KMI/ZZD – Získávání znalostí z dat

Shlukovnání

Jan Konečný

7. března 2012

Shlukovnání

Jedno ze dvou technik DM učení bez učitele:

Cíl: Najít rozklad množiny datapointů tak, aby jednotlivé prvky rozkladu (shluky) odpovídaly geometrii dat.

celkový počet rozkladů určuje tzv. Stirlingovo číslo:

$$\frac{1}{c!}\sum_{i=1}^{c}(-1)^{c-i}\binom{c}{i}i^{N}$$

Pro ilustraci $N = 100$, $c = 5 \ldots 10^{67}$ rozkladů.
Zjevně potřebujeme optimalizace známé jako *metody shlukování*

Kategorie shlukovacích metod

Shlukovací metody mohou být rozděleny do kategorií:

shlukování založené na rozkladu spoléháme se na cílovou funkci, jejíž minimalizace má vést k objevu struktury datasetu.

hierarchické shlukování postupný vývoj shluků; začíná se jedním shlukem (přes celý dataset) a ten se rozděluje, anebo jednotlivými datapointy (každý je jeden shluk) a ty se slévají.

sekvenční shlukování jednoduché algoritmy s nízkou složitostí (obvykle $\mathcal{O}(n)$).

zbytek samoorganizační mapy, genetické shlukování, shlukovaní založené na modelu, na gridu, na hustotě, valey-seeking, ...

Hierarchické shlukování

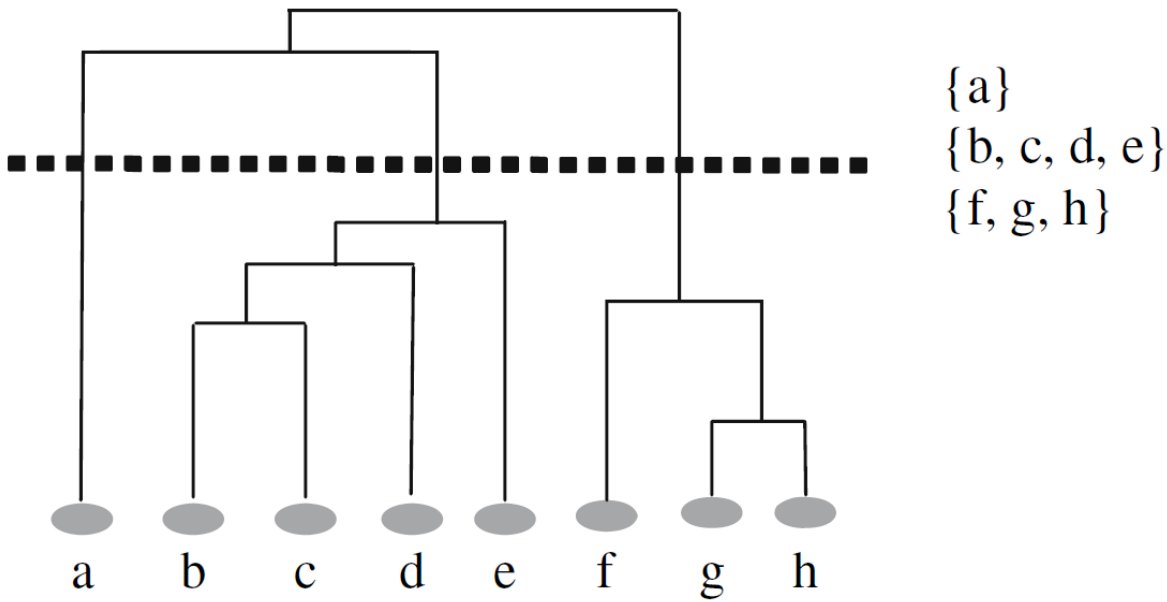
Hierarchické sklukování produkují grafickou reprezentaci dat.
Konstrukce grafu je dělána dvěma způsoby:

- *zdola-nahoru (bottom-up)* – neboli aglomerativní přístup, uvažujeme každý datapoint jako shluk o jednom prvku. Potom postupně sléváme nejbližší shluky. V každém průchodu algoritmu slejeme dva shluky, které jsou si nejblíže. Proces pokračuje, dokud se nedostaneme jeden shluk, nebo dokud nedosáhneme předem určené hranice.
- *shora dolu (top-down)* – neboli dělicí přístup, uvažujeme celý dataset jako jeden shluk, který dělíme na menší shluky.

Kvůli povaze těchto procesů zdola-nahoru a shora-dolu, jsou tyto metody často výpočetně neefektivní (možná až na binární data)

Výsledky hierarchického shlukování jsou reprezentovány v podobě tzv. *dendogramu*.

Dendogram je definován jako binární zakořeněný strom, který má všechny datapointy v listech.



Máme několik způsobů výpočtu vzdálenosti mezi dvěma shluky A, B:

Single link method vzálenost $\mathrm{d}(A, B)$ je minimální vzdálenost mezi dvěma datapointy náležejících do A a B

$$\mathrm{d}(A, B) = \min_{\vec{x} \in A, \vec{y} \in B} \mathrm{d}(x, y).$$

Complete link method vzálenost $\mathrm{d}(A, B)$ je maximální vzdálenost mezi dvěma datapointy náležejících do A a B

$$\mathrm{d}(A, B) = \max_{\vec{x} \in A, \vec{y} \in B} \mathrm{d}(x, y).$$

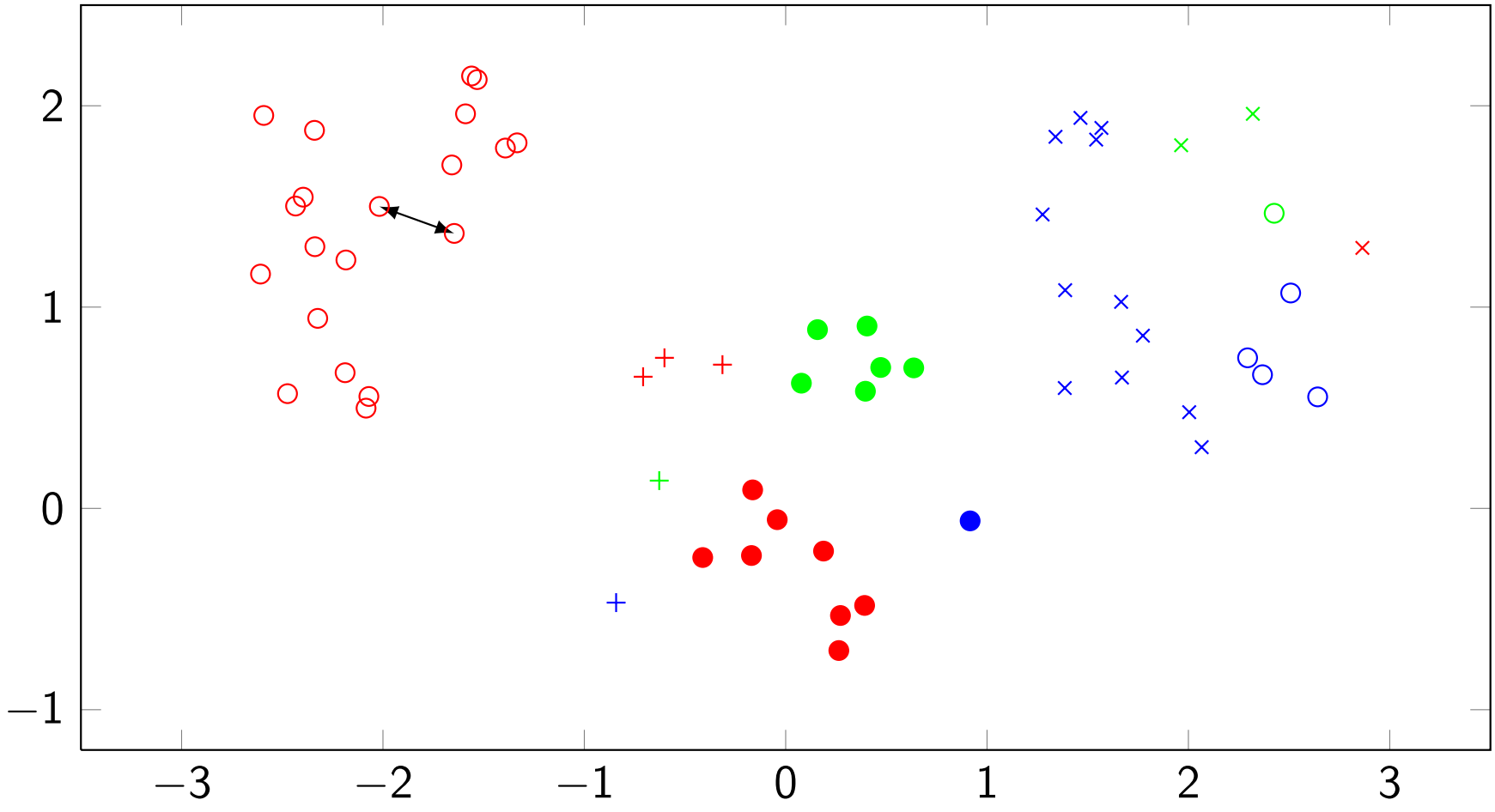
Group average link vzálenost $\mathrm{d}(A, B)$ je průměrná vzdálenost mezi dvěma datapointy náležejících do A a B

$$\mathrm{d}(A, B) = \frac{1}{|A||B|} \sum_{\vec{x} \in A, \vec{y} \in B} \mathrm{d}(x, y).$$

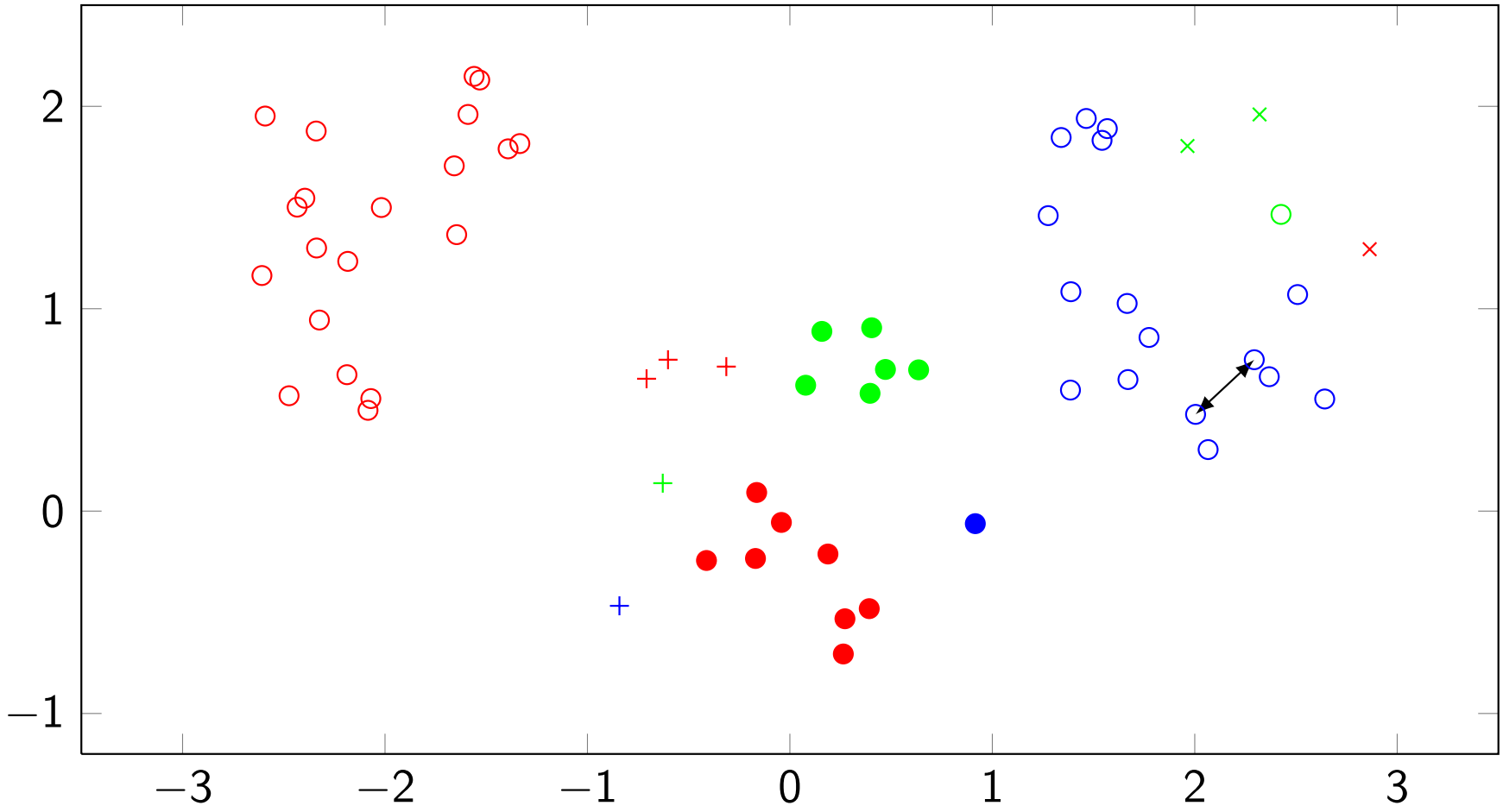
Samozřejmě se dají vymyslet i jiné:
Hausdorffova vzdálenost

$$\mathrm{d}(A, B) = \max\{\max_{\vec{x} \in A} \min_{\vec{y} \in B} \mathrm{d}(\vec{x}, \vec{y}), \max_{\vec{y} \in B} \min_{\vec{x} \in A} \mathrm{d}(\vec{x}, \vec{y})\}$$

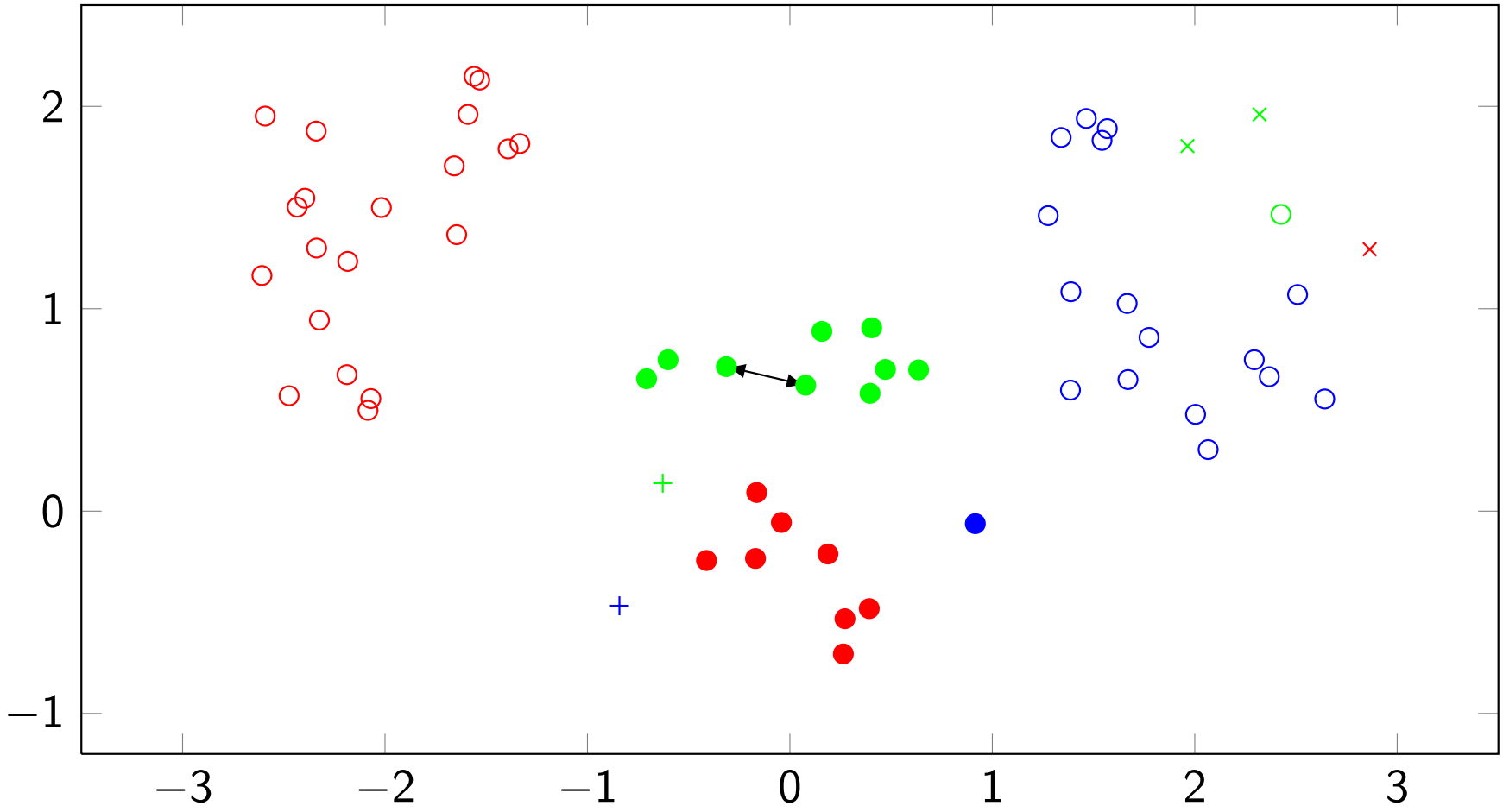
Demonstrace hierarchického shlukování (single link)



