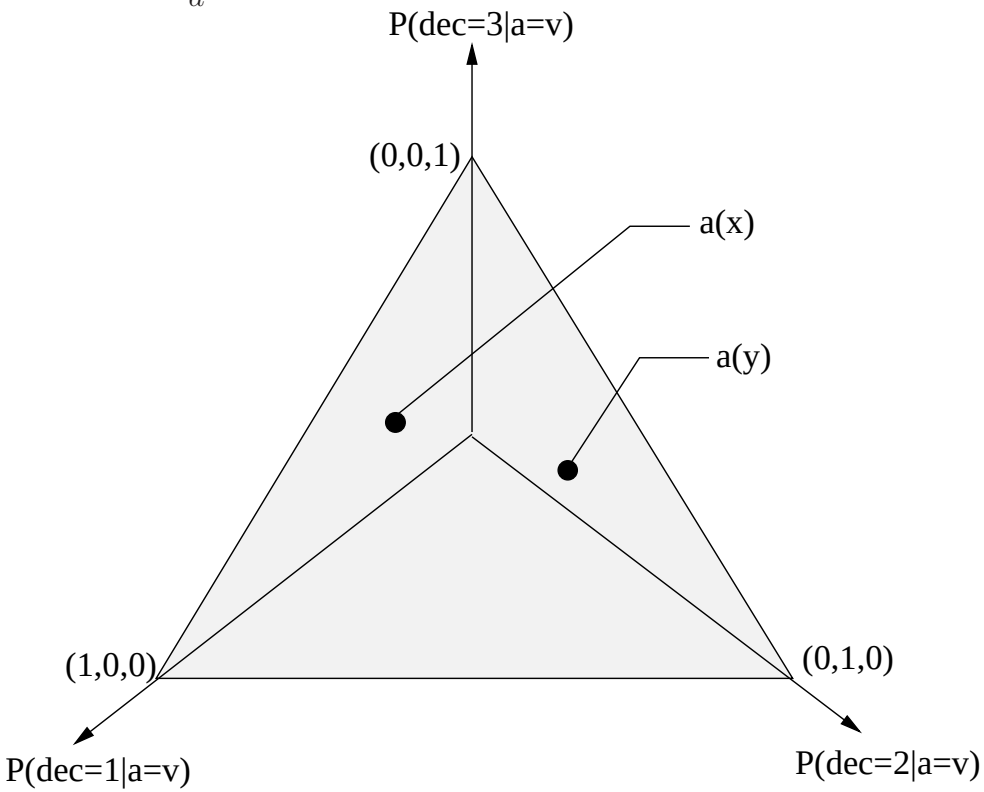
$a_{\max}$ — maksymalna wartość atrybutu $a$ w zbiorze obiektów treningowych
$a_{\min}$ — minimalna wartość atrybutu $a$ w zbiorze obiektów treningowych
$\sigma(a)$ — odchylenie standardowe wartości atrybutu $a$ w zbiorze treningowym

# Miara odleglosci: City-SVD (Domingos 1996)

Dla atrybutu numerycznego normalizowana różnica wartości jak w City-Hamming

Dla atrybutu symbolicznego prosta różnica wartości (ang. SVD):

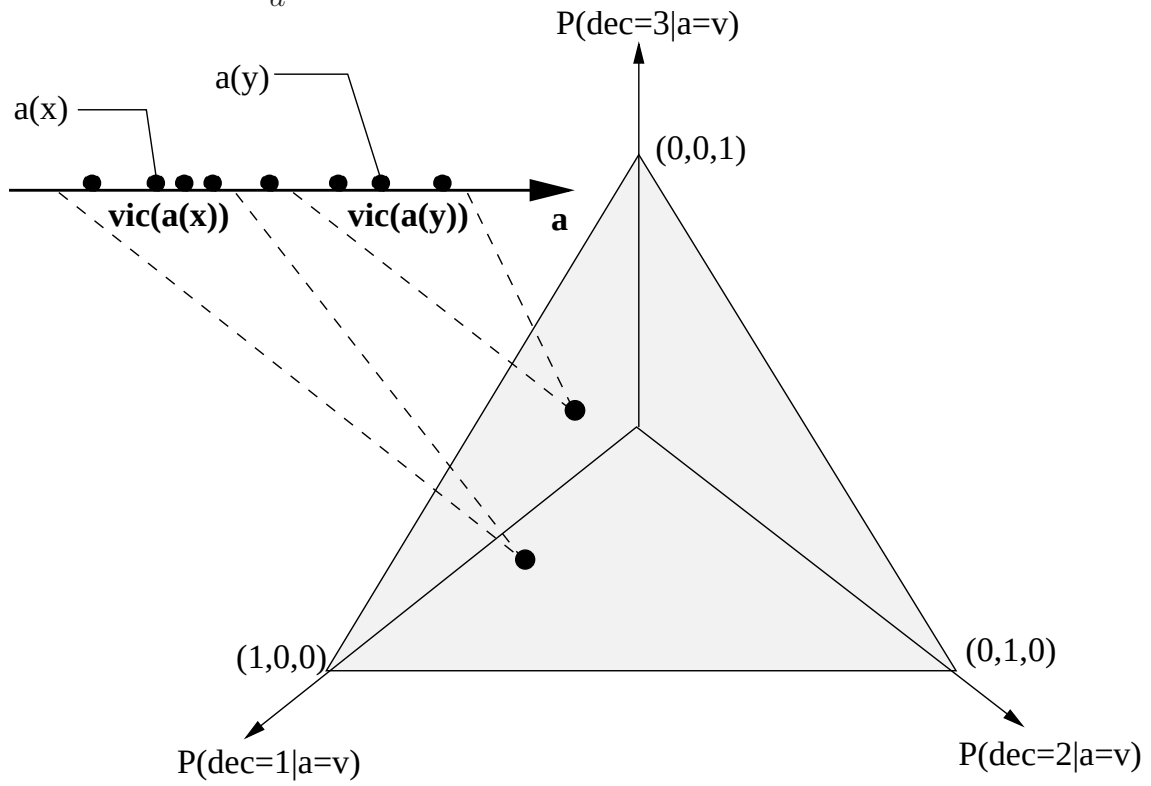
$$\rho_a(a(x), a(y)) = \sum_{d \in V_d} |P(dec = d|a = a(x)) - P(dec = d|a = a(y))|$$



# Miara odleglosci: SVD

Uogólnienie prostej różnicy wartości (SVD) dla atrybutu numerycznego

$$\rho_a(a(x), a(y)) = \sum_{d \in V_d} |P(dec = d | a \in vic(a(x))) - P(dec = d | a \in vic(a(y)))|$$