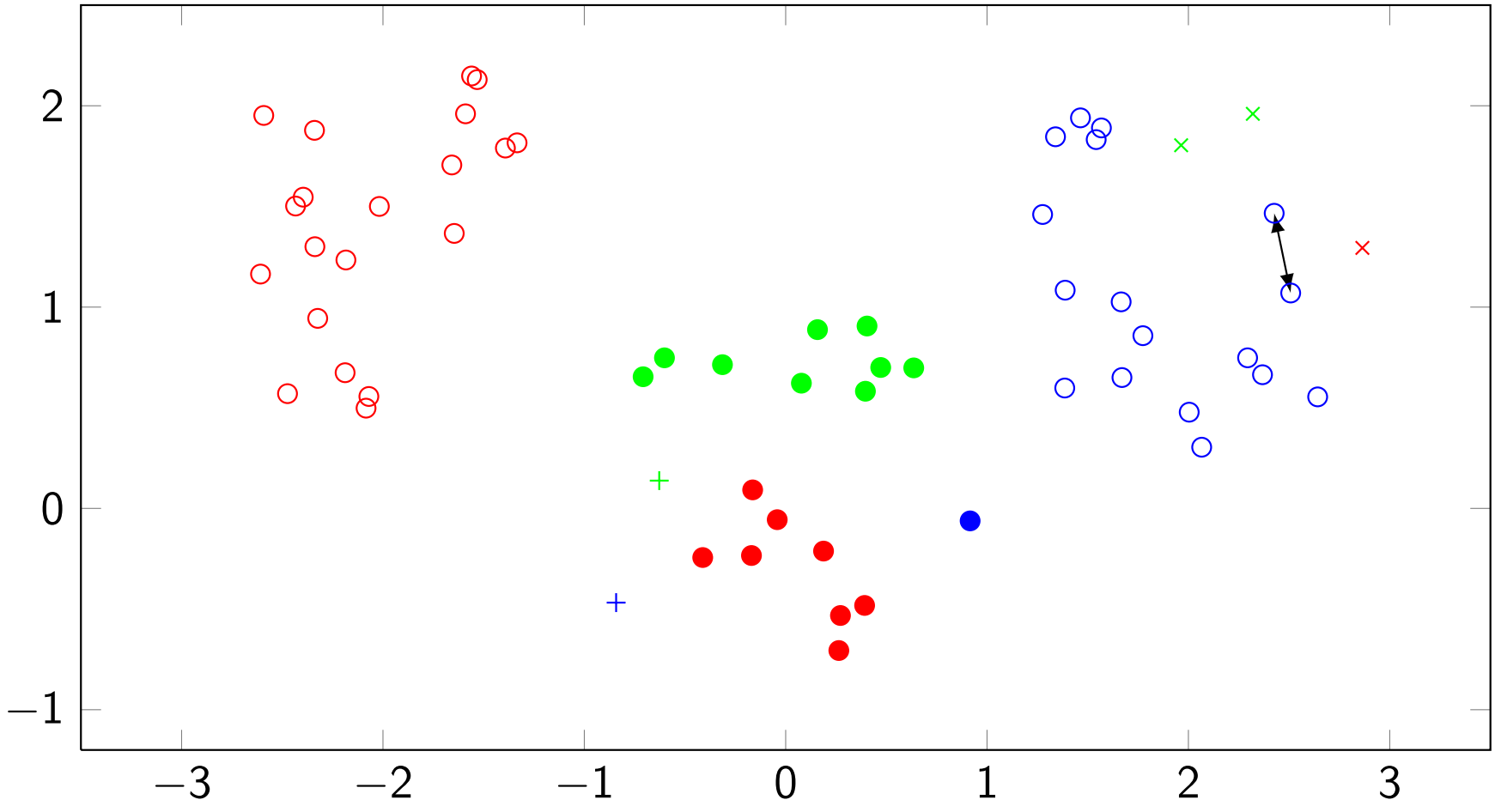
Demonstrace hierarchického shlukování (single link)



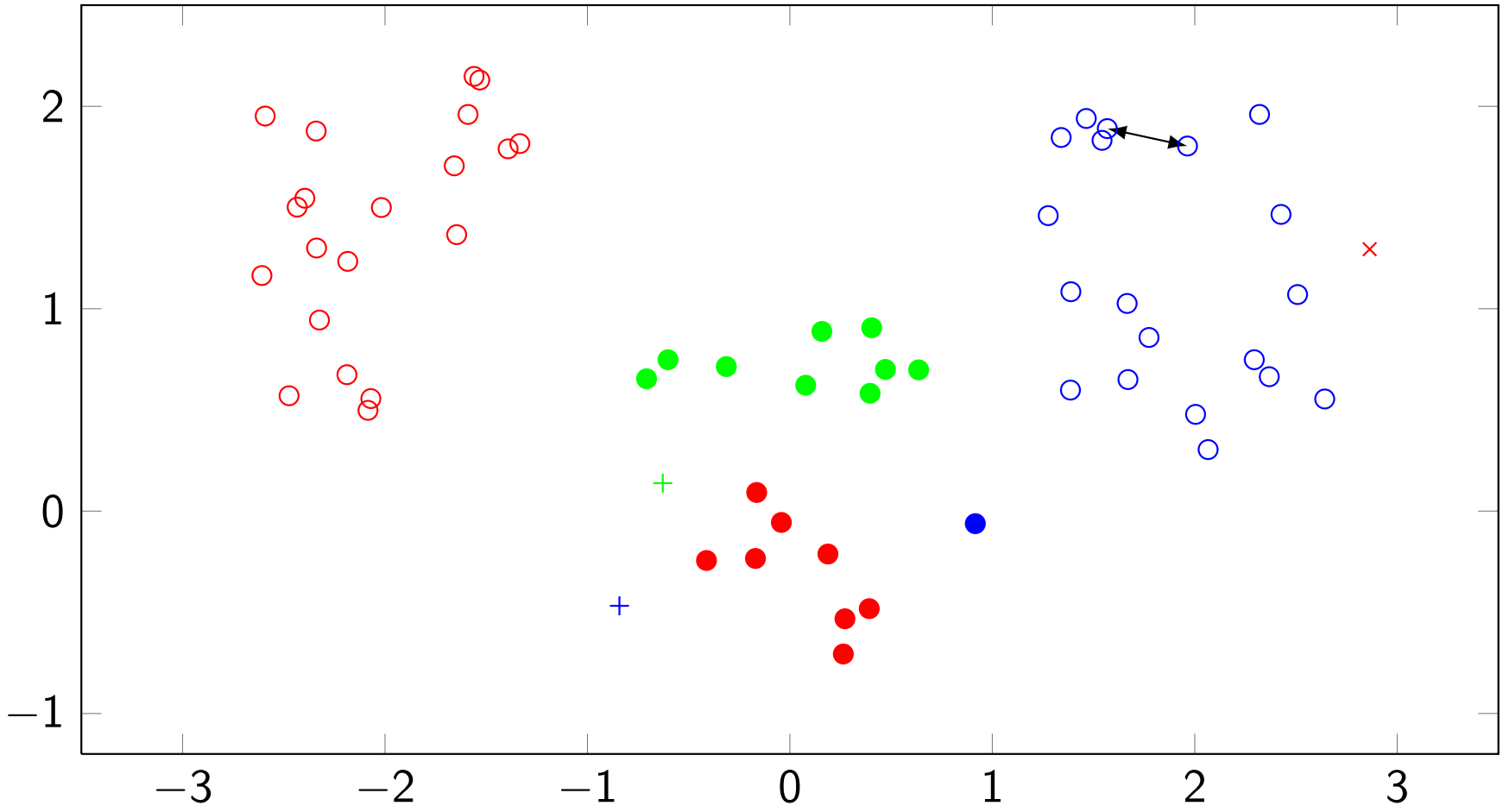
Demonstrace hierarchického shlukování (single link)



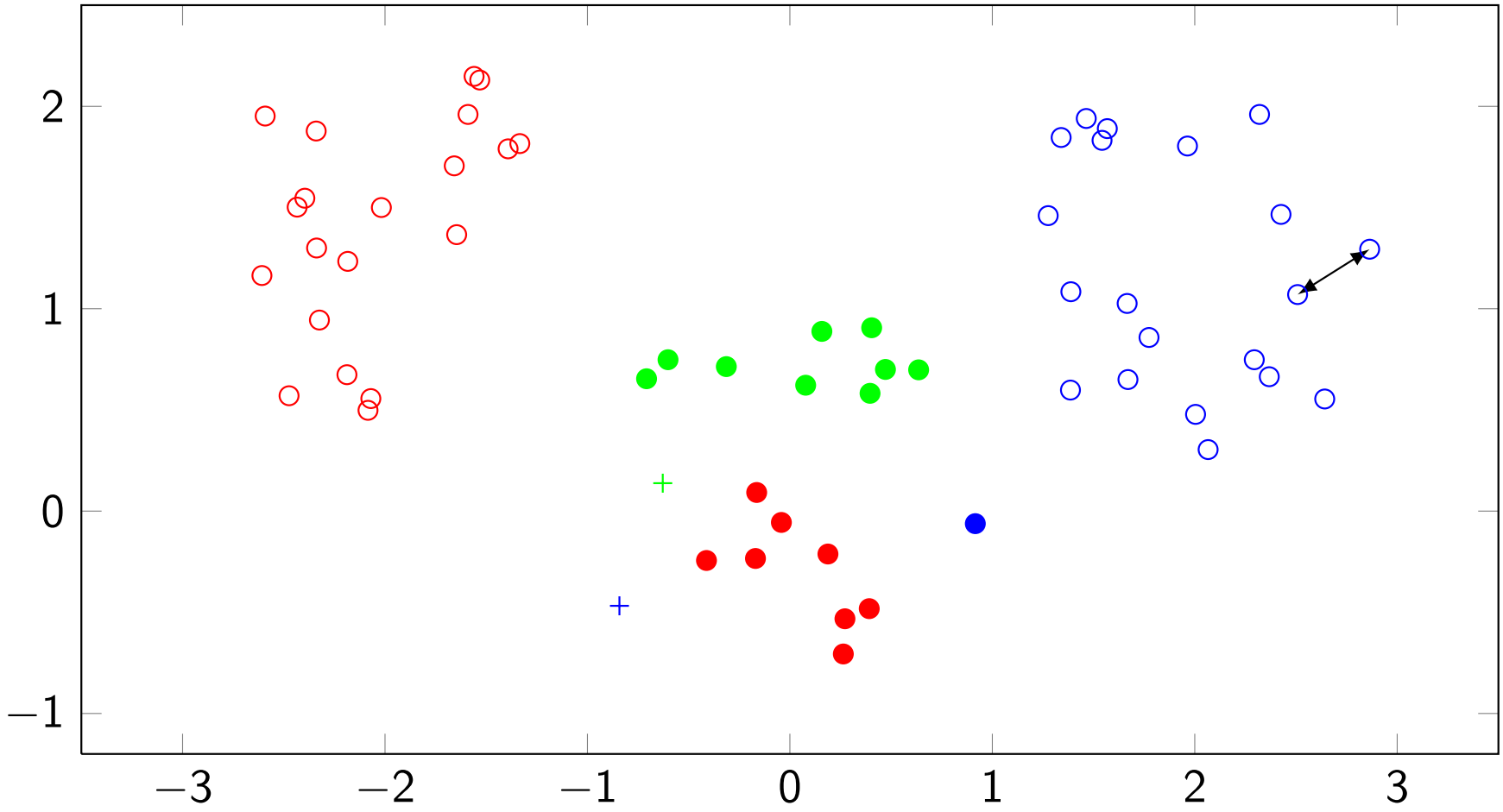
Demonstrace hierarchického shlukování (single link)



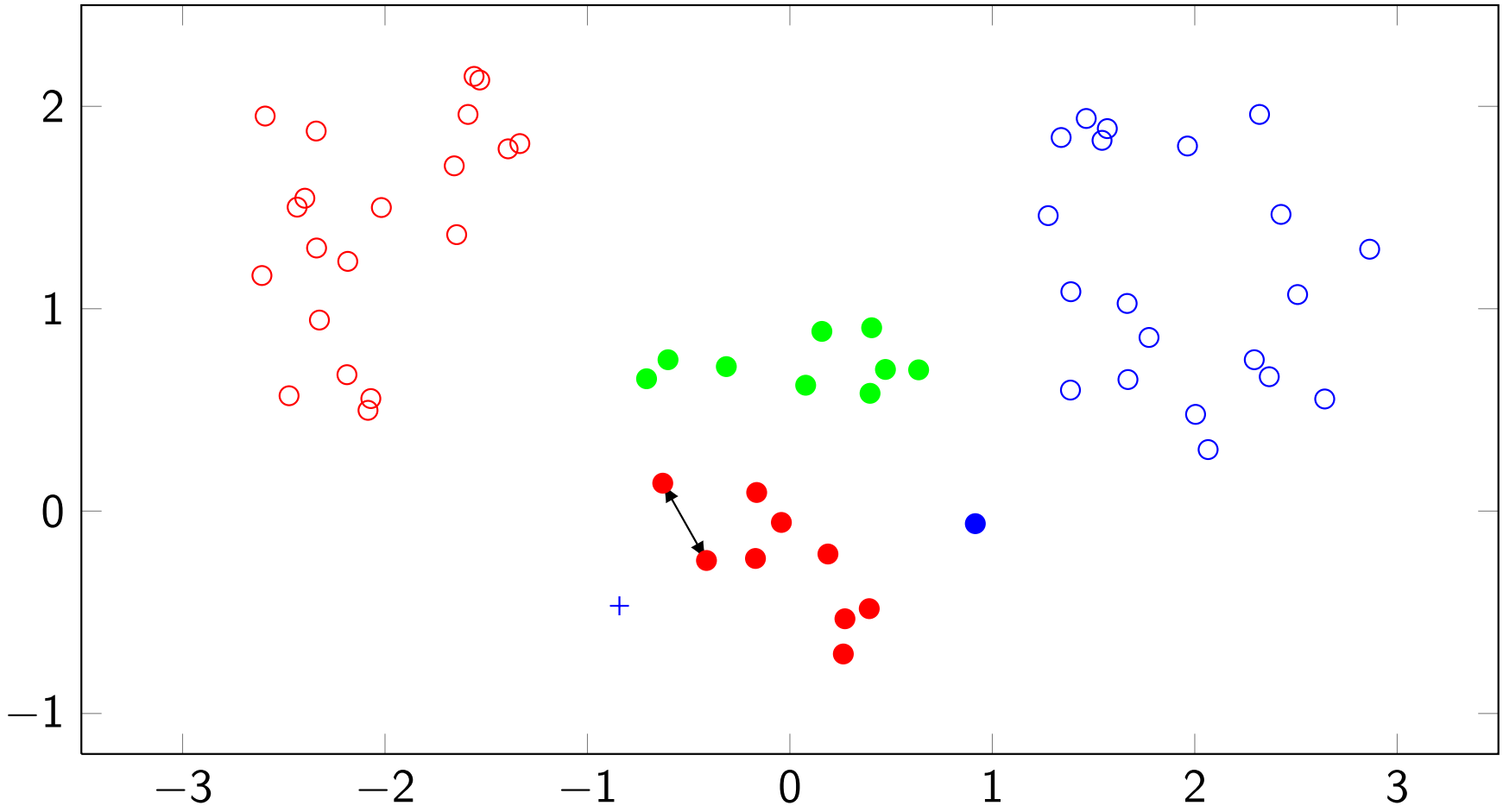
Demonstrace hierarchického shlukování (single link)



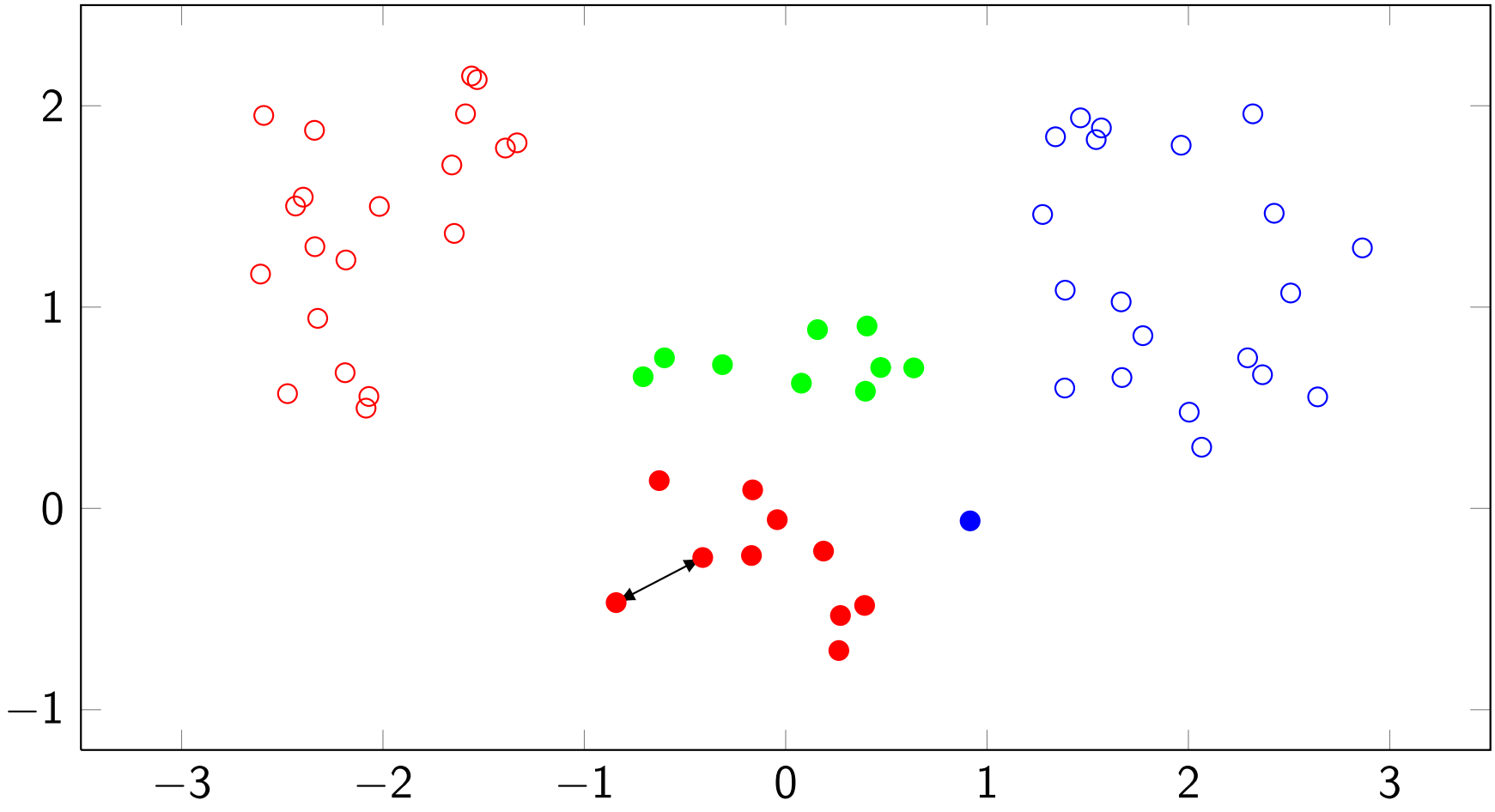
Demonstrace hierarchického shlukování (single link)