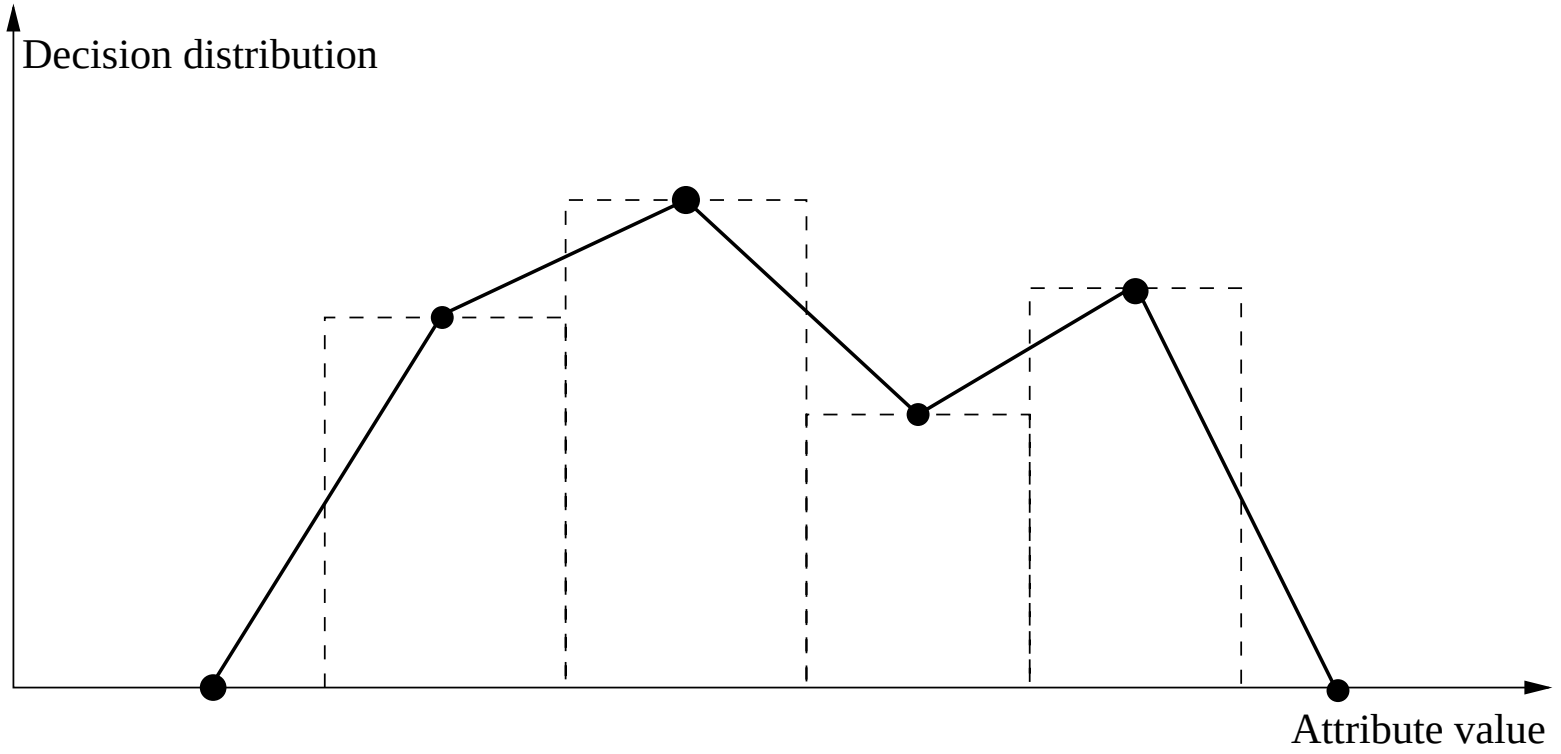



$vic(v)$ — sąsiedztwo wartości $v$, np. przedział $(v - \epsilon; v + \epsilon)$

# Miara odleglosci: IVD (Wilson, Martinez 1997)

Interpolowana różnica wartości (ang. IVD) dla atrybutu numerycznego:

- zakres wartości numerycznych jest dzielony na równe przedziały
- dla każdego przedziału wyliczany jest rozkład decyzji
- dla każdej wartości rozkład decyzji jest interpolowany pomiędzy środkami dwóch nabliższych przedziałów



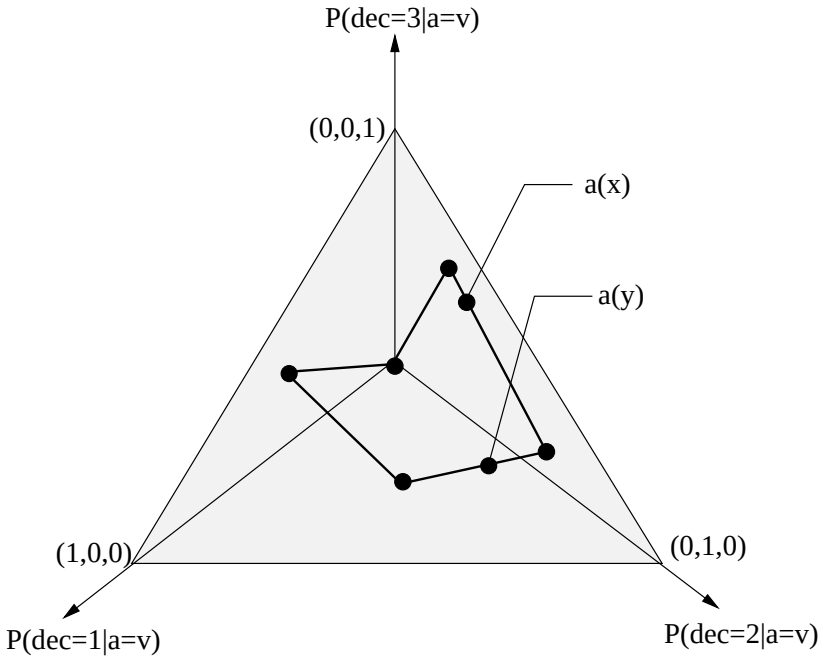
# Miara odleglosci: IVD (Wilson, Martinez 1997)

Interpolowana różnica wartości (ang. IVD) dla atrybutu numerycznego:

— zakres wartości numerycznych jest dzielony na równe przedziały

— dla każdego przedziału wyliczany jest rozkład decyzji

— dla każdej wartości rozkład decyzji jest interpolowany pomiędzy środkami dwóch nabliższych przedziałów




Różnica wartości to różnica pomiędzy interpolowanymi rozkładami decyzji

# Miara odleglosci: IVD (Wilson, Martinez 1997)

Fakt: Odległość IVD jest aproksymacją odległośći SVD

# Miara odleglosci: metryka wazona

Metryka ważona jest ważoną sumą odległości dla poszczególnych atybutów

$$\rho(x, y) = \sum_{a \in A} w_a \rho_a(a(x), a(y))$$

$A$ — zbiór atrybutów
$\rho_a$ — miara odległości dla pojedynczego atrybutu $a \in A$
$w_a$ — waga atrybutu

# Miara odleglosci: metryka wazona

Algorytm ważenia atrybutów na podstawie zbioru treningowego $U_{trn}$

1. $w_a := 1$ dla wszystkich atrybutów $a \in A$
2. $t := 0$
3. Powtarzaj, dopóki $temp(t) > 0$
   a. $t := t + 1$
   b. $x :=$ losowo wybrany obiekt treningowy z $U_{trn}$
   c. $y :=$ najbliższy sąsiad $x$ w $U_{trn}$
   d. Dla każdego atrybutu $a \in A$

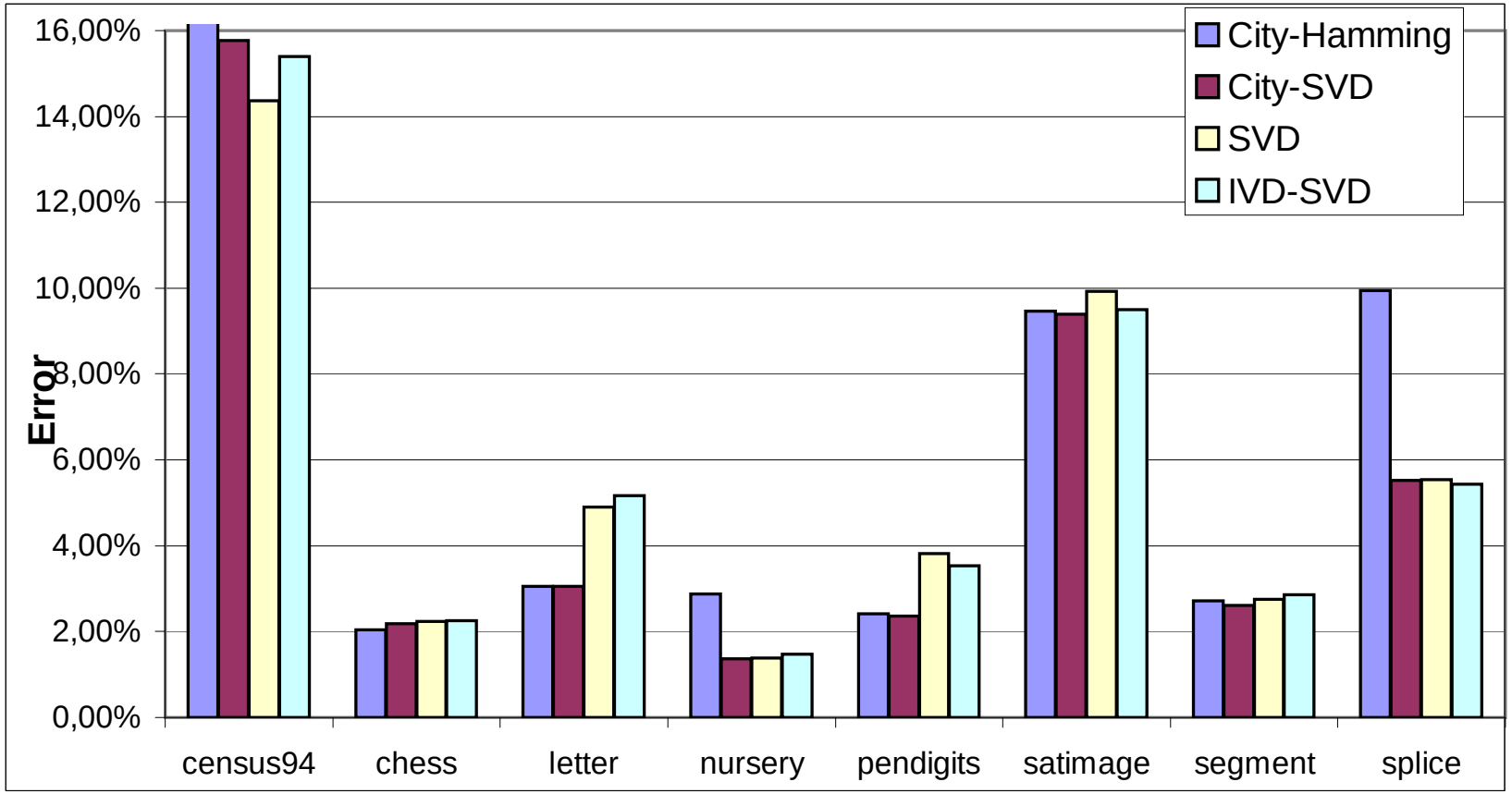
$$\delta w_a := w_a \frac{temp(t)}{\rho_a(a(x), a(y))}$$

$$w_a = \begin{cases} w_a + \delta w_a & \text{jeśli } dec(y) = dec(x) \\ w_a - \delta w_a & \text{wpp.} \end{cases}$$

$temp(t)$ — funkcja "temperatury" malejąca wraz z upływem czasu $t$

# Miara odleglosci: porownanie metryk

Atrybuty numeryczne: letter, pendigits, satimage, segment
Atrybuty symboliczne: chess, nursery, splice
Atrybuty numeryczne + symboliczne: census94



# K najblizszych sasiadow: wybor najlepszgo k

Mając już metrykę można klasyfikować każdy obiekt ze zbioru uczącego używając tego samego zbioru, z wyłączeniem klasyfikowanego obiektu (tzw. test leave-one-out).

Pomysł:

Klasyfikowanie zbioru uczącego dla różnych wartości $k$ i wybór tej, która daje najlepszą skuteczność

# K najblizszych sasiadow: wybor najlepszgo k

Mając już metrykę można klasyfikować każdy obiekt ze zbioru uczącego używając tego samego zbioru, z wyłączeniem klasyfikowanego obiektu (tzw. test leave-one-out).

Pomysł:

Klasyfikowanie zbioru uczącego dla różnych wartości $k$ i wybór tej, która daje najlepszą skuteczność

$\Rightarrow$ Jak to zrobić, żeby zajęło tyle samo czasu
co klasyfikacja dla pojedynczej wartości $k$ ?

# K najblizszych sasiadow: wybor najlepszgo k

**function** FIND-OPTIMAL-K($examples, maxk$) **returns** best $k$

**inputs**: $examples$, a set of training examples

$maxk$, determines the range [1;$maxk$] that is searched for the best $k$

**for** each $x \in examples$ **do**

$A_x \leftarrow$ GET-RESULT-VECTOR($x, examples - \{ x\}, maxk$)

**return** $\arg\max_{1 \leq k \leq maxk} |\{ x \in examples : A_x[k] = dec(x)\}|$

---

**function** GET-RESULT-VECTOR($x, examples, maxk$) **returns** results indexed by $k$

$n_1, \ldots, n_{maxk} \leftarrow$ sequence of $maxk$ nearest neighbors sorted in the increasing order of the distance to $x$

**for** each $d \in V_d$ **do** $votes[d] \leftarrow 0$

$current \leftarrow$ most frequent decision in $examples$

**for** $k$ **from** 1 **to** $maxk$

$votes[dec(n_k)] \leftarrow votes[dec(n_k)] + 1$

**if** $votes[dec(n_k)] > votes[current]$ **then** $current \leftarrow dec(n_k)$

$A_x[k] \leftarrow current$

**return** $A_x$

# K najblizszych sasiadow: wlasnosci

Zalety??

# K najblizszych sasiadow: wlasnosci

Zalety??

— Uczenie jest szybkie

# K najblizszych sasiadow: wlasnosci

Zalety??

— Uczenie jest szybkie

— Uczy się dowolnych funkcji

# K najblizszych sasiadow: wlasnosci

Zalety??

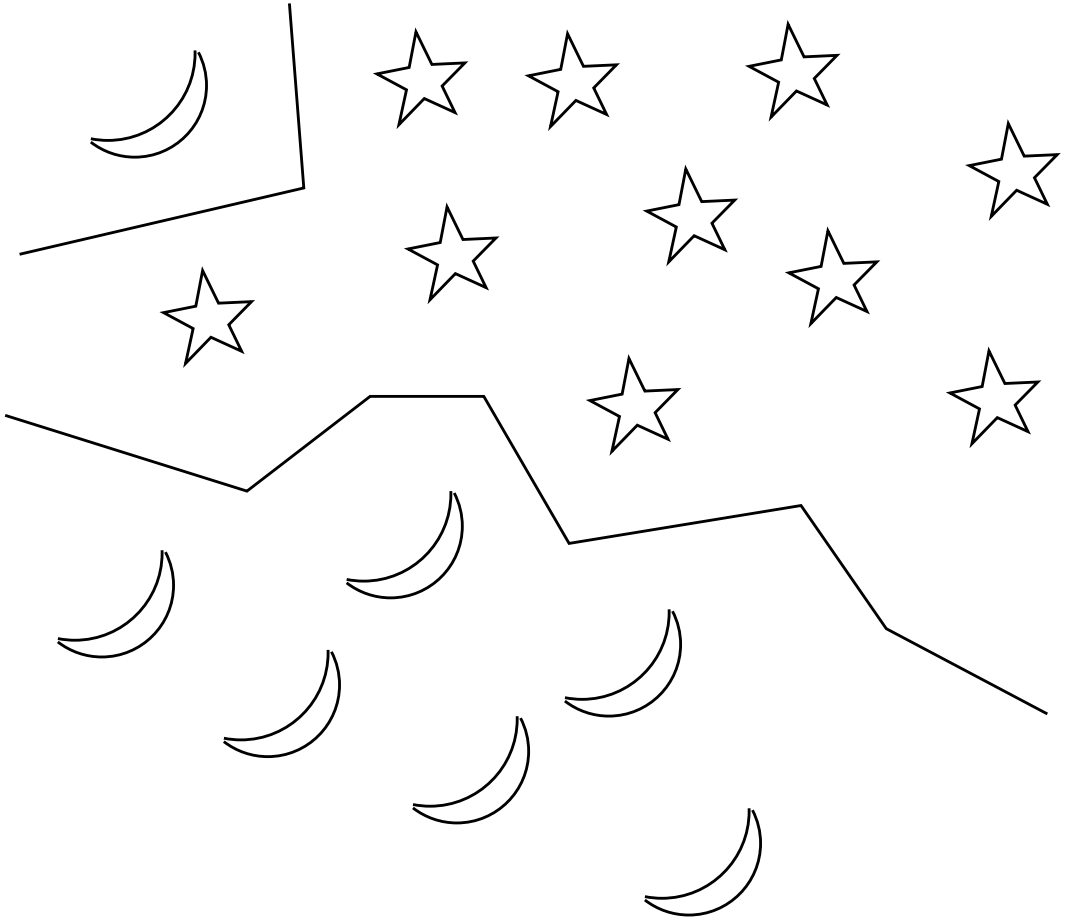
— Uczenie jest szybkie
— Uczy się dowolnych funkcji
— Nie traci informacji