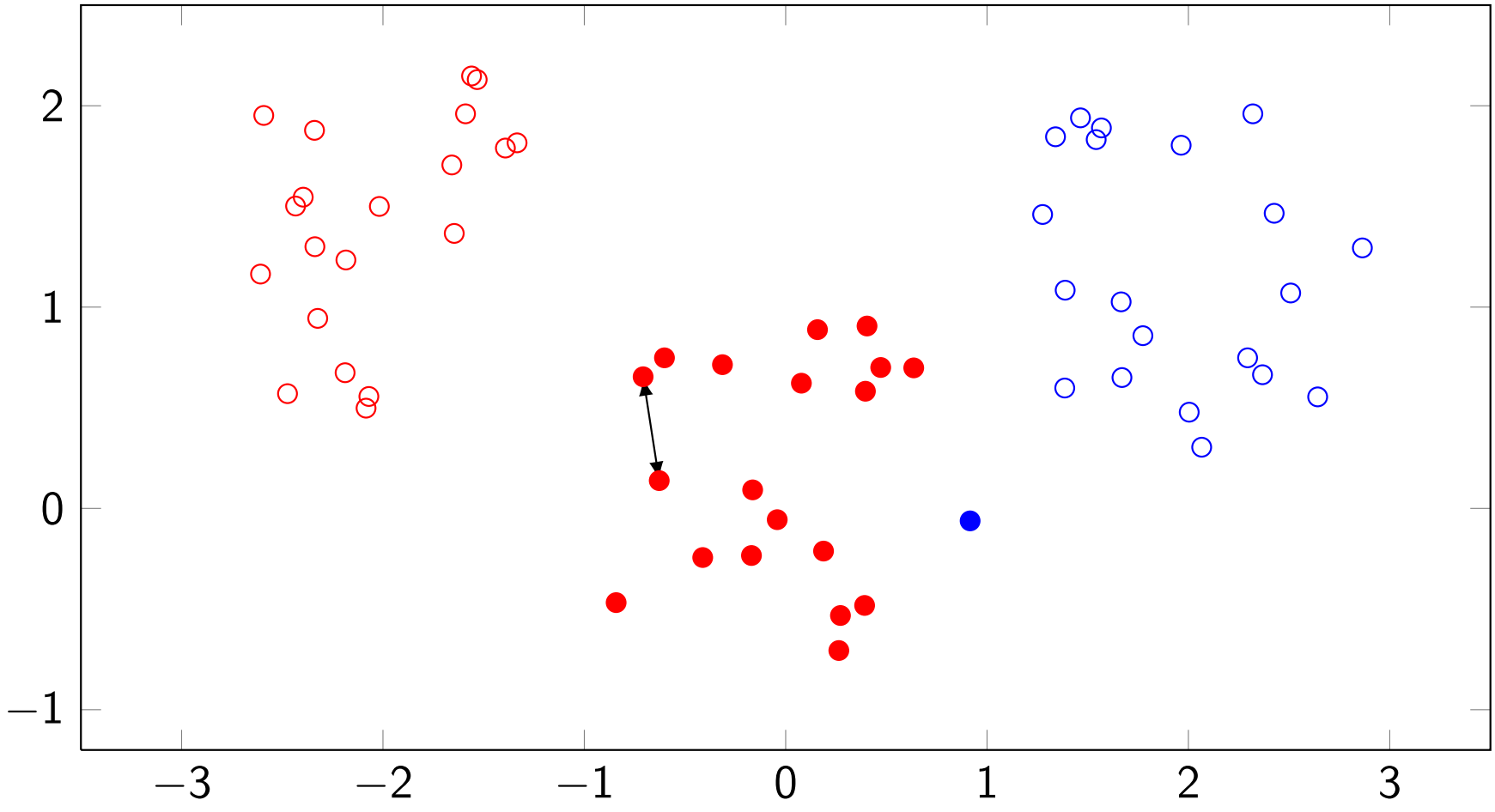
Demonstrace hierarchického shlukování (single link)



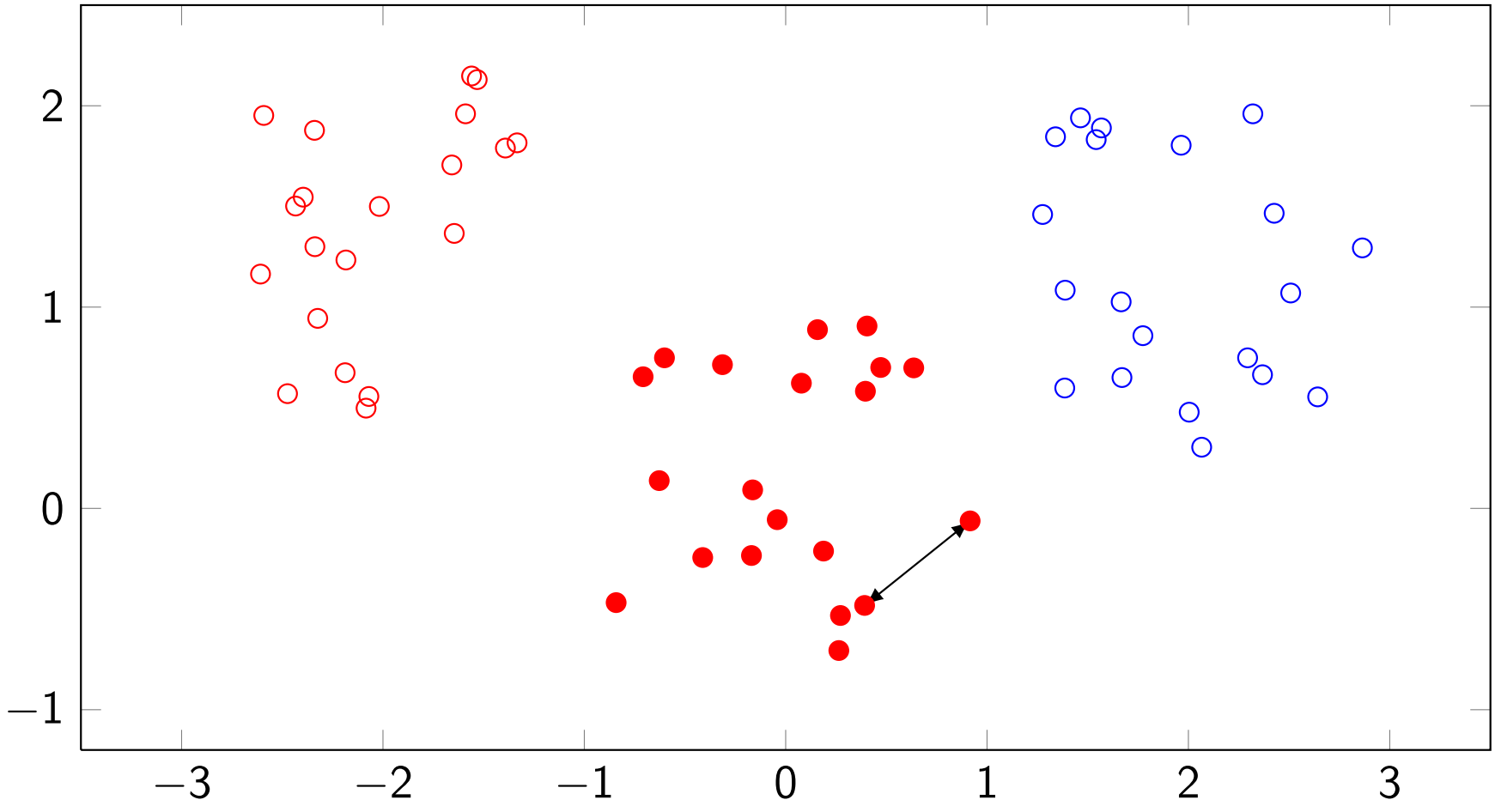
Demonstrace hierarchického shlukování (single link)



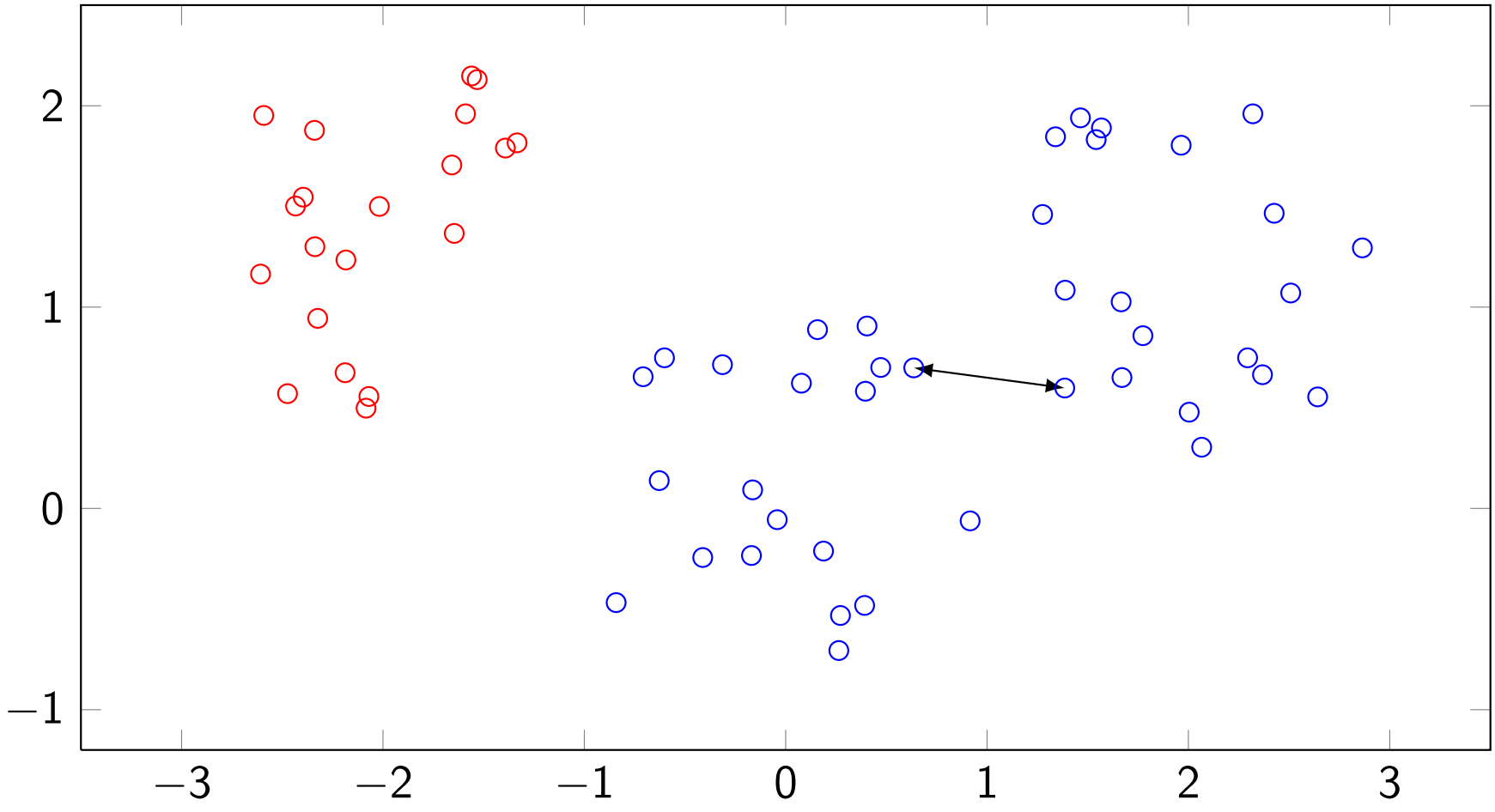
Demonstrace hierarchického shlukování (single link)



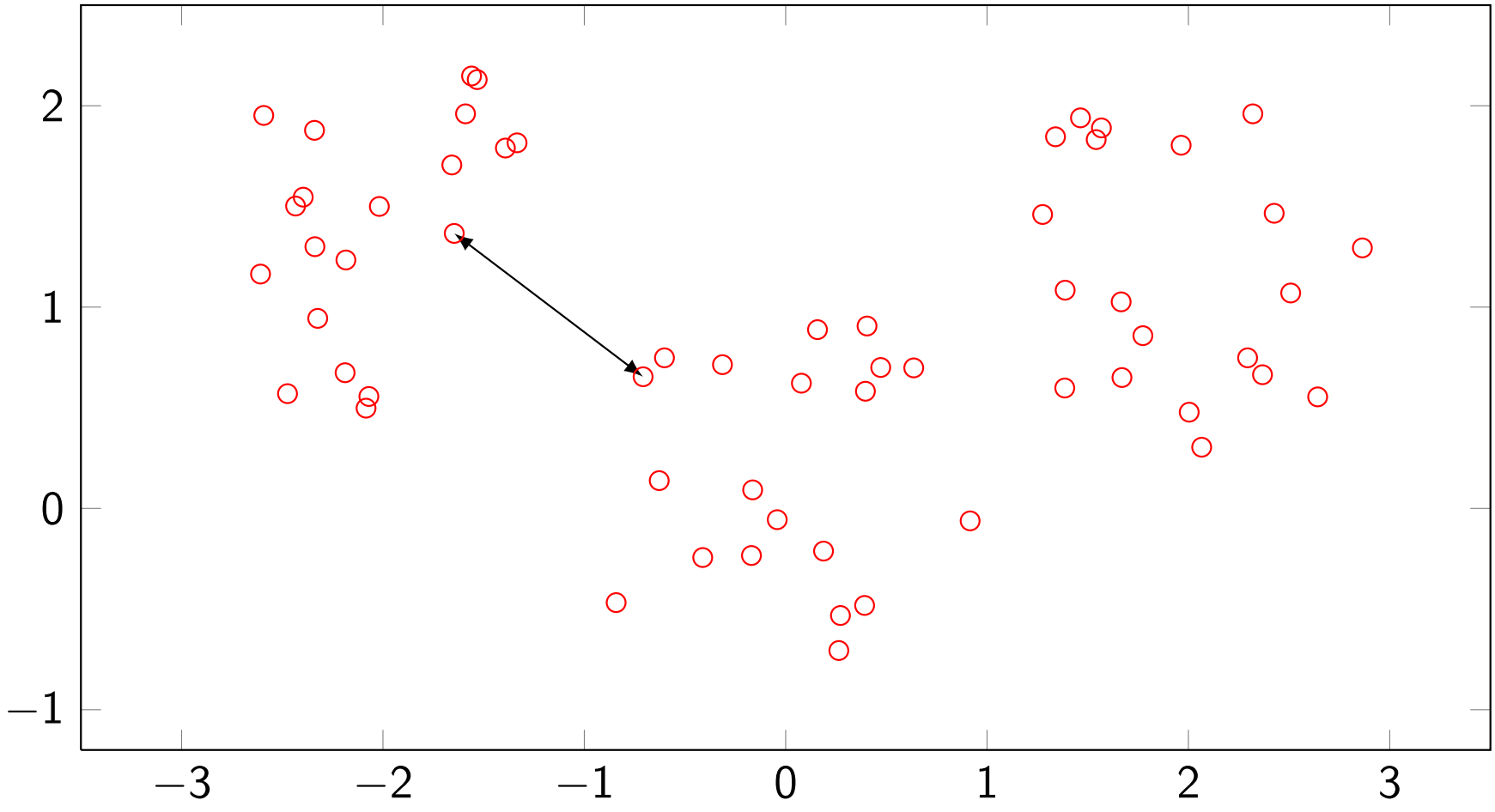
Demonstrace hierarchického shlukování (single link)



Demonstrace hierarchického shlukování (single link)



Demonstrace hierarchického shlukování (single link)



Shlukování založené na rozkladu

Algoritmus K-means shlukování
Když máme N datapointů v $\mathbb{R}^n$ a předpokládáme, že chceme zformovat c shluků.
Vypočítáme součet rozptylů mezi datapointy a množinou prototypů $\vec{v}_1, \vec{v}_2, \ldots, \vec{v}_c$:

$$Q = \sum_{i=1}^{c} \sum_{k=1}^{N} u_{ik} \|\vec{x}_k - \vec{v}_i\|,$$

kde $\|\,.\,\|$ je Euklidovská vzdálenost mezi $\vec{x}_k$ a $\vec{v}_i$.
Matice rozkladu $U = [u_{ik}]$, $i = 1, 2, \ldots, c; k = 1, 2, \ldots, N$

- její role je přidělit datapointy shlukum.
- prvky matice U jsou binární:

$$u_{ik} = \begin{cases} 1 & \text{pokud datapoint } k \text{ patří do shluku } i, \\ 0 & \textit{jinak}. \end{cases}$$

Matice rozkladu splňuje následující podmínky

- každý shluk je netriviální, tj. neobsahuje všechny datapointy a jsou neprázdné

$$0 < \sum_{k=1}^{N} u_{ik} < N, \quad i = 1, 2, \ldots, c$$

- každý datapoint patří do jednoho shluku

$$\sum_{i=1}^{c} u_{ik} = 1, \quad k = 1, 2, \ldots, N$$

Kolekce všech matic rozkladu bude značena $\mathbb{U}$
Jako výsledek minimalizace Q konstruujeme matici rozkladu a množinu prototypů.
Formálně vyjadřujeme tento konstrukt jako optimalizační problém s omezeními:

$$\text{Minimální } Q \text{ vzhledem k } \vec{v}_1, \vec{v}_2, \ldots, \vec{v}_c \text{ a } \mathbf{U} \in \mathbb{U}.$$

Existuje mnoho přístupů k této optimalizaci.
Nejběžnější je K-means.