# K najblizszych sasiadow: wlasnosci

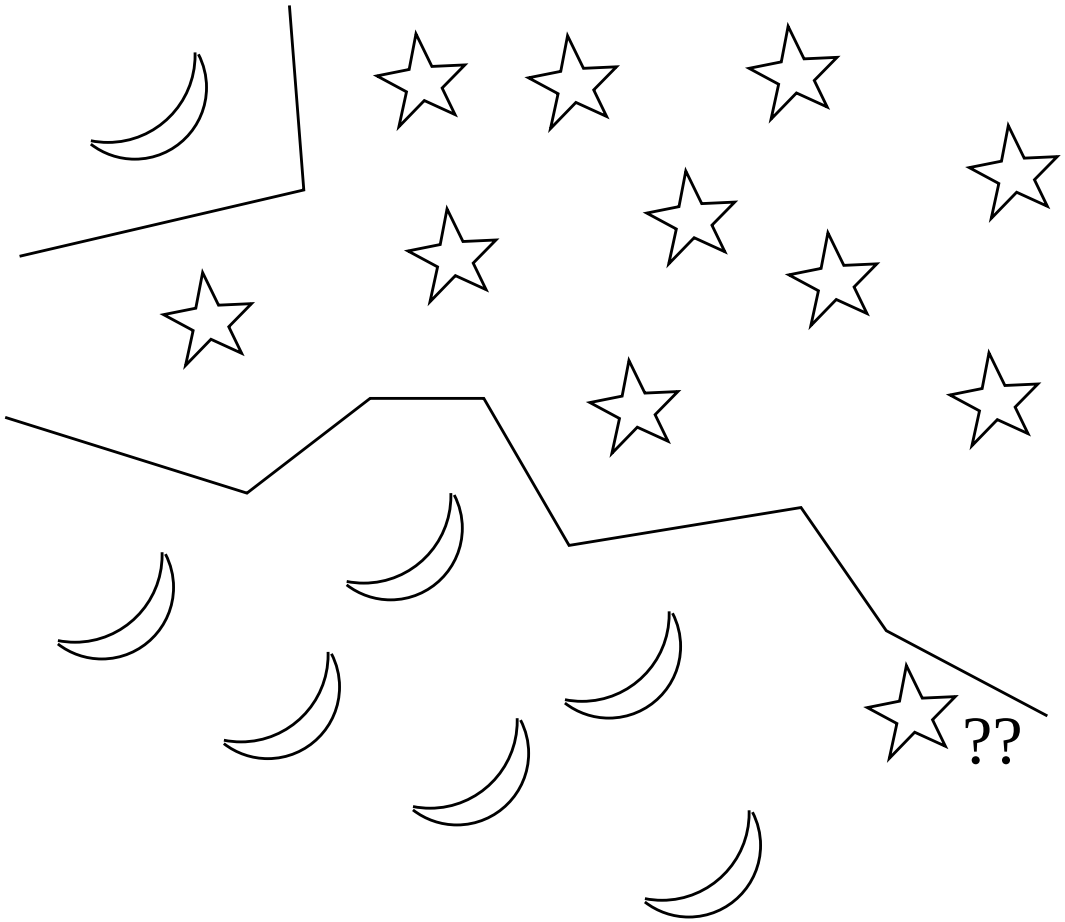
Zalety??

— Uczenie jest szybkie
— Uczy się dowolnych funkcji
— Nie traci informacji
— Naturalnie stosuje się do dynamicznych (rosnących) zbiorów danych

# K najblizszych sasiadow: rosnacy zbior danych



# K najblizszych sasiadow: rosnacy zbior danych




Nowy przykład treningowy dodawany jest do aktualnej bazy przykładów

# K najblizszych sasiadow: rosnacy zbior danych

Nowy przykład treningowy dodawany jest do aktualnej bazy przykładów
$\Longrightarrow$ automatycznie zmienia się funkcja wyznaczana przez bazę przykładów

# K najblizszych sasiadow: wlasnosci

Zalety??

— Uczenie jest szybkie
— Uczy się dowolnych funkcji
— Nie traci informacji
— Naturalnie stosuje się do dynamicznych (rosnących) zbiorów danych

Wady??

# K najblizszych sasiadow: wlasnosci

Zalety??

— Uczenie jest szybkie
— Uczy się dowolnych funkcji
— Nie traci informacji
— Naturalnie stosuje się do dynamicznych (rosnących) zbiorów danych

Wady??

— Niepraktyczny przy dużej liczbie atrybutów
$\Rightarrow$ wymaga redukcji liczby atrybutów

# K najblizszych sasiadow: wlasnosci

Zalety??

— Uczenie jest szybkie
— Uczy się dowolnych funkcji
— Nie traci informacji
— Naturalnie stosuje się do dynamicznych (rosnących) zbiorów danych

Wady??

— Niepraktyczny przy dużej liczbie atrybutów
$\Rightarrow$ wymaga redukcji liczby atrybutów
— Wolny podczas klasyfikacji
$\Rightarrow$ wymaga złożonych struktur indeksujących

# Wyszukiwanie k najblizszych sasiadow

Bez indeksowania:

— Liniowe przeglądanie zbioru przykładów dla każdego zapytania

— Złożoność $O(mn)$

$m$ — liczba zapytań

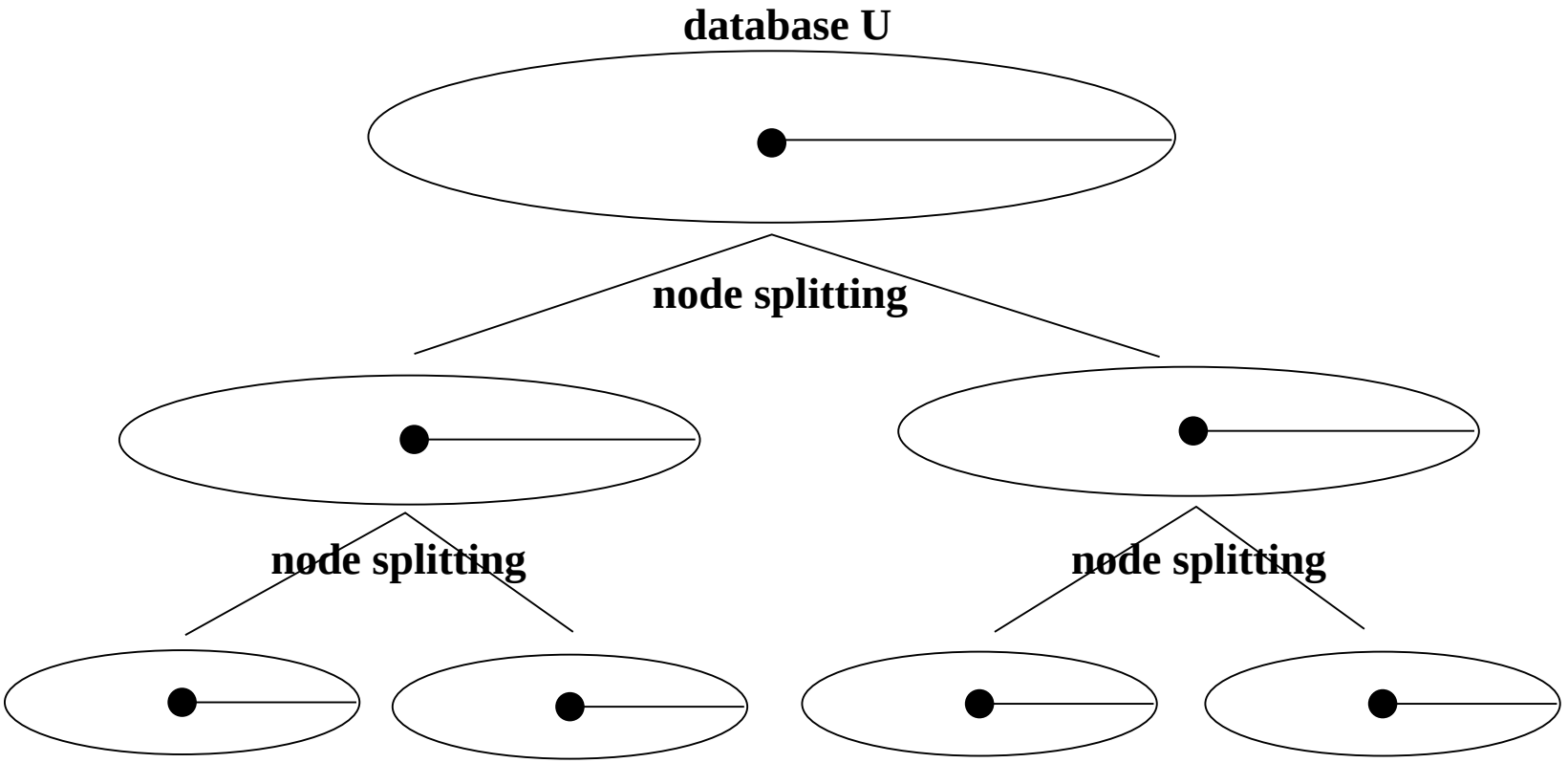
$n$ — rozmiar zbioru przykładów

$\Rightarrow$ Zbyt kosztowne dla dużych zbiorów danych

# Indeksowanie

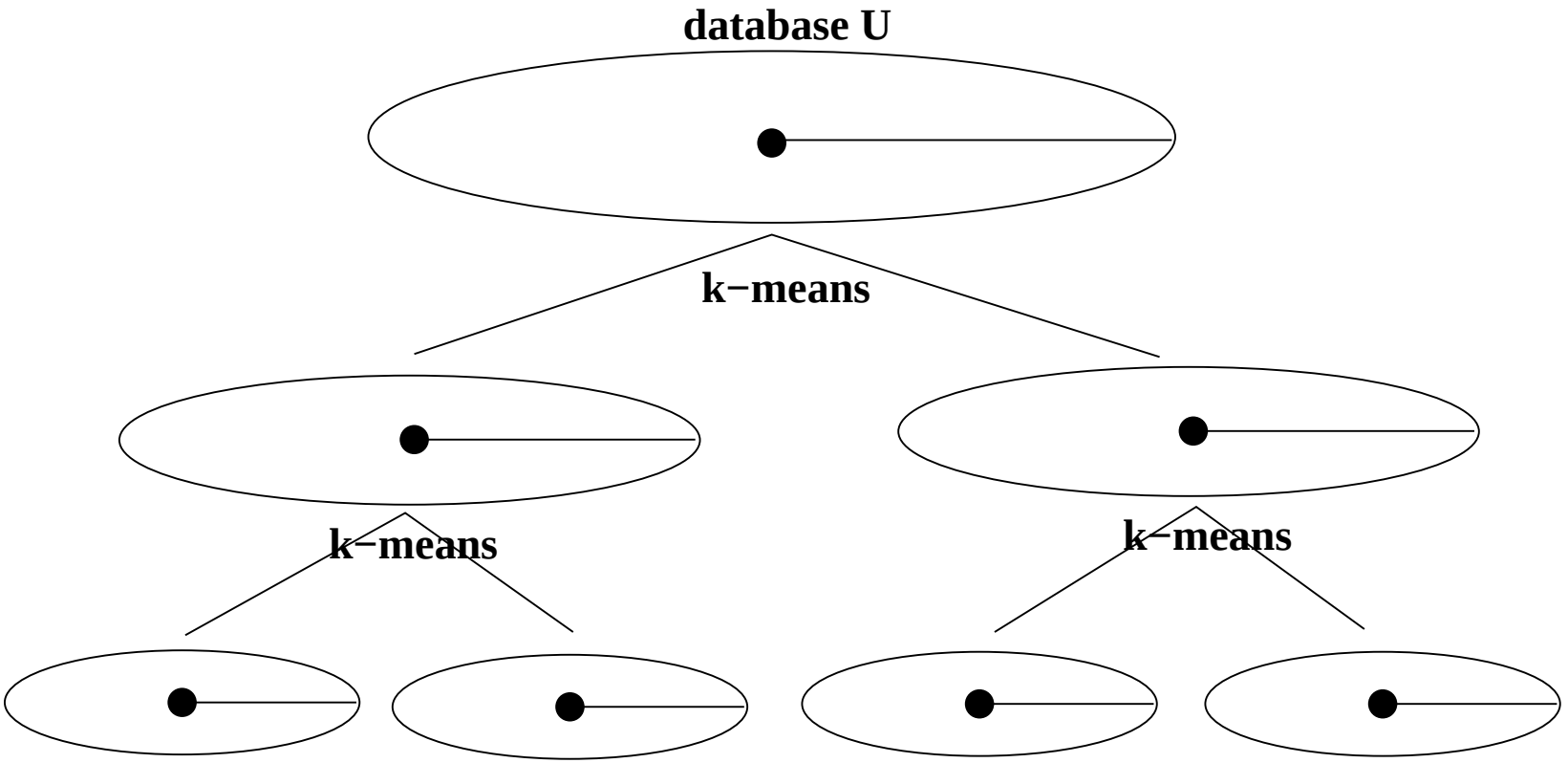
Drzewo indeksujące jest konstruowane metodą top-down:

zaczyna od korzenia zawierającego cały zbiór przykładów treningowych i rekurencyjnie dzieli klastry na coraz mniejsze



# Podzial klastra metoda k-srodkow

Do podziału klastrów dobra jest metoda $k$-środków (ang. $k$-means)



# Podzial klastra metoda k-srodkow

Algorytm wyboru $k$-środków:

**function** KMEANS($examples, k$) **returns** a set of clusters

  **for each** $0 \leq i < k$ **do** $center_i \leftarrow$ a random example from $examples$
  **repeat**
    **for each** $0 \leq i < k$ **do** $cluster_i \leftarrow \{\}$
    **for each** example $x \in examples$ **do**
      $nearest \leftarrow \min \arg_{0 \leq i < k}$DISTANCE($center_i, x$)
      $cluster_{nearest} \leftarrow cluster_{nearest} \cup \{x\}$
    **for each** $0 \leq i < k$ **do** $center_i \leftarrow$ the mean of $cluster_i$
  **until** no center has changed (comparing with the last but one iteration)
  **return** the set of $cluster_i$

# Podzial klastra metoda k-srodkow: wlasnosci

Uniwersalność??

# Podzial klastra metoda k-srodkow: wlasnosci

Uniwersalność??

Indeksowanie z podziałem klastrów metodą $k$-środków używa tylko pojęć

— metryki

— środka zbioru obiektów

# Podzial klastra metoda k-srodkow: wlasnosci

Uniwersalność??

Indeksowanie z podziałem klastrów metodą $k$-środków używa tylko pojęć

— metryki

— środka zbioru obiektów

$\Rightarrow$ wyszukiwanie jest poprawne przy dowolnej definicji środka klastra,

wybór środka ma istotne znaczenie dla efektywności wyszukiwania,

ale nie ma wpływu na poprawność

# Podzial klastra metoda k-srodkow: wlasnosci

Uniwersalność??

Indeksowanie z podziałem klastrów metodą $k$-środków używa tylko pojęć
— metryki
— środka zbioru obiektów

$\Rightarrow$ wyszukiwanie jest poprawne przy dowolnej definicji środka klastra,
wybór środka ma istotne znaczenie dla efektywności wyszukiwania,
ale nie ma wpływu na poprawność

$\Rightarrow$ można stosować nie tylko do przestrzeni $\mathbf{R^n}$

# Podzial klastra metoda k-srodkow: wlasnosci

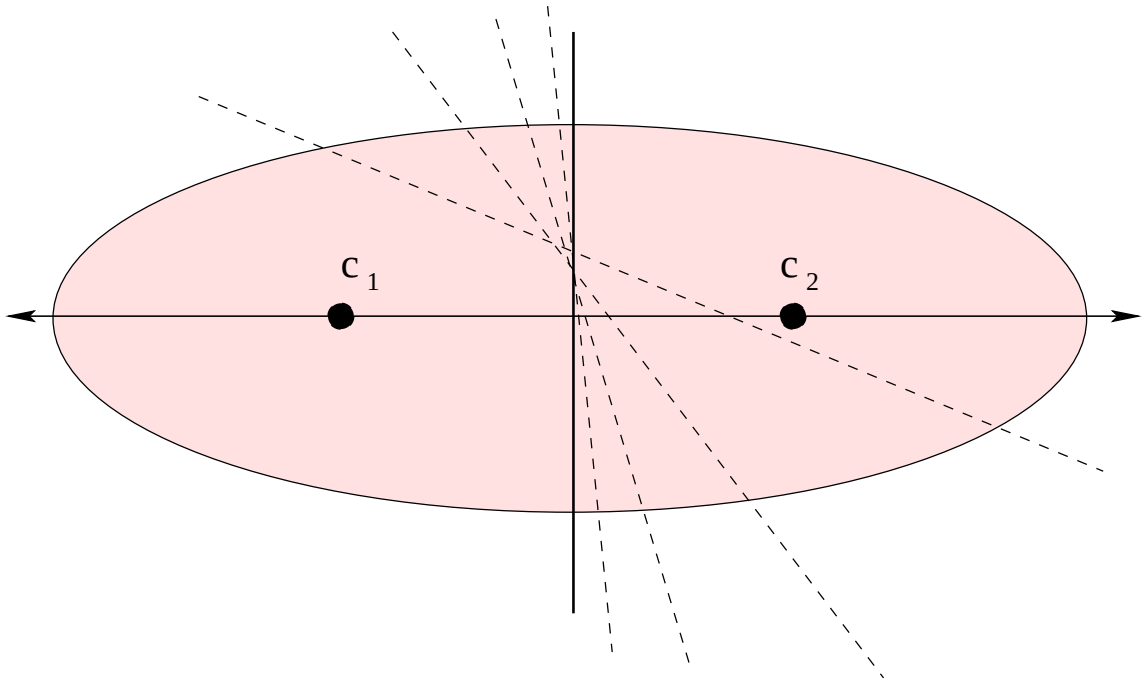
Optymalność??

# Podzial klastra metoda k-srodkow: wlasnosci

Optymalność??