Algoritmus K-means shlukování

inicializuj prototypy $\vec{v}_i$, $i = 1, 2, \ldots, c$ (např. náhodně)

iteruj, dokud se Q neustálí:

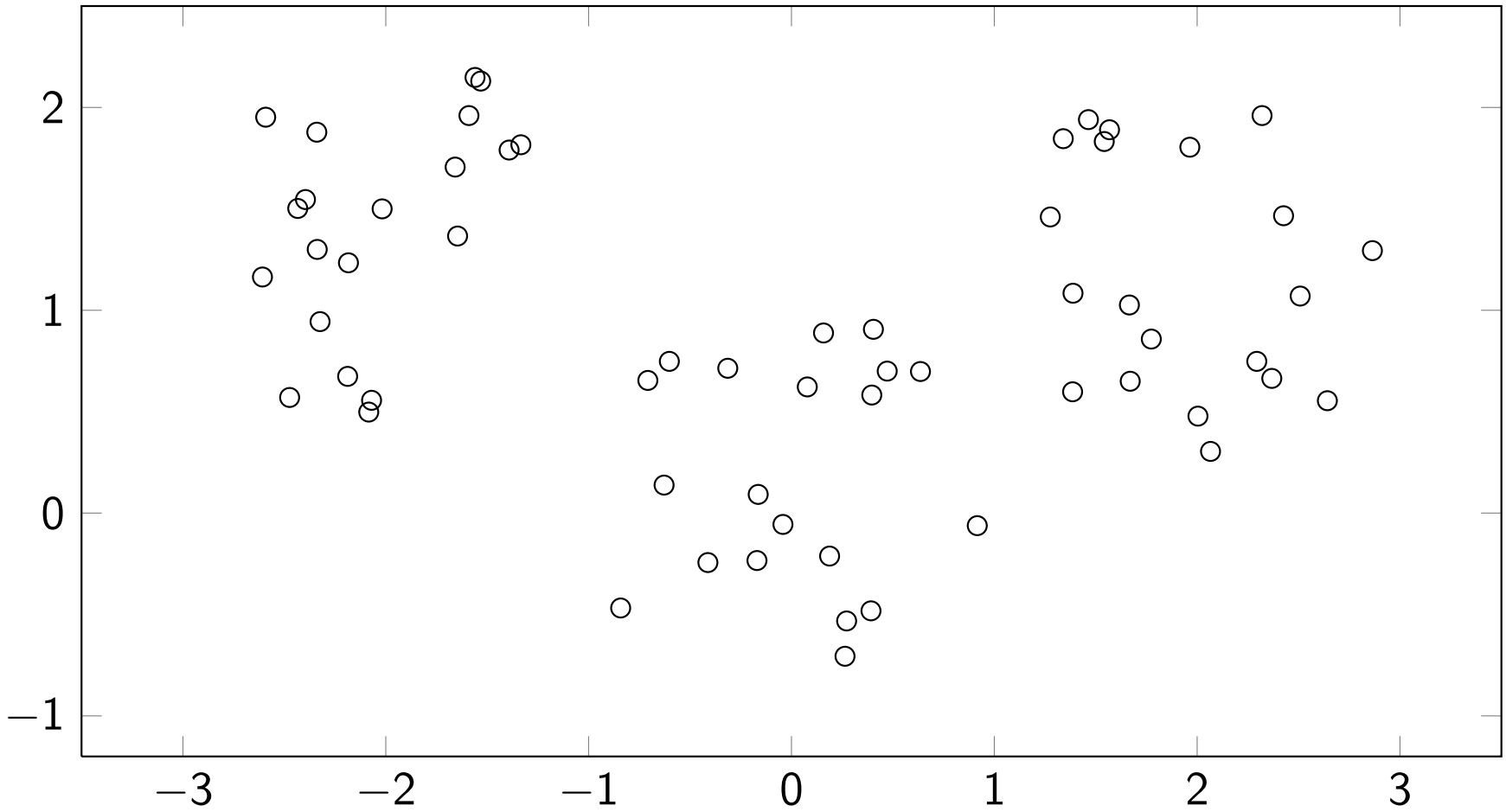
- zkonstruuj rozkladovou matici U takto:

$$u_{ik} = \begin{cases} 1, & \text{pokud } d(\vec{x}_k, \vec{v}_i) = \min d(\vec{x}_k, \vec{v}_i), \\ 0, & \text{jinak.} \end{cases}$$

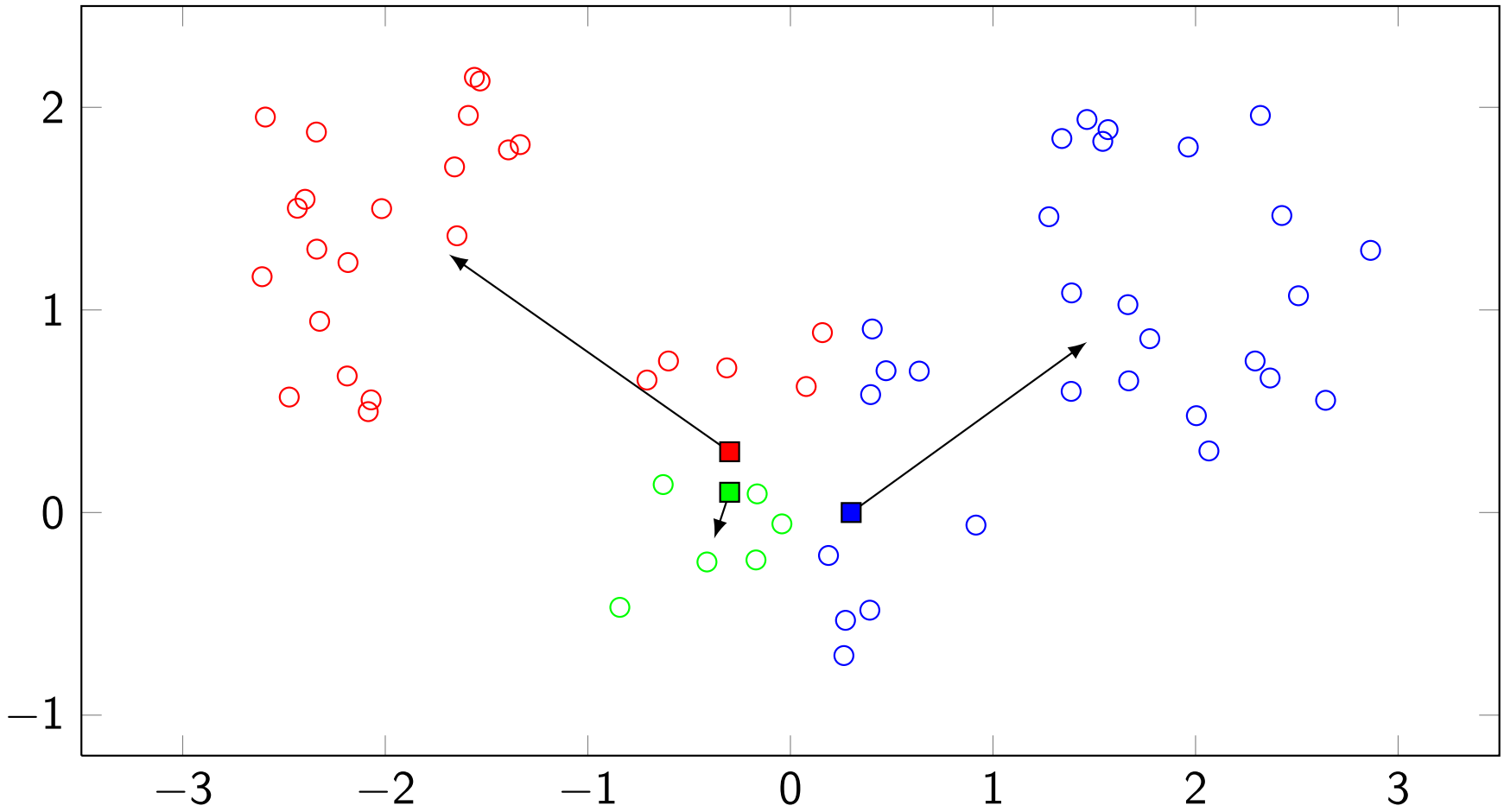
- změň prototypy výpočtem průměru

$$\vec{v}_i = \frac{\sum_{k=1}^{N} u_{ik} \vec{x}_k}{\sum_{k=1}^{N} u_{ik}}$$

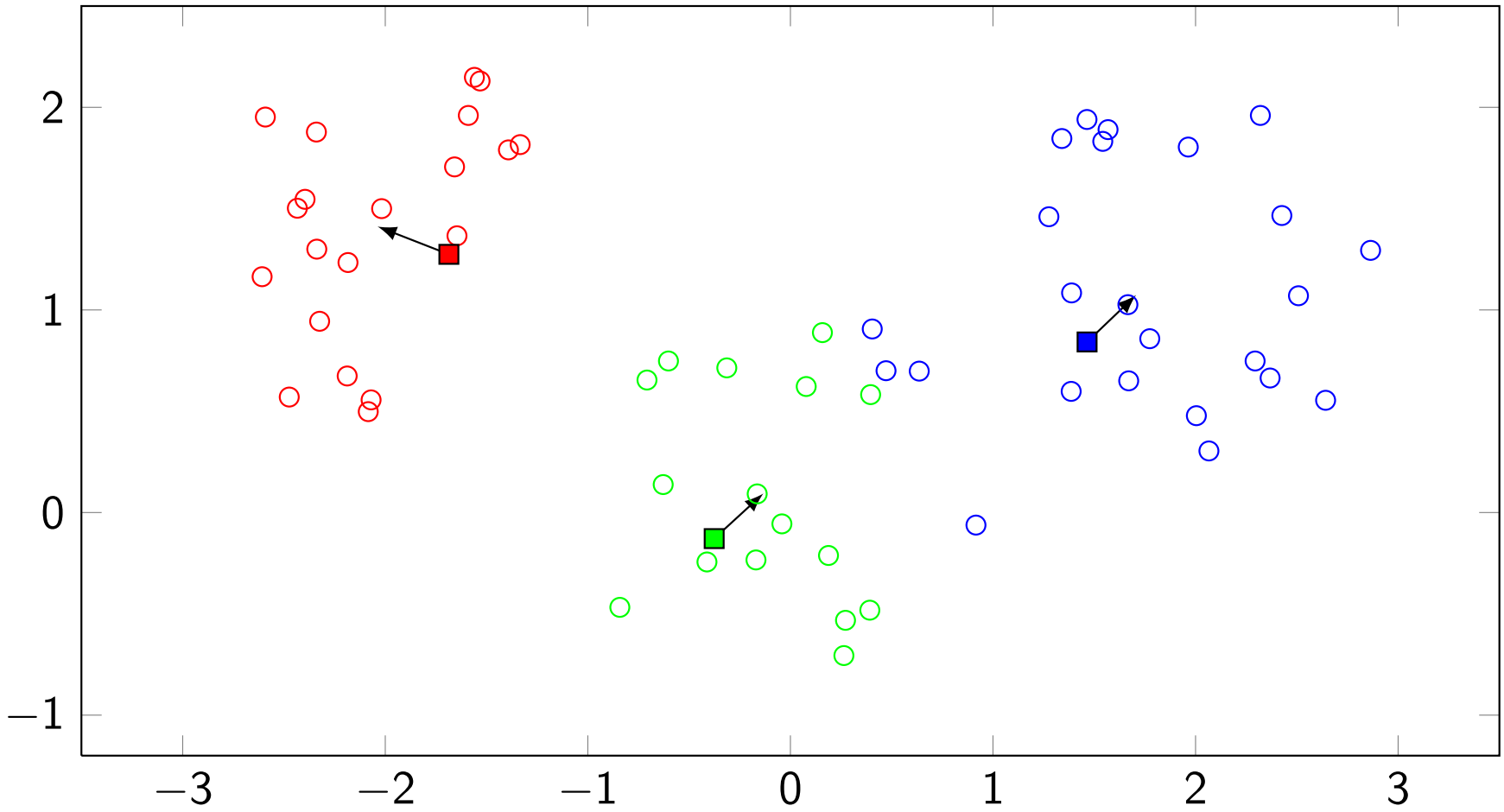
Demonstrace algoritmu K-means shlukování



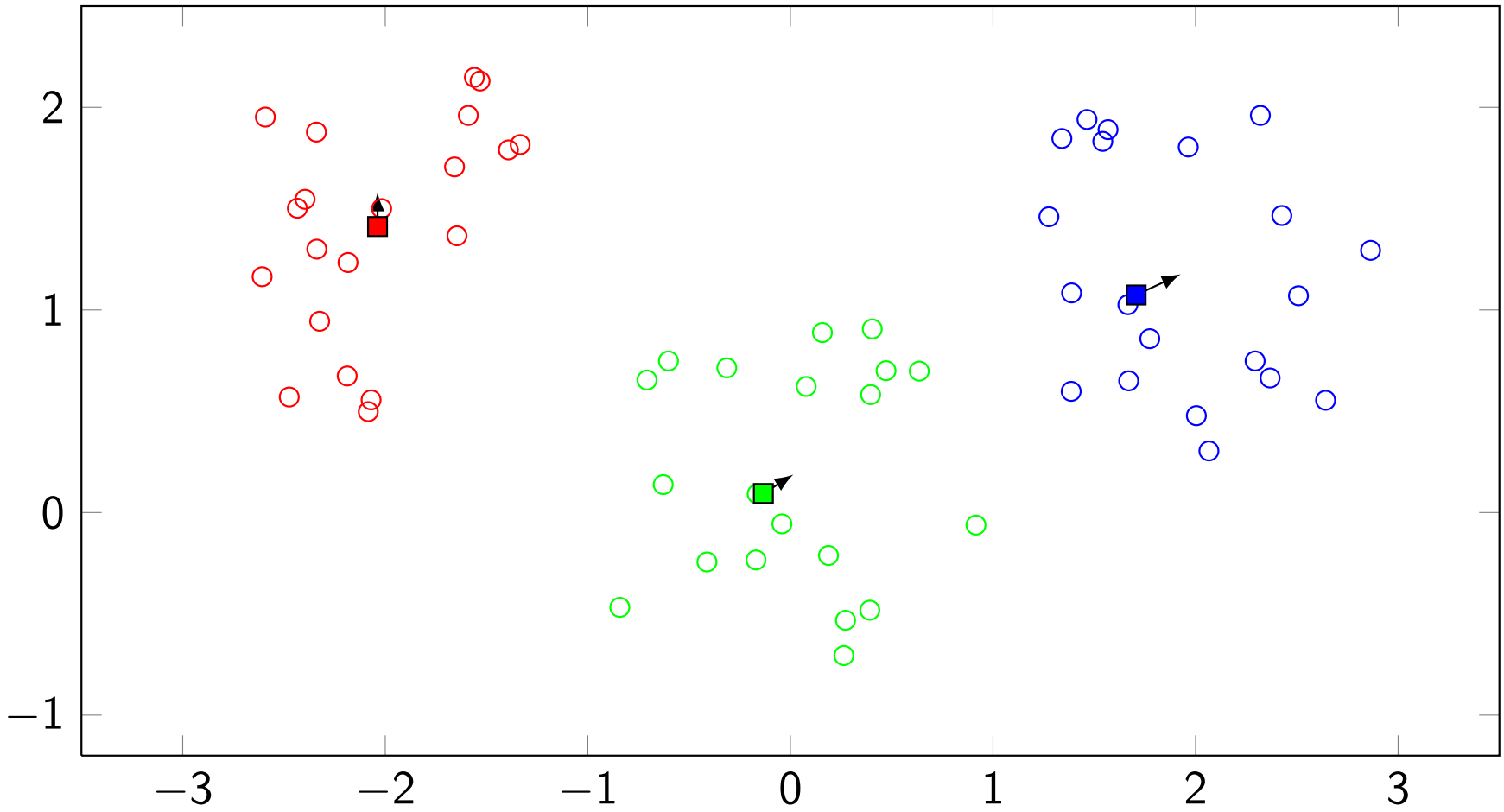
Demonstrace algoritmu K-means shlukování



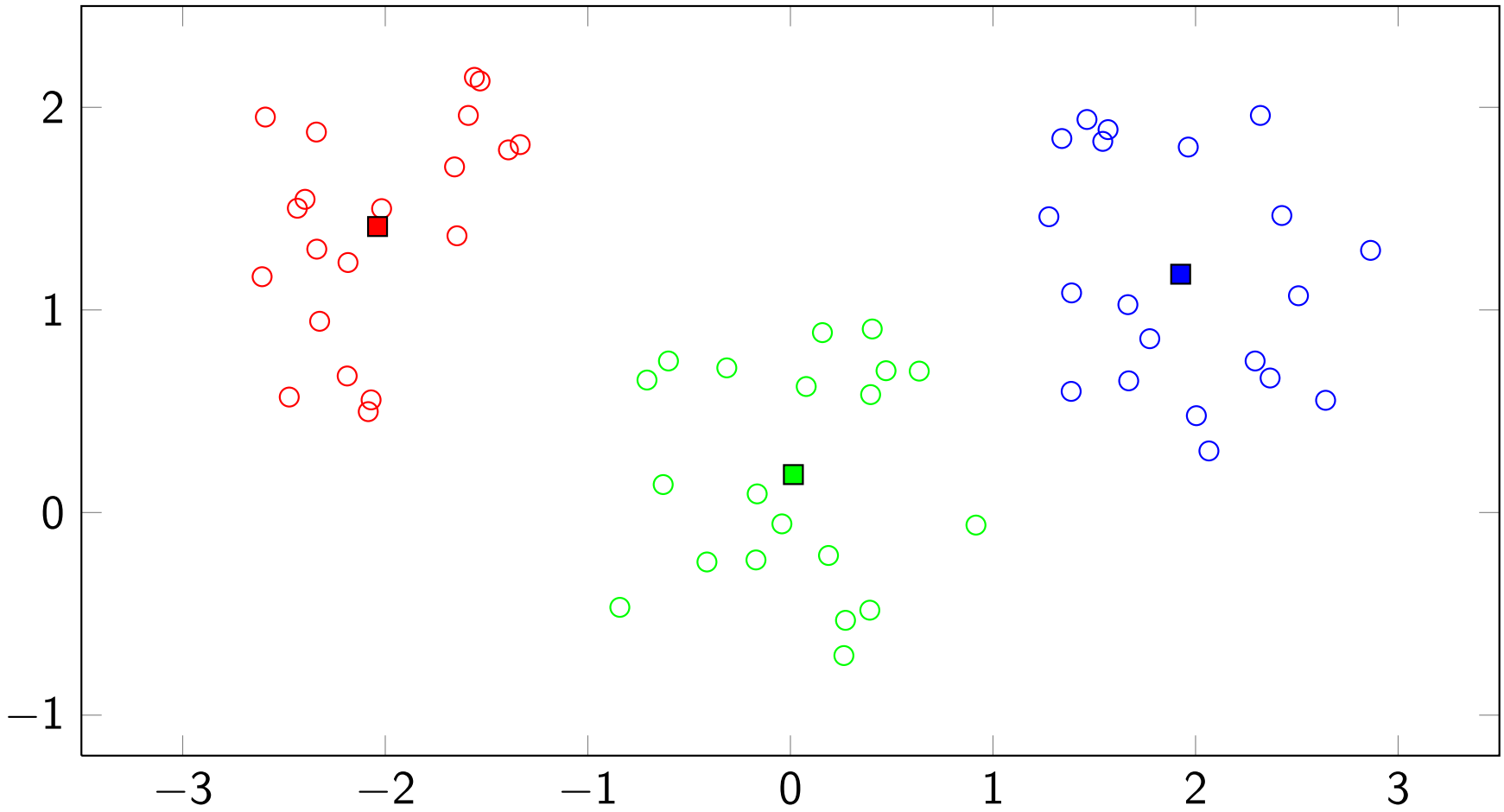
Demonstrace algoritmu K-means shlukování



Demonstrace algoritmu K-means shlukování



Demonstrace algoritmu K-means shlukování

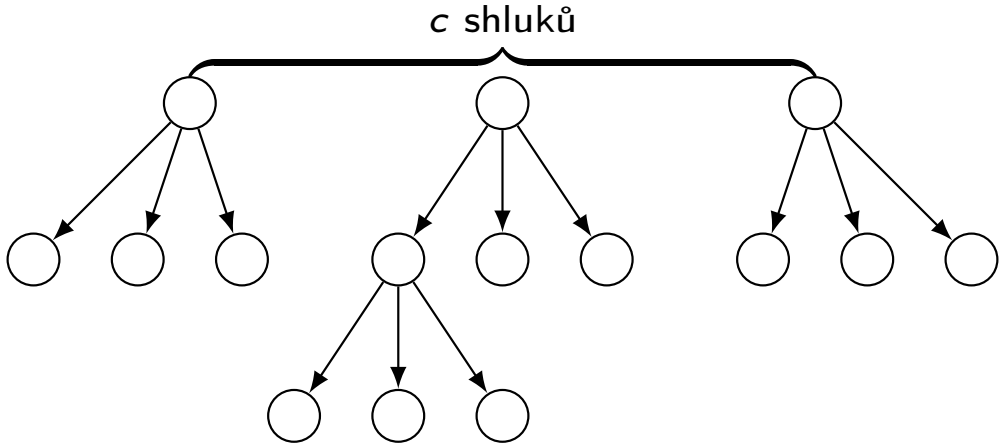


Rostoucí hierarchie shluků

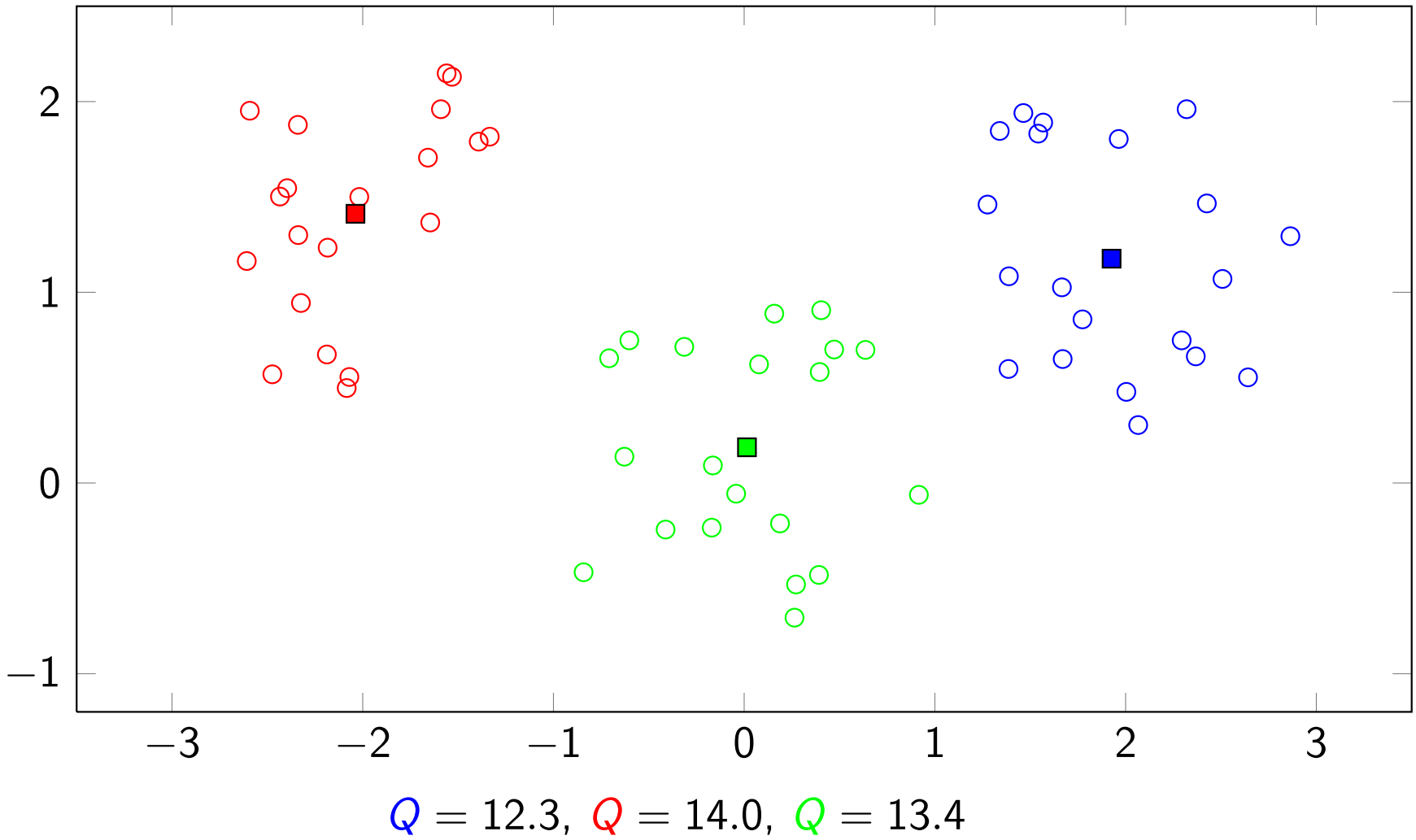
Shlukování založené na rozkladu může být organizováno v nějaké hierarchické topologii shluků, která umožní odhalit strukturu postupně a redukovat tak výpočení nároky.

Podstata tohoto přístupu je následující:

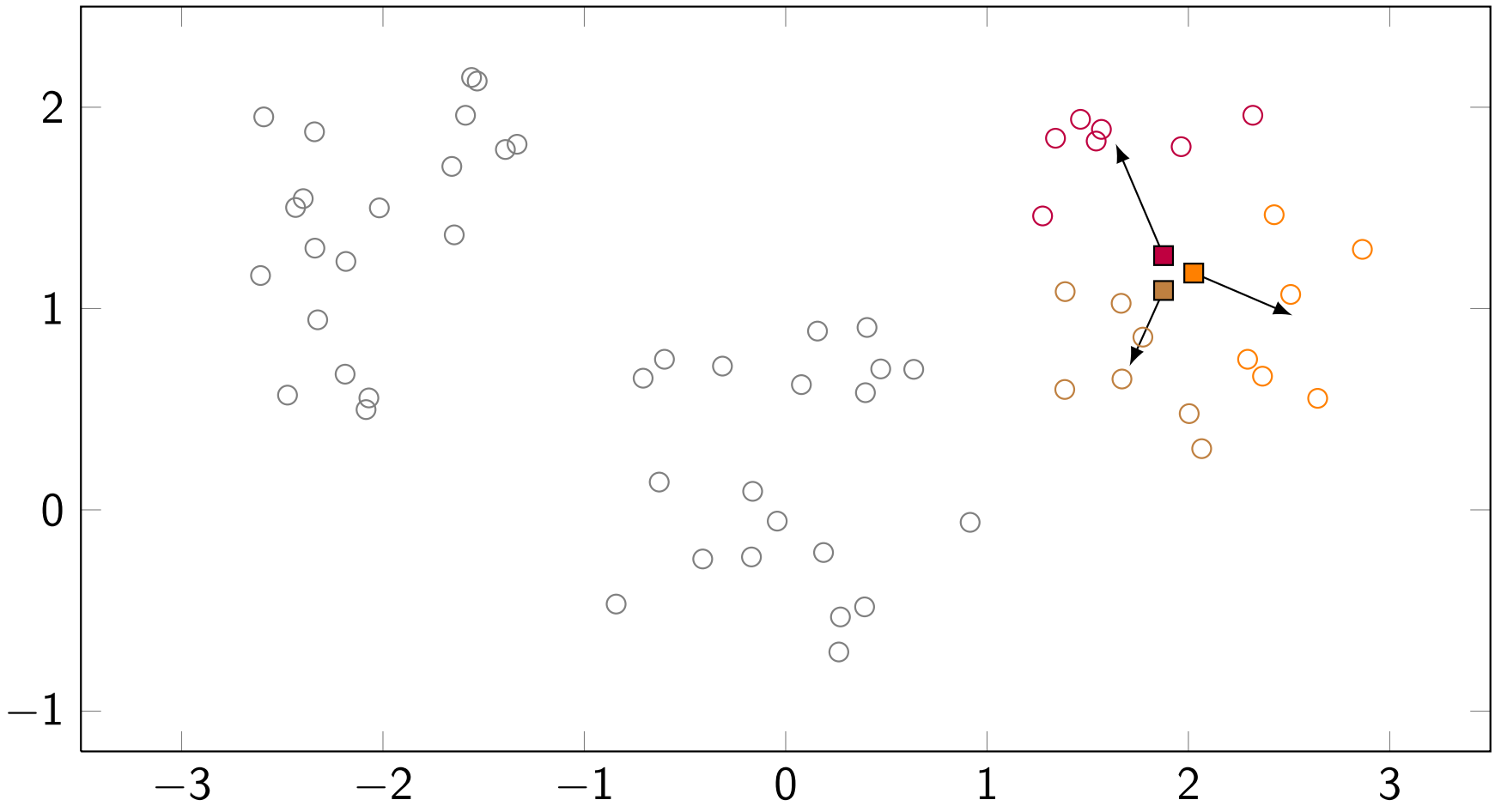
- Začneme s malým počtem shluků $c = 3 \ldots 5$.
- Určíme hodnotu cílové funkce pro každý shluk zvlášť, ozn. Q_i.
- Provádíme shlukování na ve shluku s největším Q_i; rozdělíme data z odpovídajícího shluku do dalších c shluků.
- To opakujeme, dokud cílová funkce každého shluku je menší než předem definovaná hranice, nebo se nedosáhne maximálního počtu shluků.



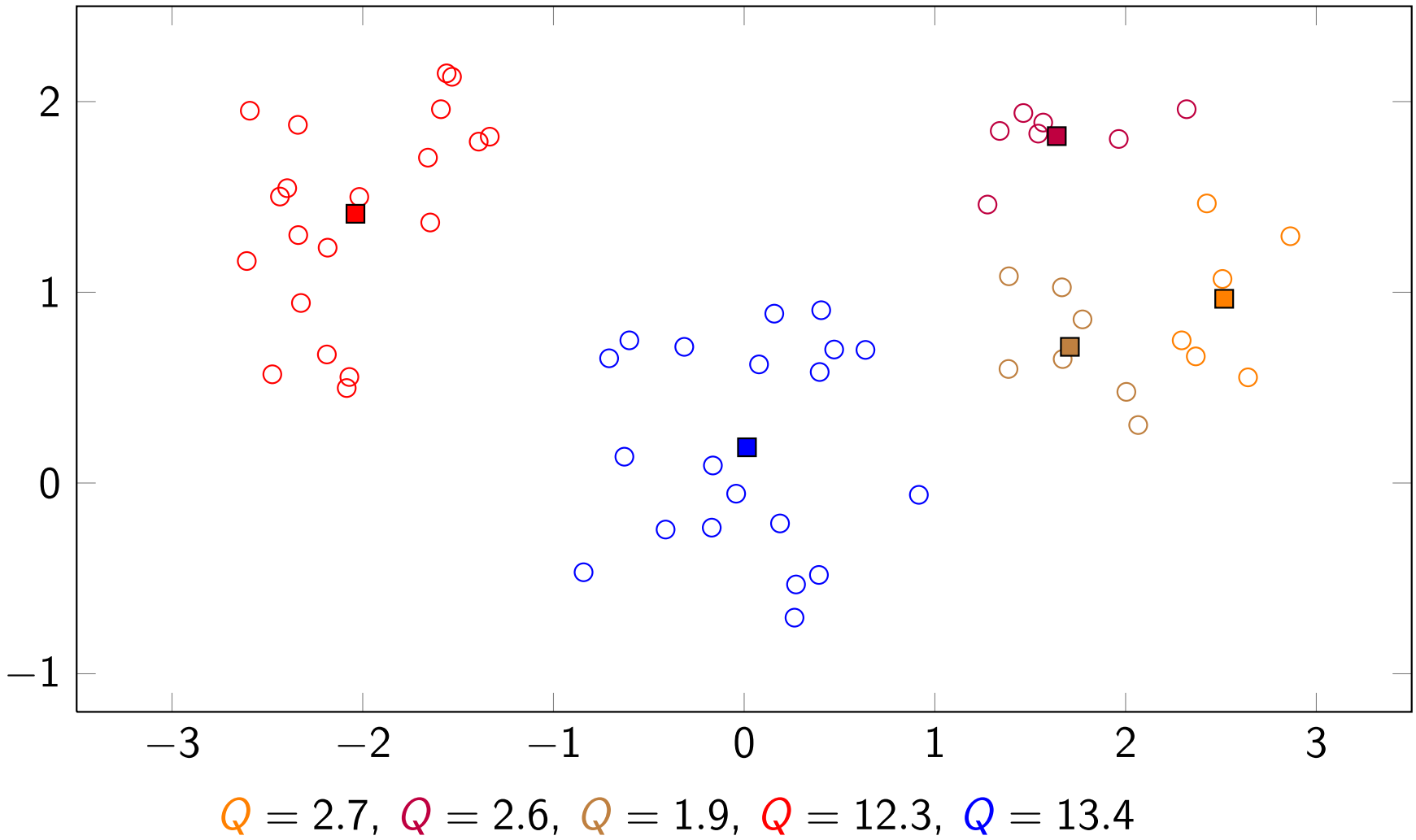
Demonstrace rostoucí hierarchie shluků



Demonstrace rostoucí hierarchie shluků



Demonstrace rostoucí hierarchie shluků



K-medoids

Problémy prototypů

- Problém s prototypy shluků je, že se počítají na základě všech prvků. Pokud jsou data ovlivněna šumem, ovlivní to i prototypy.
- Navíc, prototyp není jeden z datapointů.

Místo průměru se použije robustnější reprezentant: *medoid*

. . . začněme pojmem *medián*: $\mathrm{med}\{x_1, x_2, \ldots, x_N\}$ reálných čísel.
Předpokládáme, že data jsou uspořádaná: $x_1 \leq x_2 \leq \cdots \leq x_3$

medián je definován následovně:

$$\text{medián} = \begin{cases} x_{(N+1)/2} & \text{pokud } N \text{ je liché} \\ (x_{N/2} + x_{N/2+1})/2 & \text{pokud } N \text{ je sudé} \end{cases}$$

nebo

$$\text{medián} = \begin{cases} x_{(N+1)/2} & \text{pokud } N \text{ je liché} \\ x_{N/2} & \text{pokud } N \text{ je sudé} \end{cases}$$

Výhody mediánu oproti průměru

- *robustnost* proti odlehlým bodům
- *interpretovatelnost* prototypem je vždy prvek dat, a tedy se dá snadno interpretovat.

Dá se jednoduše ukázat, že medián je řešení následujícího optimalizačního problému

$$\min_{i=1}^{N} \sum_{k=1}^{N} |x_k - x_i| = \sum_{k=1}^{N} |x_k - \mathrm{med}|,$$

kde med označuje medián
(v této cílové funkci vidíme Hammingovu vzdálenost).

Obecně, pro n-dimenzionální data, cílová funkce je ve formě

$$Q = \sum_{i=1}^{c} \sum_{k=1}^{N} u_{ik} \|\vec{x}_k - \vec{v}_i\| = \sum_{i=1}^{c} \sum_{k=1}^{N} \sum_{j=1}^{n} u_{ik} |x_{kj} - v_{ij}|$$

Nalezení minima této funkce vede k centrům shluků; ovšem to je dost zdlouhavé.

Abychom se vyhnuli zdlouhavému optimalizačnímu procesu, je potřeba zahrnout jiná opatření.