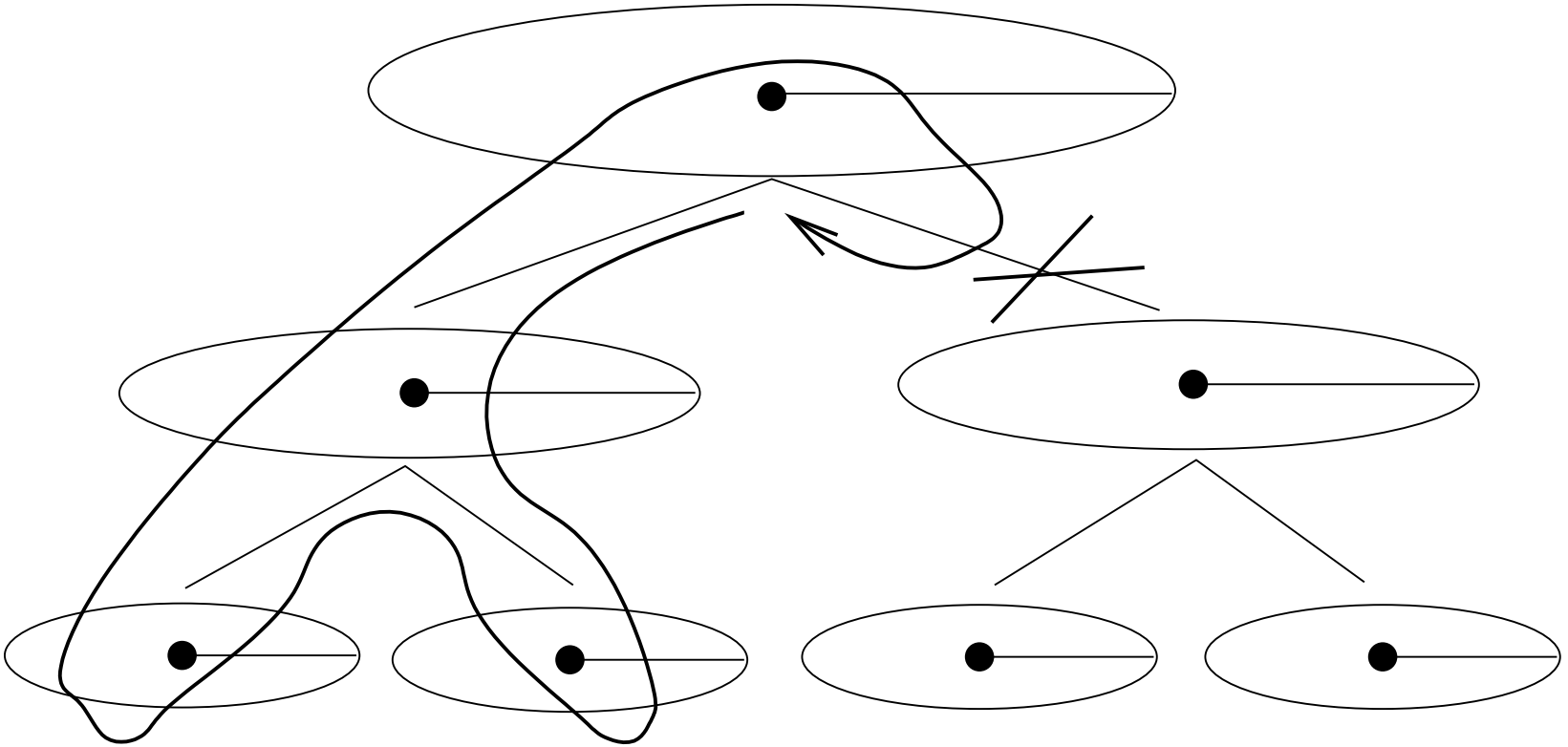
Twierdzenie (Savaresi, Boley, 2001)

Dla eliptycznego modelu zbioru danych metoda 2-środków zbiega do podziału ortogonalnego względem kierunku głównego (tzn. takiego, wzdłuż którego wariancja danych jest największa)



# Wyszukiwanie najblizszych sasiadow

Przeszukuje drzewo indeksujące stosując kryteria odcięcia



# Wyszukiwanie najblizszych sasiadow: algorytm

*nearest* - kolejka priorytetowa dotychczas znalezionych najbliższych sąsiadów

**function** KNN-SEARCH(*node*, *query*, *nearest*) **returns** an updated *nearest*

**if** *node* is a leaf
**then for each** $x \in node$
**if** *nearest* is not full **then** $nearest \leftarrow nearest \cup \{x\}$
**else**
$y \leftarrow \max \arg_{z \in nearest}$ DISTANCE$(query, z)$
**if** DISTANCE$(query, x)$ < DISTANCE$(query, y)$
**then** $nearest \leftarrow$ replace $y$ with $x$
**elseif** *node* has child nodes
**then for each** $child \in$ child nodes of *node*
$radius \leftarrow \max_{z \in nearest}$ DISTANCE$(query, z)$
**if** *nearest* is not full **or** $\neg$DISCARD$(child, query, radius)$
**then** $nearest \leftarrow$ KNN-SEARCH$(child, query, nearest)$
**return** *nearest*

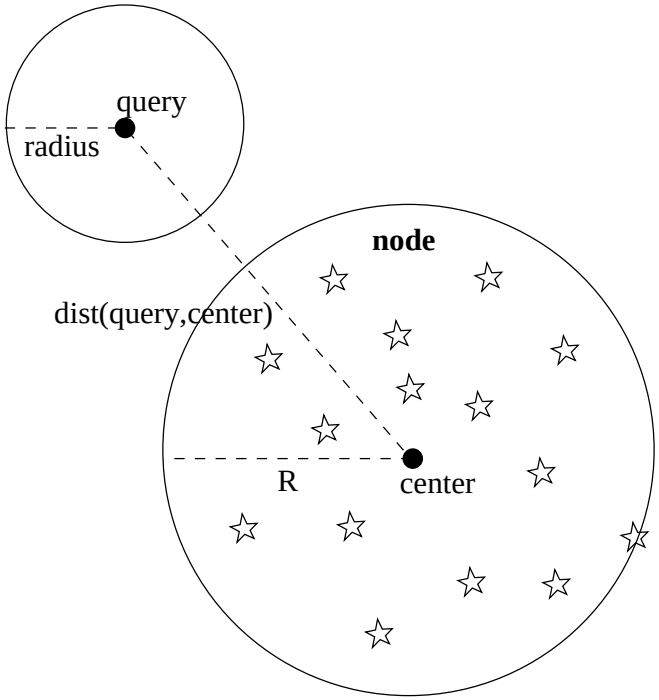
Funkcja DISCARD sprawdza kryteria odcięcia węzła przy przeszukiwaniu

# Kryteria odciecia wezla

- ♢ Cięcie kuliste
- ♢ Cięcie symetralne
- ♢ Cięcie pierścieniowe

# Kryteria odciecia wezla: ciecie kuliste

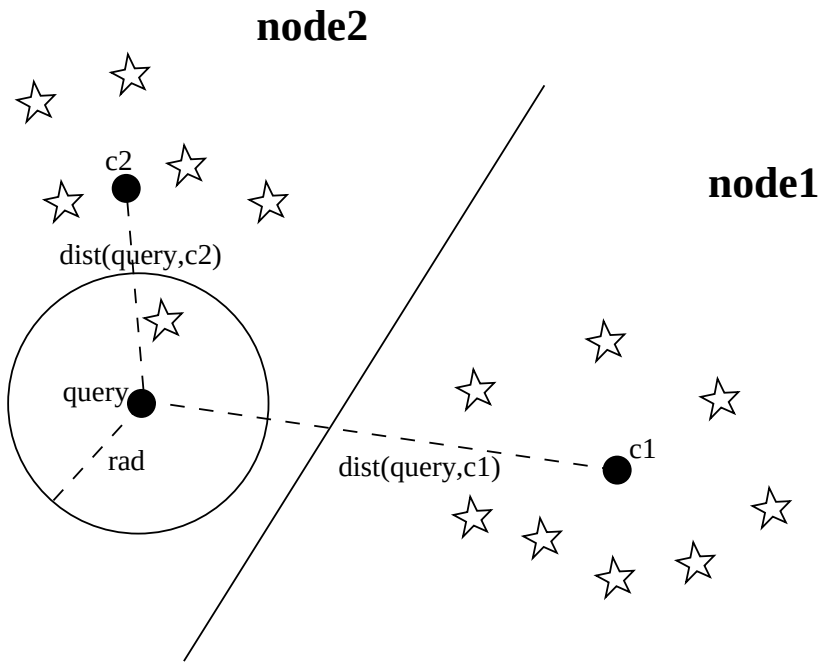
$query$ — zapytanie (testowany obiekt)
$radius$ — promień zapytania
$center$ — środek węzła (klastra)
$R$ — promień pokrywający węzeł: $R = \max_{x \in node} dist(center, x)$