- **PAM** (**P**artitioning **A**round **M**edoids)
- **CLARA** (**C**lustering **LAR**ge **A**pplications)

PAM (**P**artitioning **A**round **M**edoids)
Idea je reprezentovat strukturu dat kolekcí *medoidů* (nejcentrálněji položených datapointů).
V podstatě pracuje stejně jako k-means, ale místo průměrů se uvažují medoidy:

Algoritmus:
inicializuj prototypy $\vec{v}_i$, $i = 1, 2, \ldots, c$ (např. náhodně vyber datapointy)

iteruj, dokud se Q neustálí:

- zkonstruuj rozkladovou matici U takto:

$$u_{ik} = \begin{cases} 1, & \text{pokud } d(\vec{x}_k, \vec{v}_i) = \min d(\vec{x}_k, \vec{v}_i), \\ 0, & \text{jinak.} \end{cases}$$

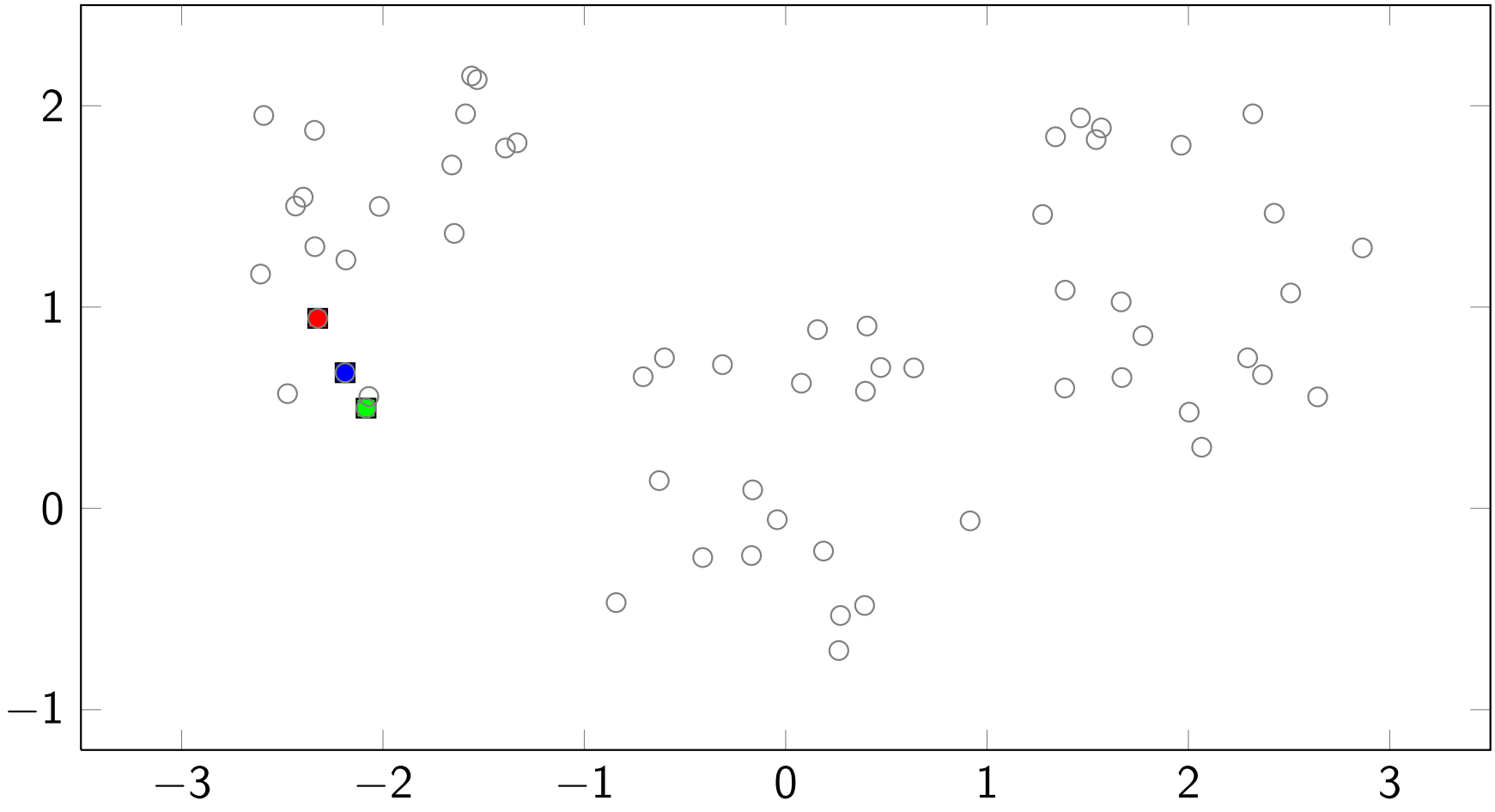
- změň prototypy na medoidy

$$\vec{v}_i = \mathrm{med}\{\vec{x}_k \mid u_{ik} = 1\}$$

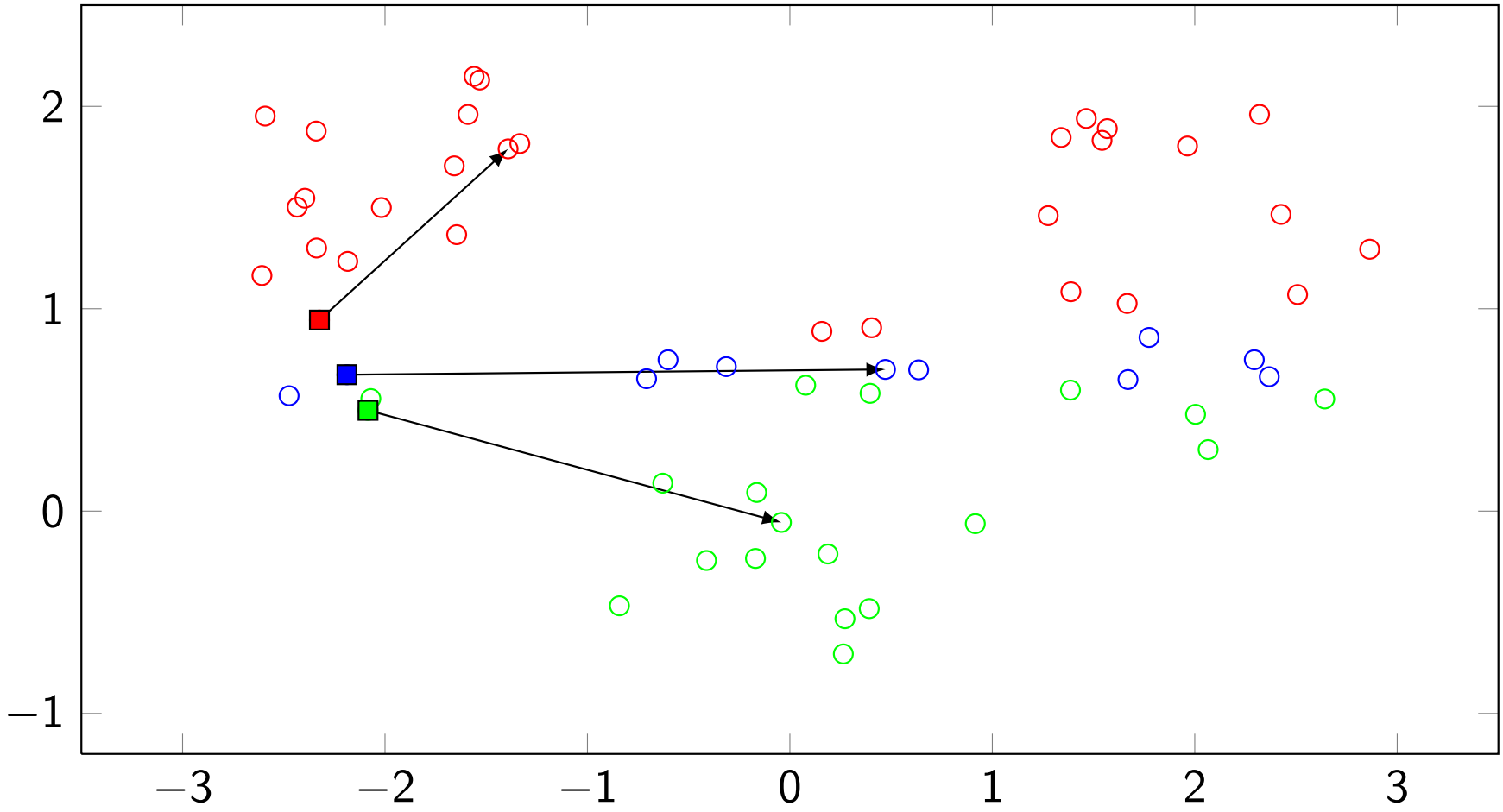
Použitelné jen na malé datasety (100 datapointů, 5 shluků)

CLARA (**C**lustering **LAR**ge **A**pplications) zachází s většími datasety: vybere vzorek (několik), aplikuje PAM na vzorky, vybere ten nejlepší.

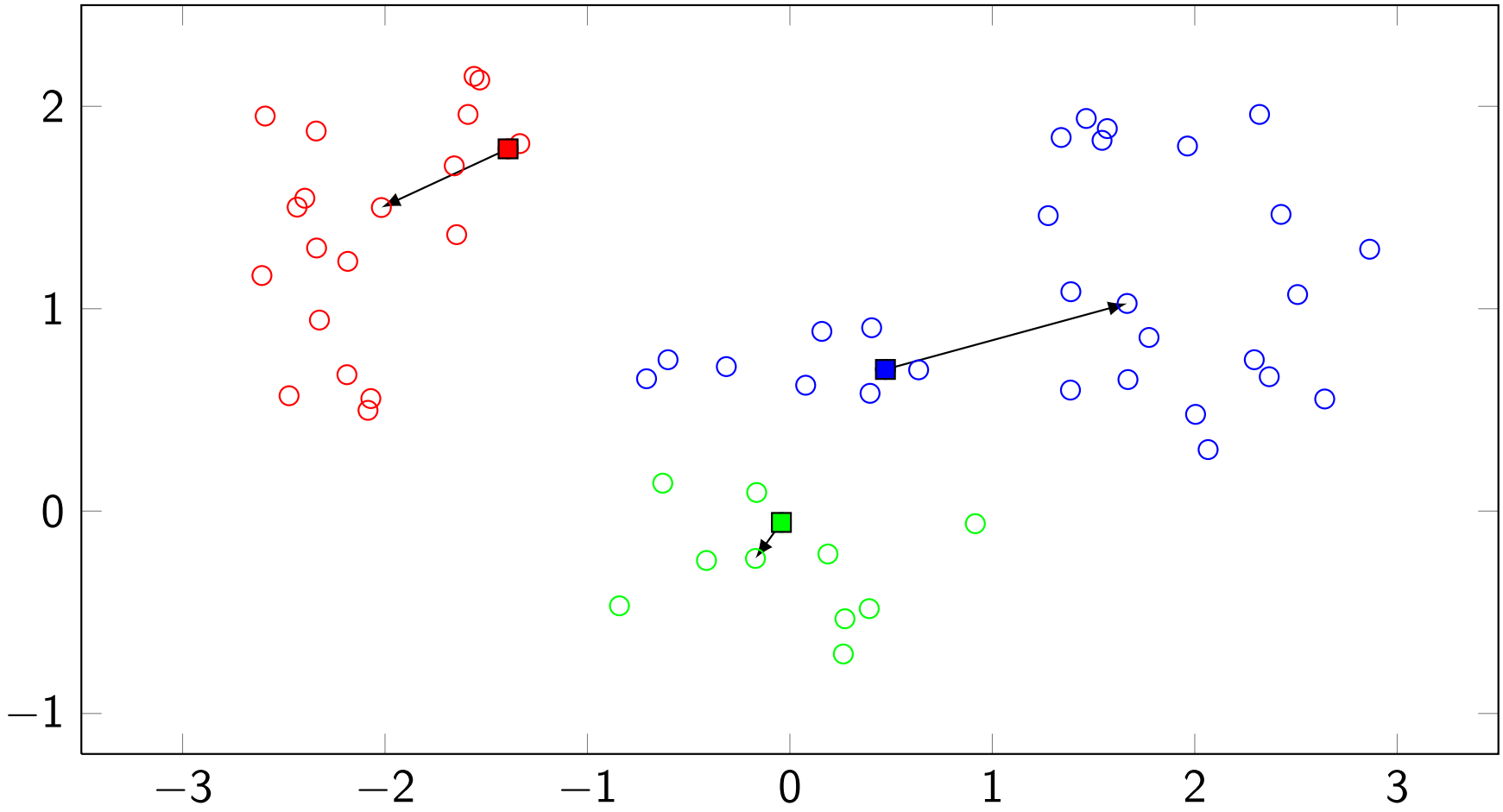
Demonstrace PAM



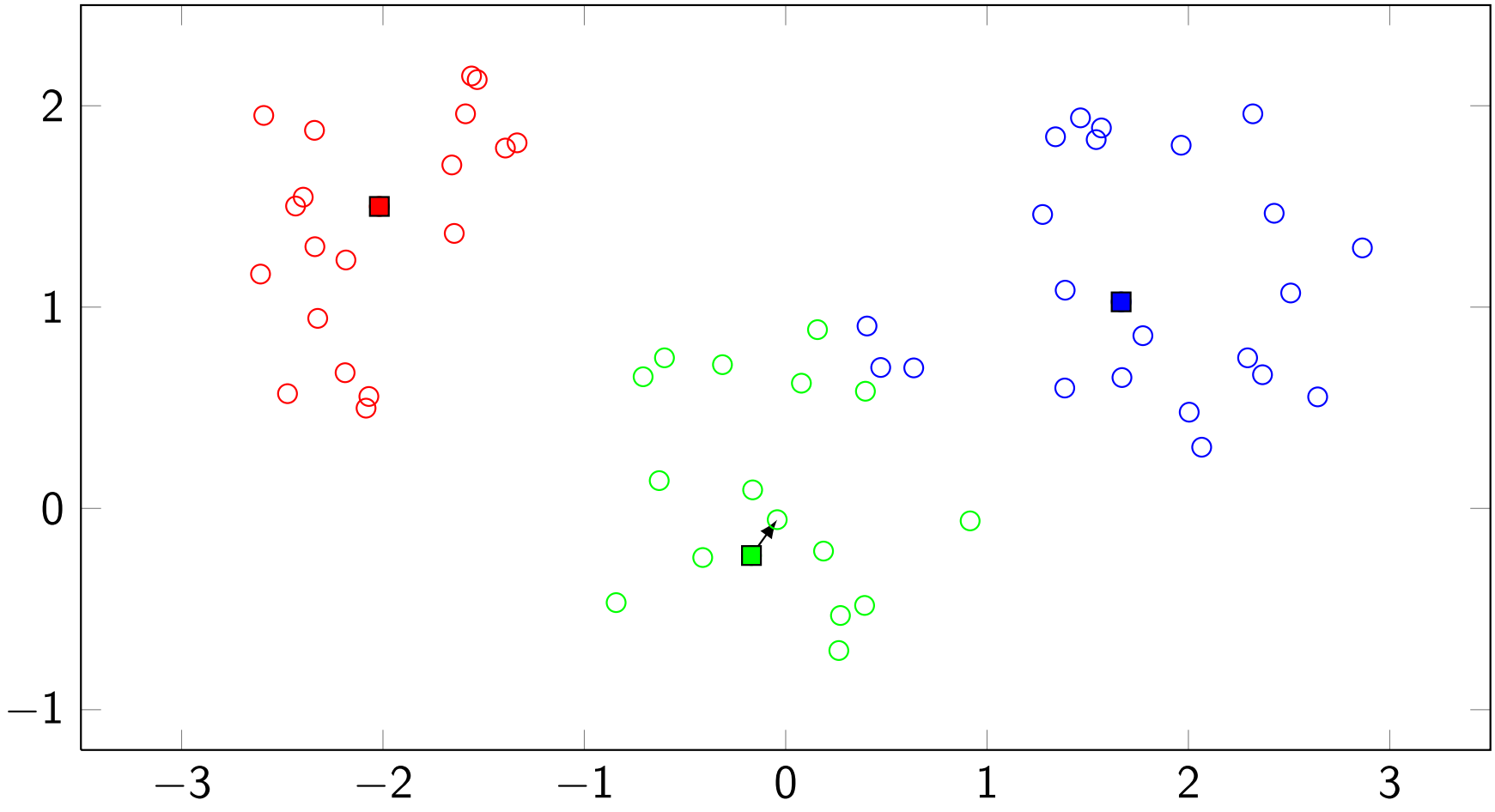
Demonstrace PAM



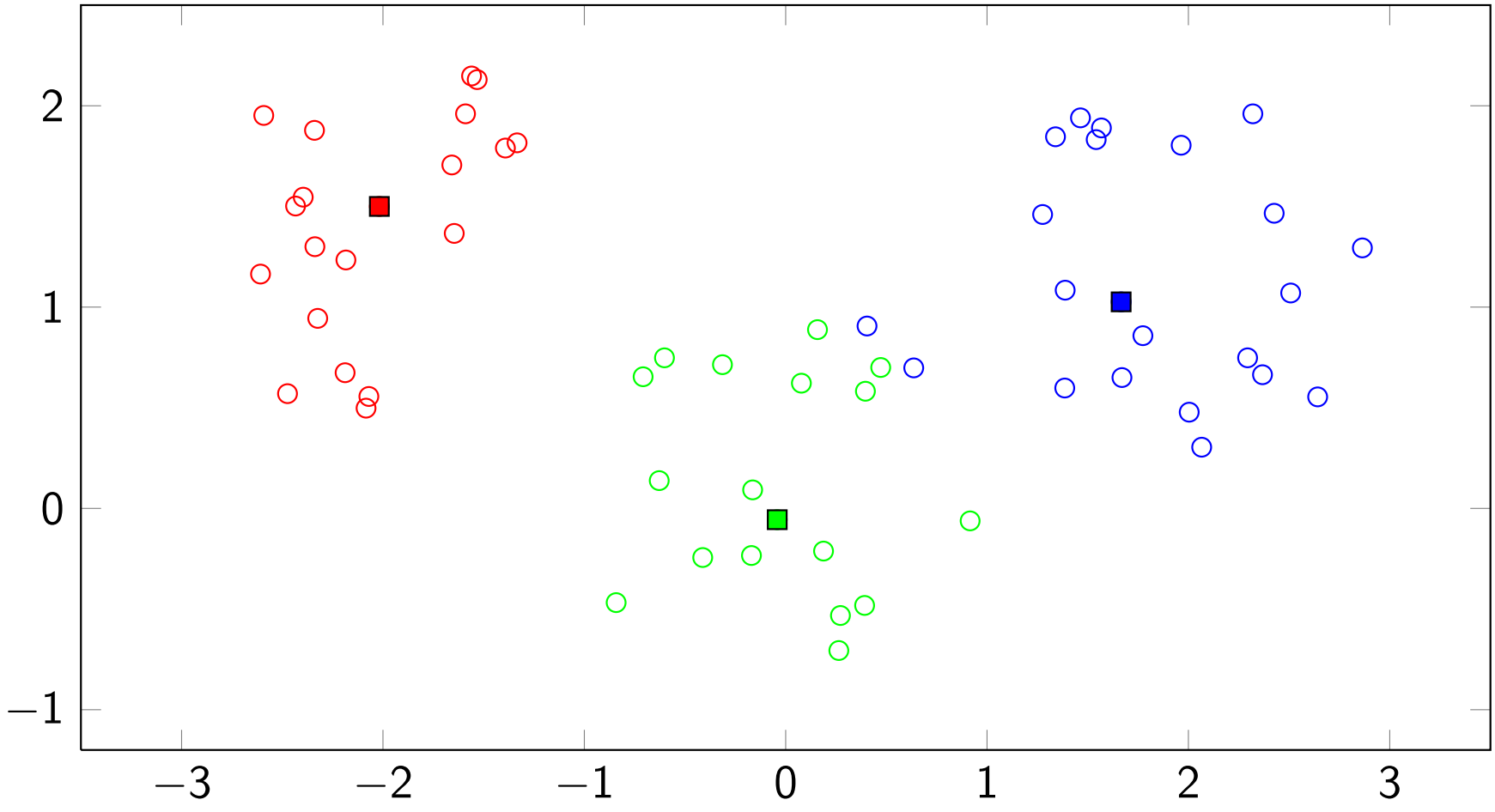
Demonstrace PAM



Demonstrace PAM



Demonstrace PAM



Fuzzy C-means

Inicializace: Vyber

- počet shluků c,
- kritérium zastaveni ϵ,
- fuzzyfikační koeficient m – číslo větší něž 1.