$node$ jest pomijany $\Longleftrightarrow dist(query, center) > radius + R$

# Kryteria odciecia wezla: ciecie symetralne

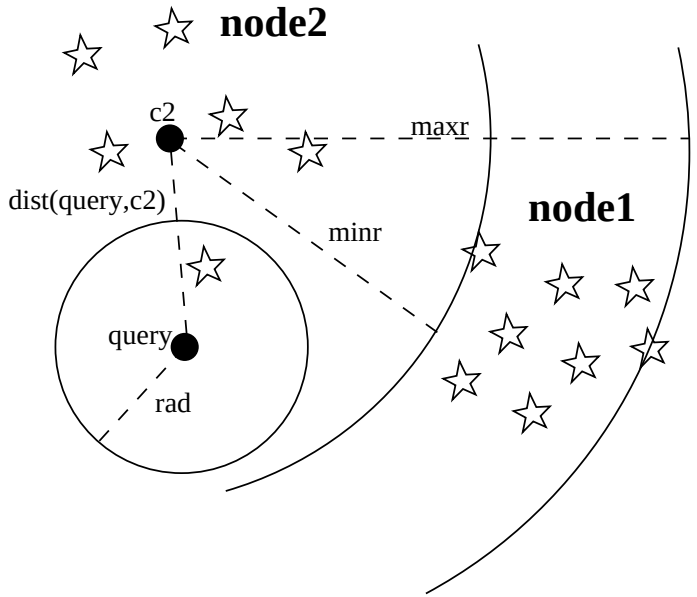
$query$ — zapytanie (testowany obiekt)
$rad$ — promień zapytania
$node1$,$node2$ — węzły będące dziećmi tego samego rodzica w drzewie indeks.
$c1$, $c2$ — środki węzłów




$node1$ jest pomijany $\Longleftrightarrow dist(query, c1) - rad > dist(query, c2) + rad$

# Kryteria odciecia wezla: ciecie pierscieniowe

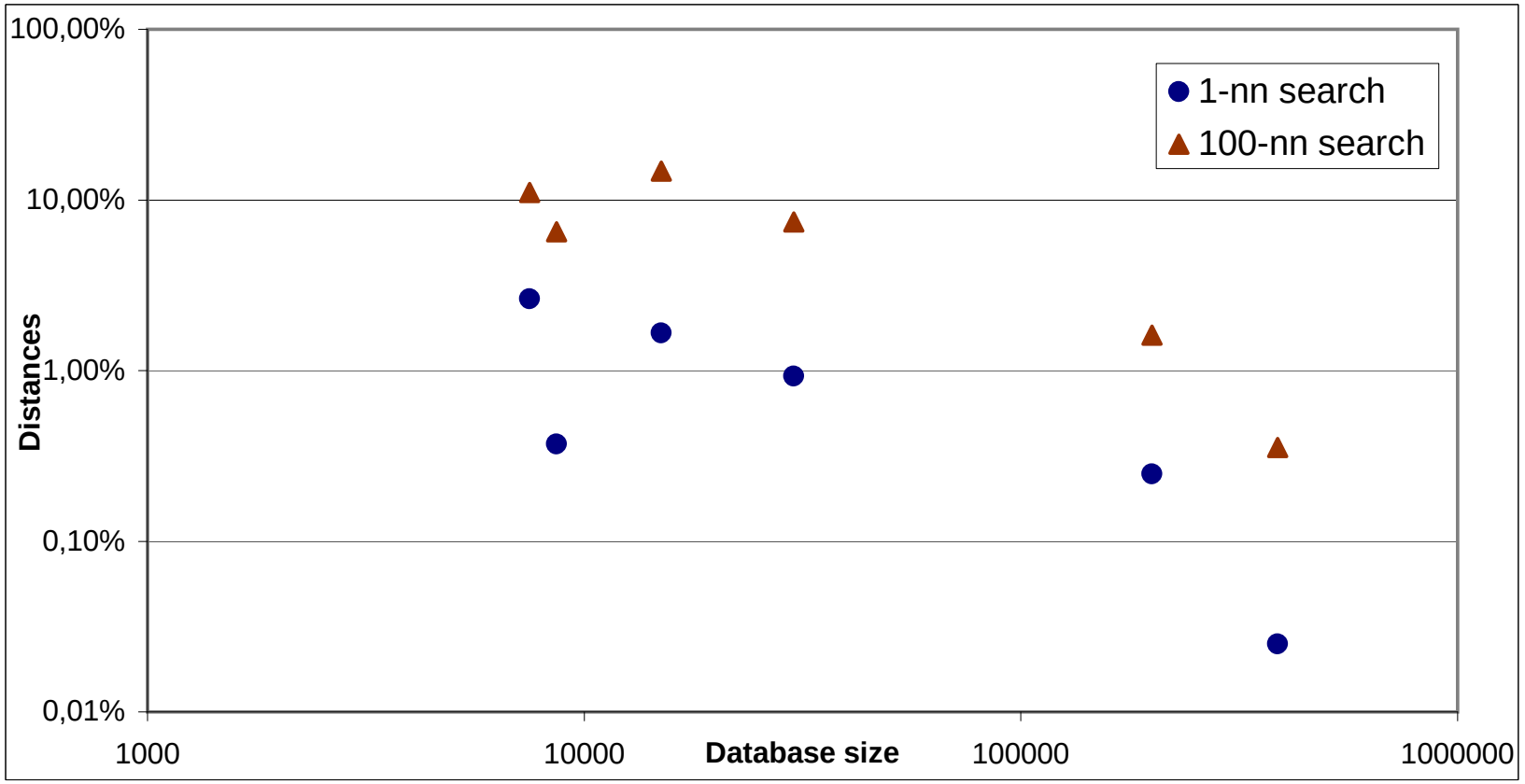
$query$ — zapytanie (testowany obiekt)
$rad$ — promień zapytania
$node1$,$node2$ — węzły będące dziećmi tego samego rodzica w drzewie indeks.
$c2$ — środek węzła $node2$
$minr = \min_{x \in node1} dist(c2, x)$, $maxr = \max_{x \in node1} dist(c2, x)$




$node1$ jest pomijany
$\iff dist(query, c2) + rad < minr$ lub $dist(query, c2) - rad > maxr$

# Wyszukiwanie najblizszych sasiadow

Przyspiszenie wyszukiwania dla różnych wielkości bazy przykładów:



# Metody k-nn z lokalna adaptacja metryki

Atrybuty numeryczne:

- ♢ lokalna adaptacja wag atrybutów
- ♢ głosowanie przy użyciu macierzy kowariancji

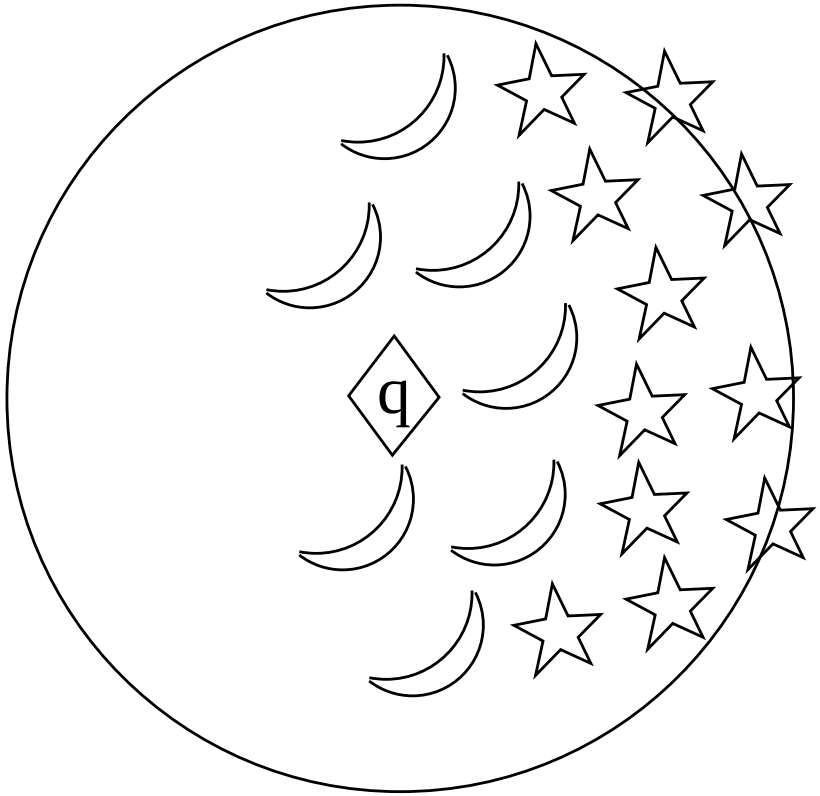
Atrybuty symboliczne:

- ♢ indukcja lokalnej metryki z sąsiedztwa obiektu

# Lokalna adaptacja wag atrybutow

Hastie, Tibshirani, 1996

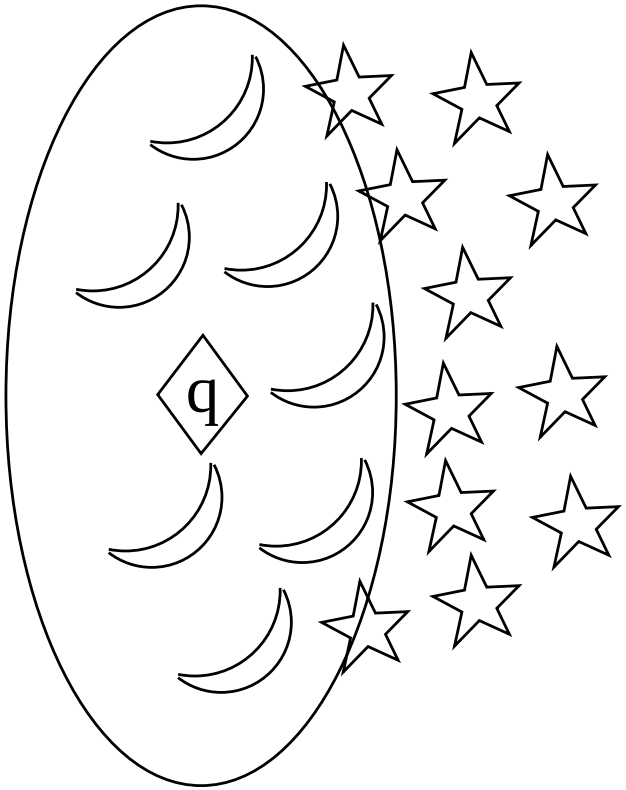
Atrybuty są lokalnie ważone na podstawie analizy najbliższego sąsiedztwa obiektu testowanego:



# Lokalna adaptacja wag atrybutow

Hastie, Tibshirani, 1996

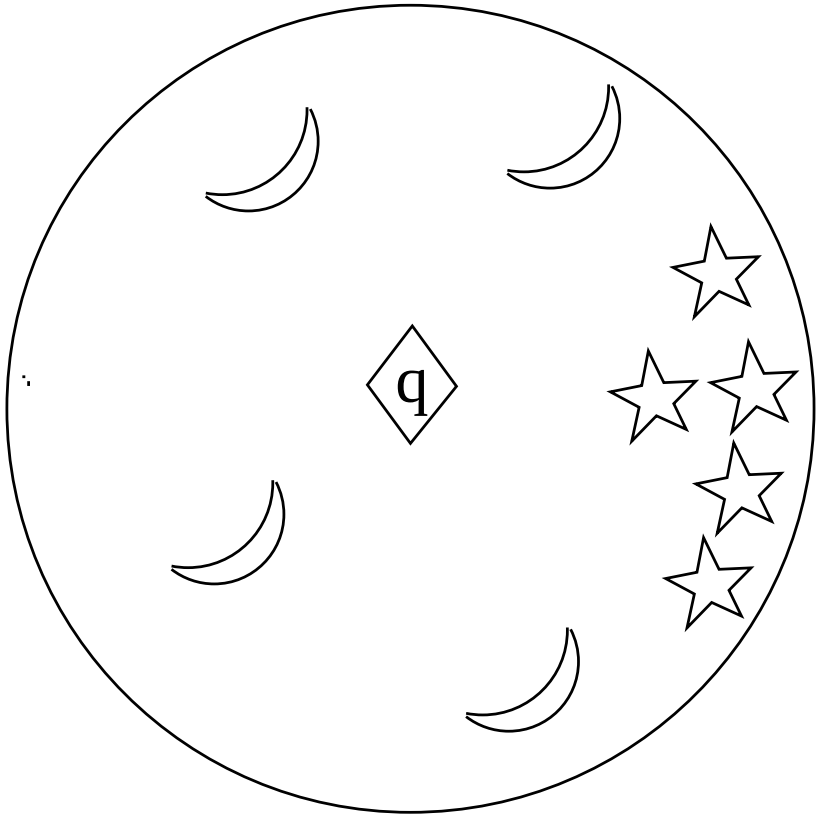
Atrybuty są lokalnie ważone na podstawie analizy najbliższego sąsiedztwa obiektu testowanego:



# Glosowanie przy uzyciu macierzy kowariancji

Problem:

Niektóre obiekty mogą dublować informacje w innych obiektach,
w związku z tym nie powinny być brane pod uwagę przy głosowaniu



# Glosowanie przy uzyciu macierzy kowariancji

Rozwiązanie: Głosowanie przy użyciu macierzy kowariancji

Dla każdej klasy decyzyjnej $d$ w zbiorze treningowym $U_{trn}$ tworzona jest macierz kowariancji między parami obiektów tej klasy $x_i, x_j \in U_{trn}$:

$$C_d = [C_{i,j}] \qquad C_{i,j} = \gamma(\rho(x_i, x_j)) \qquad \gamma \text{ — funkcja monotoniczna}$$

Klasyfikacja obiektu testowego $x_q$

1. Dla każdej klasy decyzyjnej $d$ tworzony jest wektor kowariancji z obiektami należącymi do tej klasy decyzyjnej:

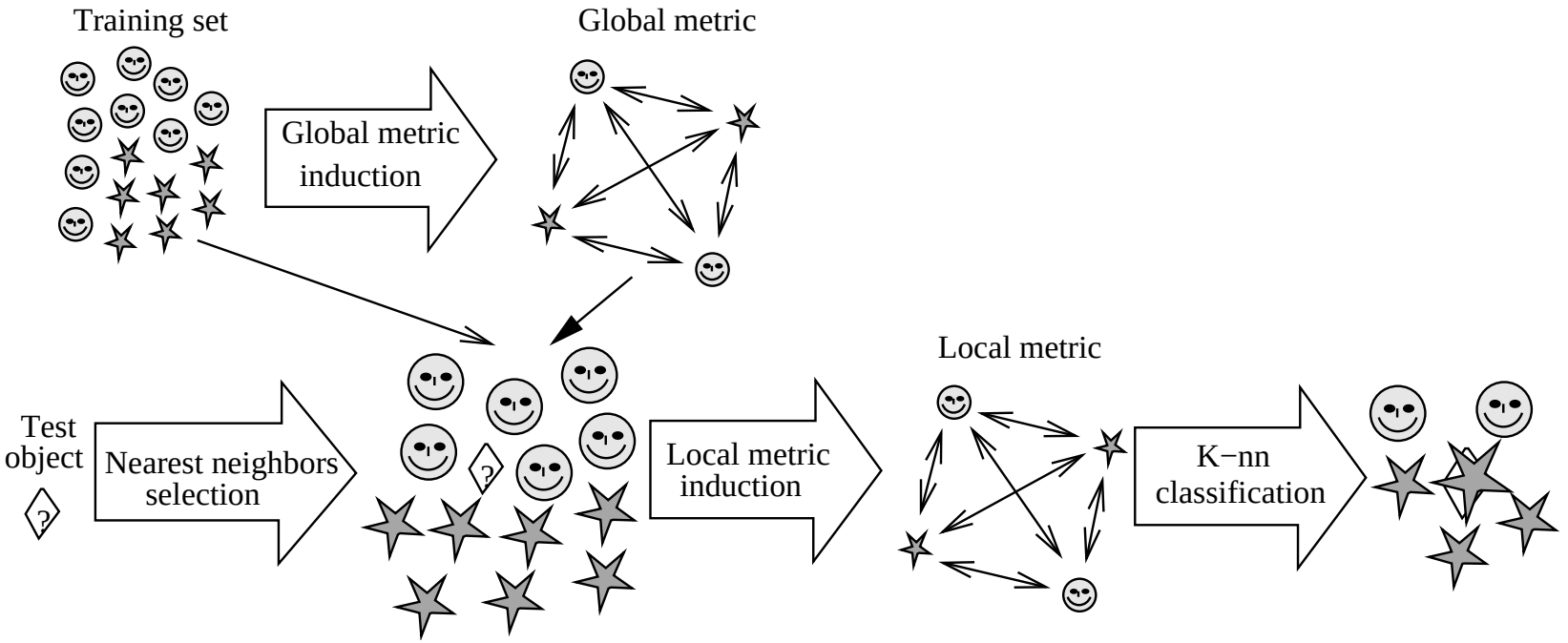
   $$c_d(x_q) = [c_j] \qquad c_j = \gamma(\rho(x_q, x_j))$$

2. Głosowanie odbywa się na podstawie macierzy i wektora kowariancji:

   $$\alpha_d = (\alpha_{d,1}, \ldots, \alpha_{d,n}) = C_d^{-1} \circ c_d(x_q)$$

   Decyzja dla $x_q = \max \arg_{d \in V_d} \sum_{i=1}^{n} \alpha_{d,i}$

# Indukcja lokalnej metryki z sasiedztwa obiektu



# Indukcja lokalnej metryki z sasiedztwa obiektu

Porównanie skuteczności klasyfikacji:

atrybuty: symboliczne
metryki globalna i lokalna: SVD
$n$ — rozmiar sąsiedztwa branego do indukcji metryki lokalnej