Iteruj, dokud nenastane kritérium zastavení:

- zkonstruuj rozkladovou matici U takto:

$$u_{ik} = \frac{1}{\sum_{j=1}^{c} \left(\frac{\|\vec{x}_k - \vec{v}_i\|}{\|\vec{x}_k - \vec{v}_i\|} \right)^{\frac{2}{m-1}}}$$

- změň prototypy výpočtem průměru

$$\vec{v}_i = \frac{\sum_{k=1}^{N} u_{ik} \vec{x}_k}{\sum_{k=1}^{N} u_{ik}}$$

Kritérium zastavení

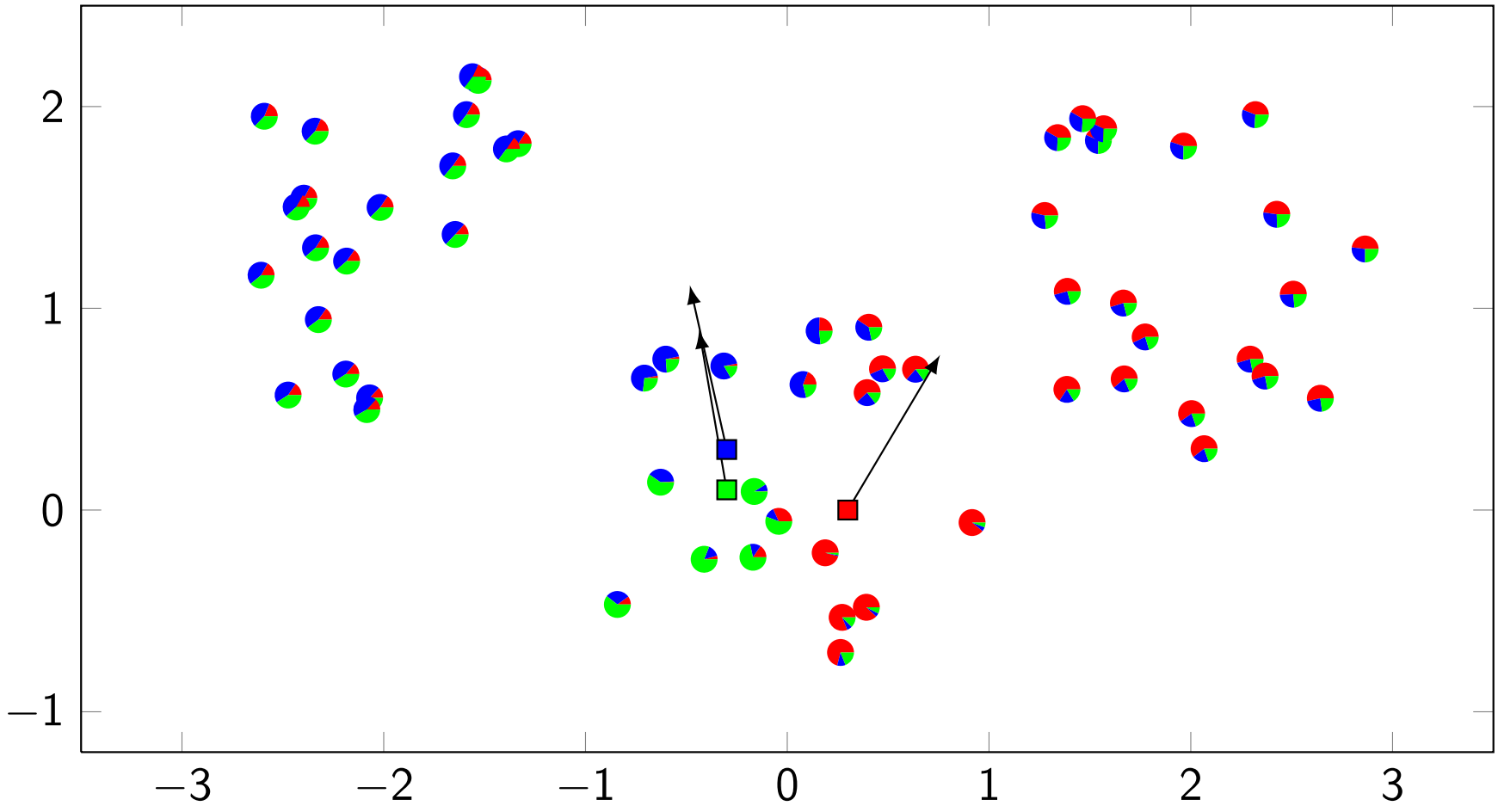
obvykle vzdálenost mezi dvěma po sobě jdoucími maticemi rozkladu: $U(\text{iter})$ a $U(\text{iter}+1)$.

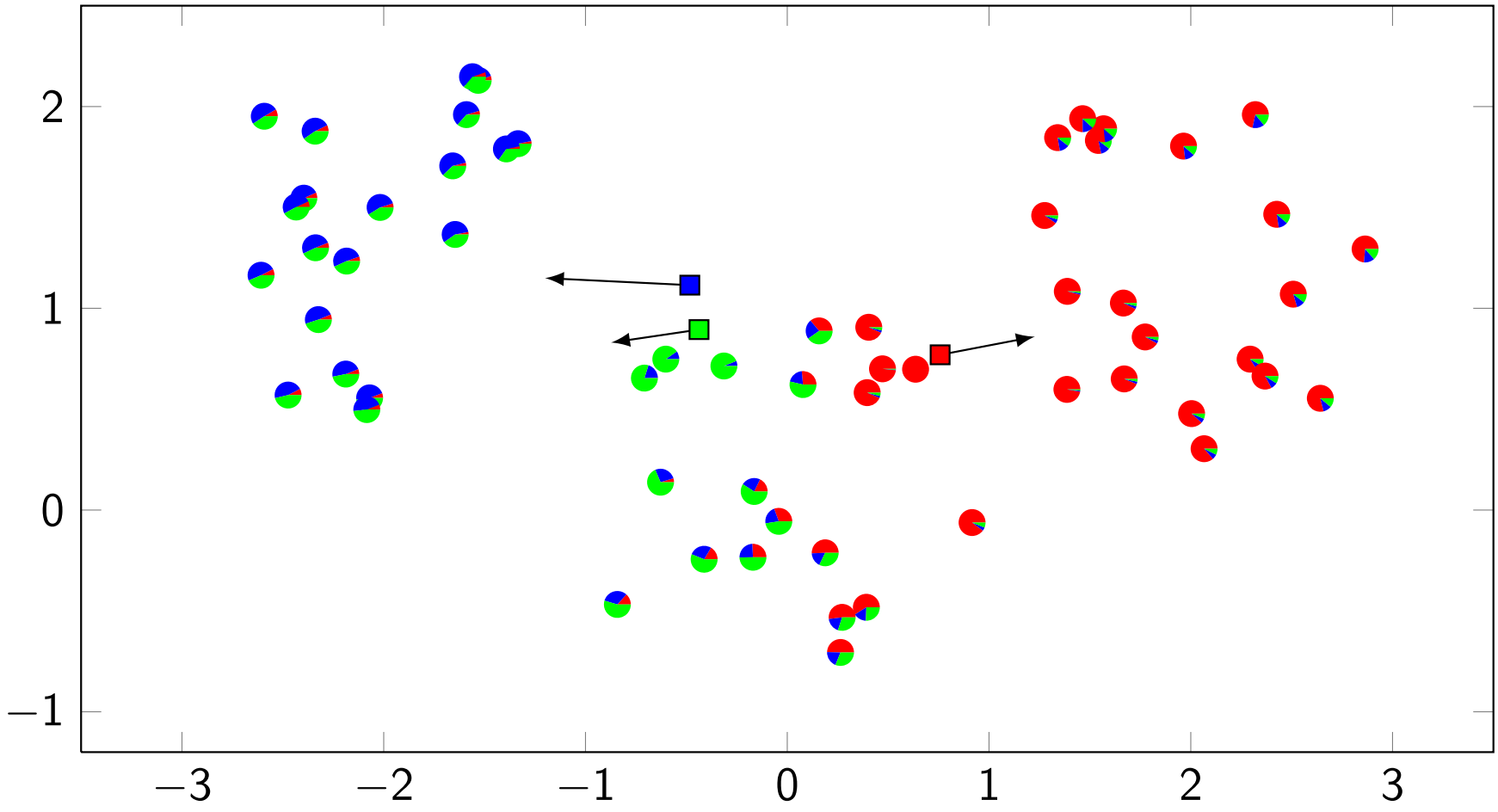
Algoritmus je ukončen, pokud platí

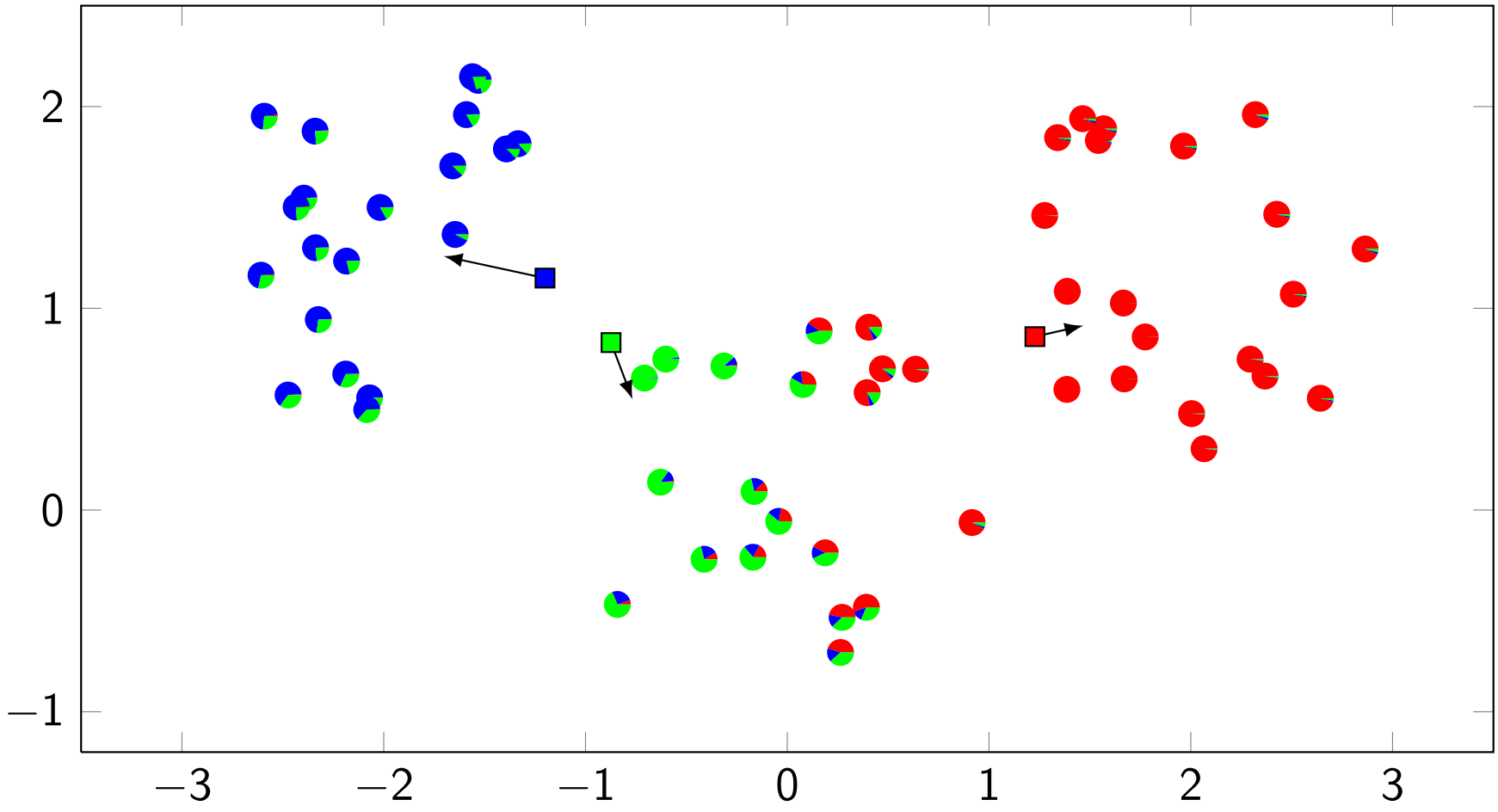
$$\max_{ik} |u_{ik}(\text{iter}) - u_{ik}(\text{iter}+1)| \leq \varepsilon$$

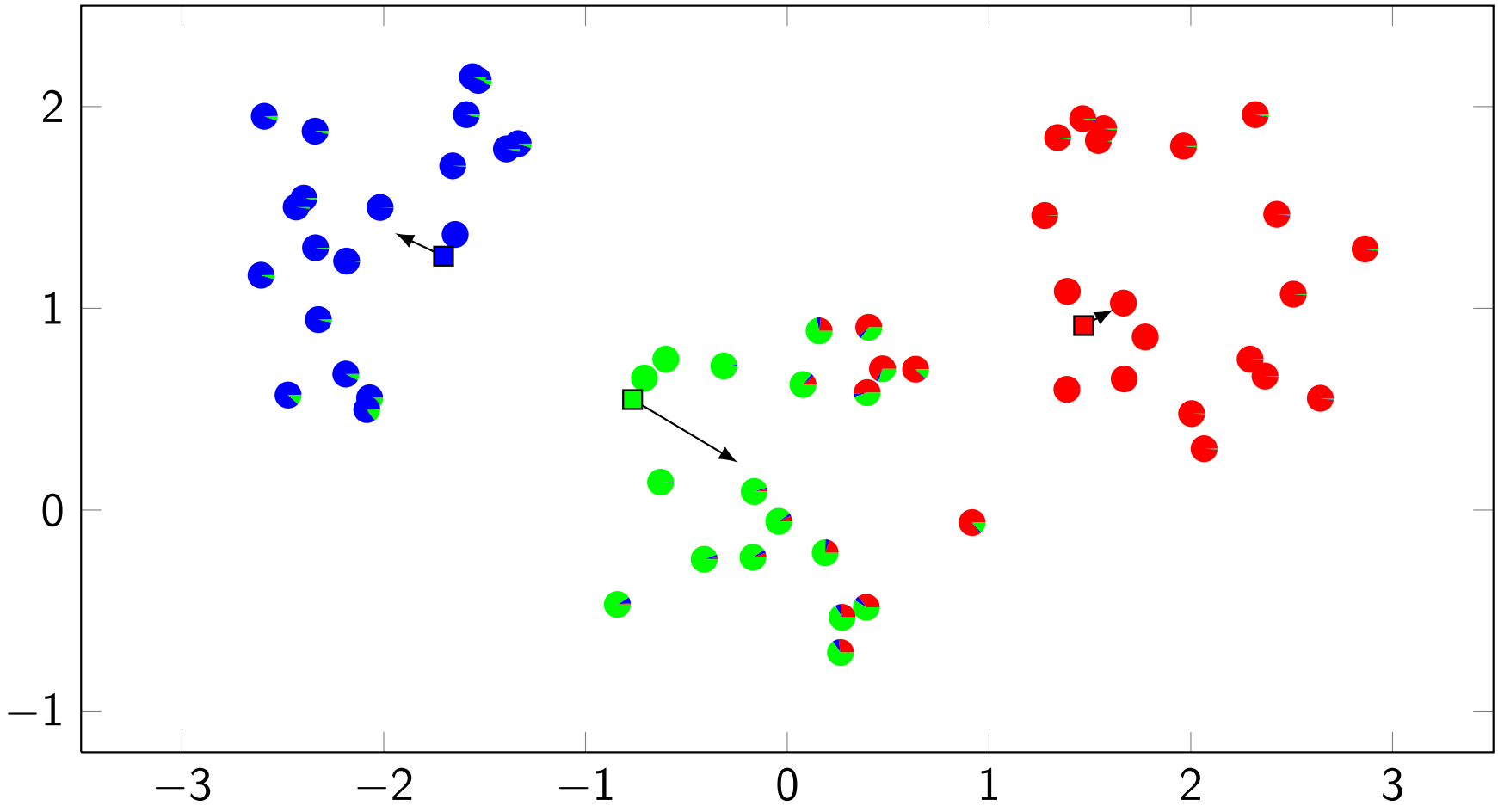
pro nějaké předem dané ε. (Tchebyshevova vzdálenost)

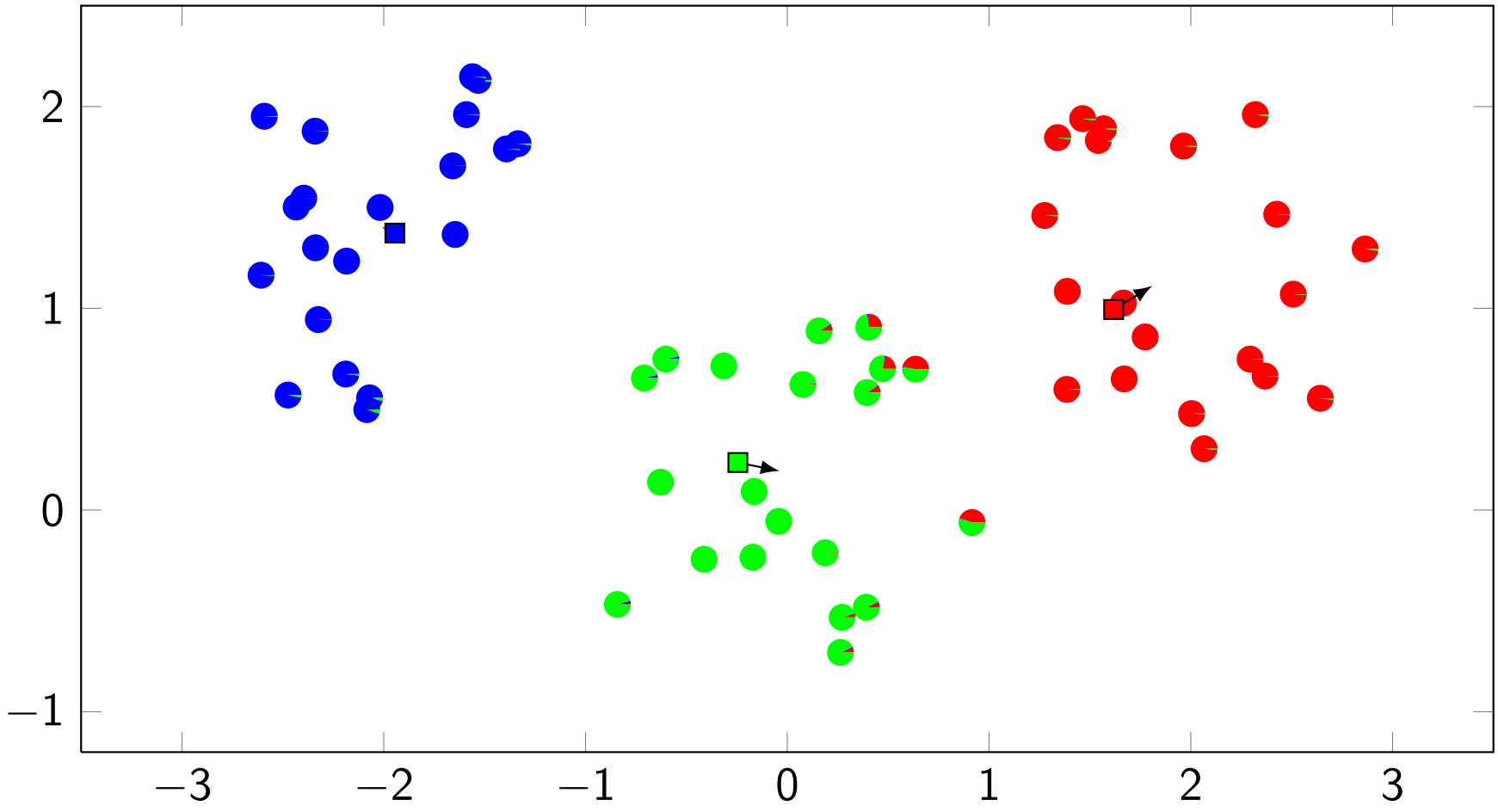
jinými slovy:
zastaví, když největší změna stupně příslušnosti ke shluku je $\leq \varepsilon$.

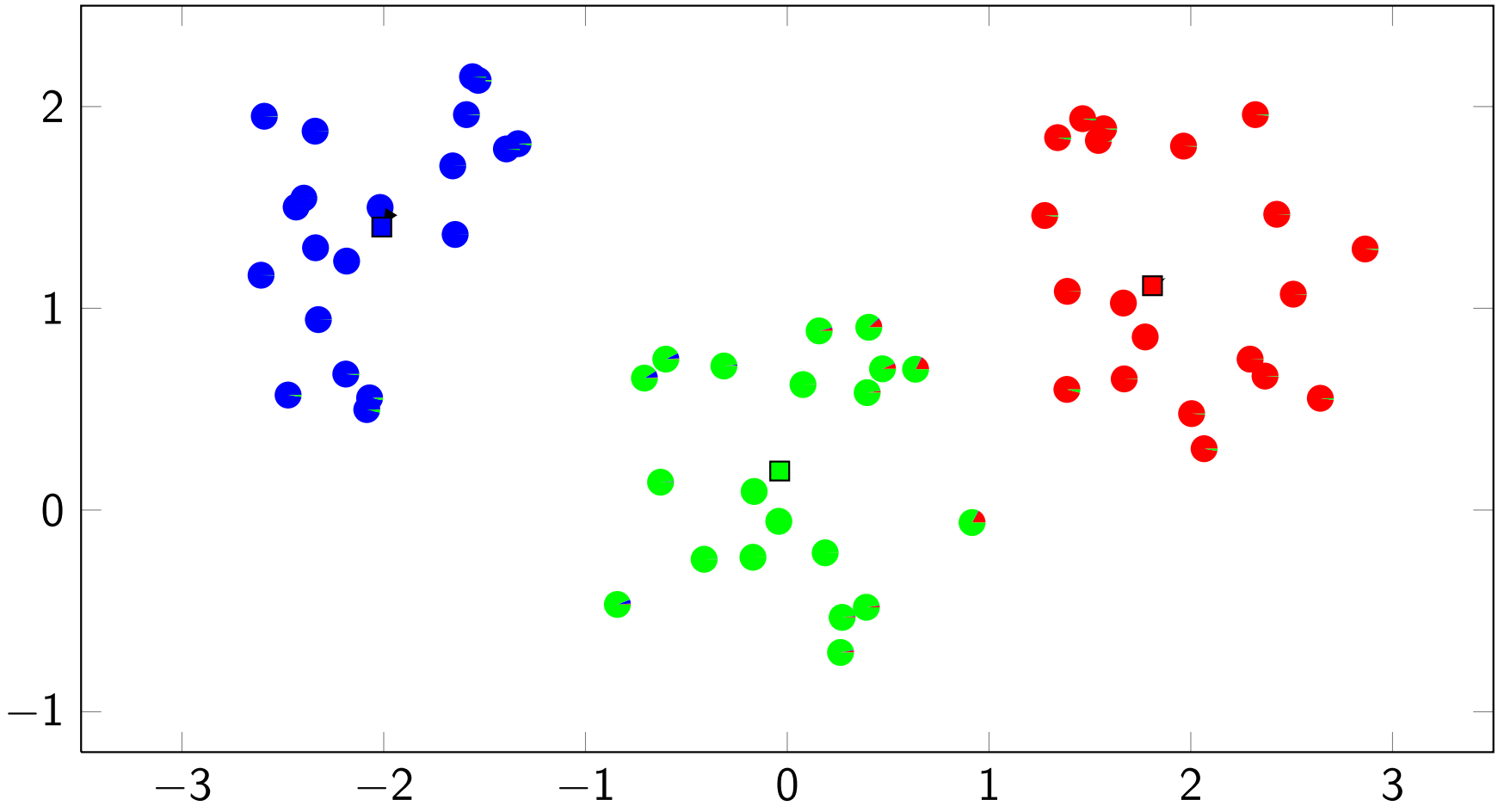

2
1
0
−1
−3
−2
−1
0
1
2
3


2
1
0
−1
−3
−2
−1
0
1
2
3


2
1
0
−1
−3
−2
−1
0
1
2
3


2
1
0
−1
−3
−2
−1
0
1
2
3


2
1
0
−1
−3
−2
−1
0
1
2
3


2
1
0
−1
−3
−2
−1
0
1
2
3

Shlukování založené na hustotě

Metody, které počítají s formací shlukování na základě hustoty datapointů.

Jednoduché pozorování: pokud jsou v nějaké oblasti data blízko sebe, je jejich hustota velká a tedy tvoří shluk.
Na druhou stranu, pokud nějaká data v oblasti s nízkou hustotou, jedná se pravděpodobně o odlehlé body a o šum.

Tyto algoritmy obvykle naleznou shluky jakýchkoli tvarů, a mají nízkou složitost $\mathcal{O}(N^2)$

Typické algoritmy jsou

- **DBSCAN** (**D**ensity-**B**ased **S**patial **C**lustering of **A**ppliactions with **N**oise)
- **DBCLASD** (**D**istribution-**B**ased **C**lustering of **LA**rge **S**patial **D**atabases)
- **DENCLUE** (**DEN**sity-based **CLU**st**E**ring)

Tyto algoritmy maji stejnou filosofii, ale liší se způsobem, kterým je hustota kvantifikována.

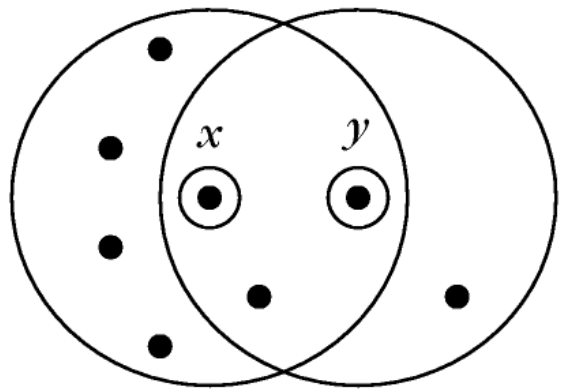
DBSCAN (**D**ensity-**B**ased **S**patial **C**lustering of **A**ppliactions with **N**oise)
ε-sousedství datapointu $\vec{x}$ v X je množina datapontů $\in X$ t.ž.

$$\mathrm{d}(\vec{x}, \vec{x}) \leq \epsilon.$$

Značíme $V_\varepsilon(\vec{x})$. Dále: $N_\varepsilon(\vec{x}) = |V_\varepsilon(\vec{x})|$

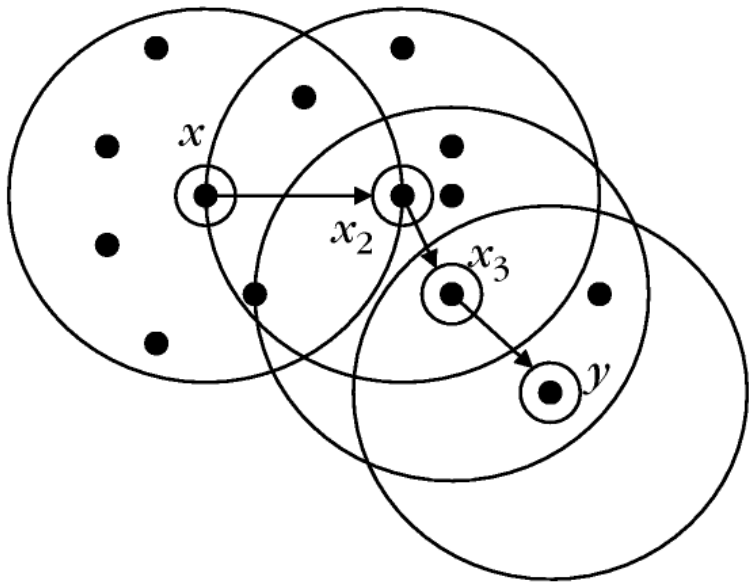
$\vec{y}$ je *přímo hustotně dosažitelný* z bodu $\vec{x}$ pokud

- $\vec{y} \in V_\varepsilon(\vec{x})$,
- $N_\varepsilon(\vec{x}) \geq q$.

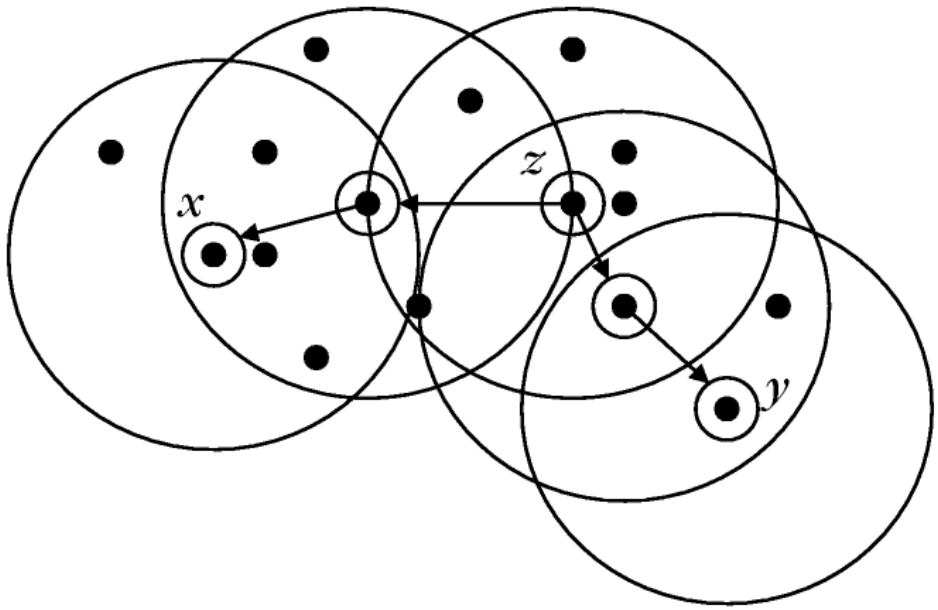


Zde $\vec{y}$ je přímo hustotně dosažitelný z $\vec{x}$,
ale $\vec{x}$ není přímo hustotně dosažitelný z $\vec{y}$ (pro $q = 5$).

$\vec{y}$ je *hustotně dosažitelný* z bodu $\vec{x}$, pokud existuje sekvence bodů $\vec{x}_1, \vec{x}_2, \ldots, \vec{x}_p \in X$, t.ž. $\vec{x}_1 = \vec{x}$ a $\vec{x}_p = \vec{y}$, a $\vec{x}_{i+1}$ je přímo hustotně dosažitelný z $\vec{x}_o$



$\vec{x}$ je *hustotně spojený* s bodem $\vec{y}$,
pokud existuje $\vec{z} \in X$ t.ž. $\vec{x}$ a $\vec{y}$ jsou hustotně dosažitelné z $\vec{z}$.



Shluk C je (v DBSCAN) definový jako neprázdná podmnožina X splňující

- Pokud $\vec{x}$ patří do C a y je hustotně dosažitelný z $\vec{x}$, pak $\vec{y} \in C$.
- Pro každý pár $(\vec{x}, \vec{y}) \in C$, $\vec{x}$ a $\vec{y}$ jsou hustotně spojené.

Nechť $C_1, C_2, \ldots, C_m$ jsou shluky v X. Datapointy, které nejsou obsaženy v žádném ze shluků $C_1, C_2, \ldots, C_m$ jsou nazývány *šum*.

$\vec{x}$ se nazýva *jádrový bod*, pokud má nejméně q bodů v sousedství.
Jinak se nazýva *nejádrový bod*.

Nejádrový bod je buďto *okrajový bod*, nebo *šumový bod*
(podle toho, jestli je dostažitelný z nějakého jádrového bodu).

Tvrzení: Pokud $\vec{x}$ je jádrový bod, D je množina bodů v X, která je hustotně dosažitelná z $\vec{x}$, pak D je shluk.

Tvrzení: Pokud C je shluk a $\vec{x}$ je jádrový bod v C, pak C je rovna množině bodů v X, které jsou hustotně dosažitelné z $\vec{x}$.

Tato dvě tvrzení implikují, že:
shluk je unikátně dán jakýmkoli ze svých jádrových bodů.

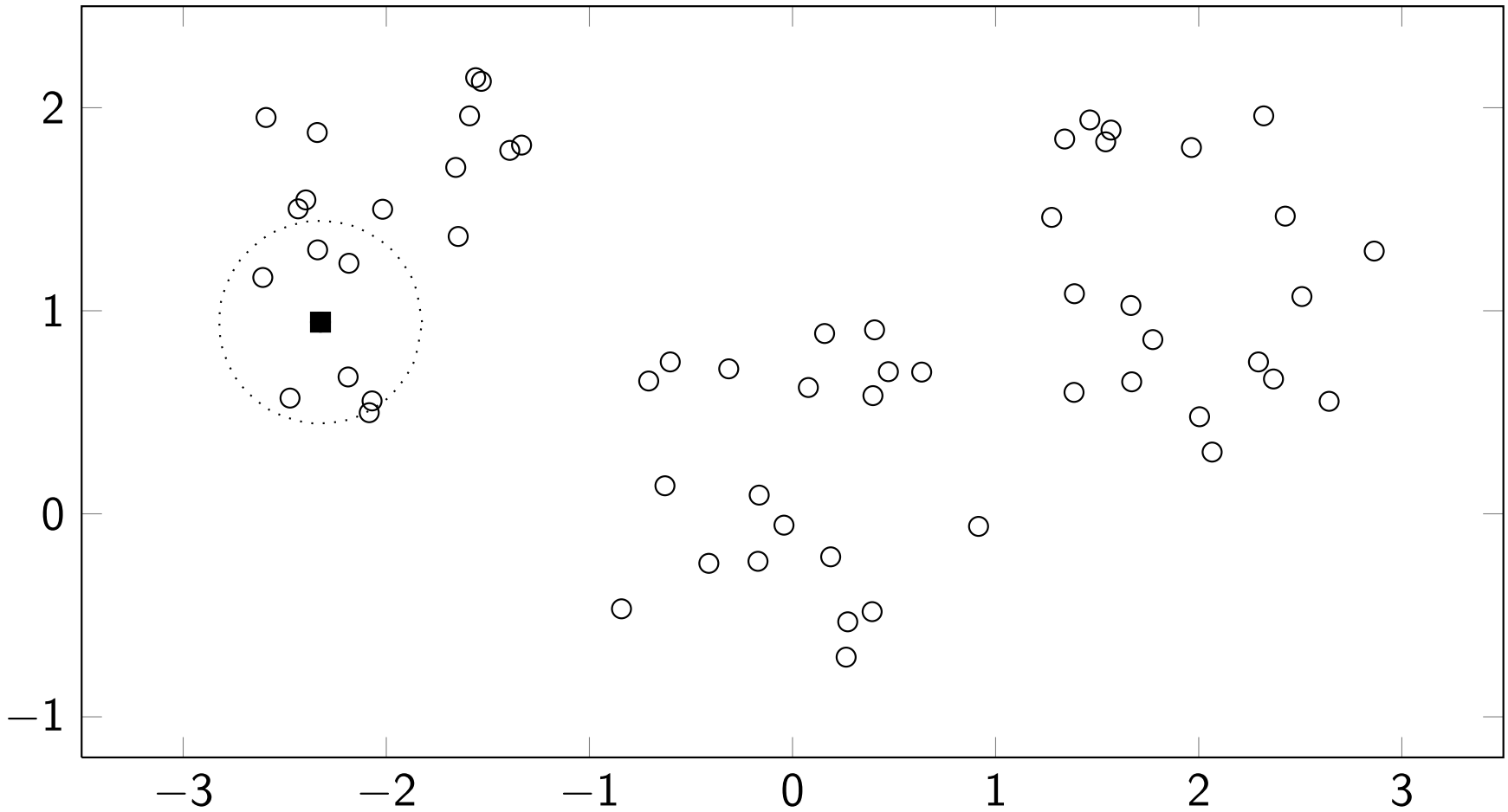
Algoritmus DBSCAN

Inicializuj $X_{\text{un}} = X$, $m = 0$.

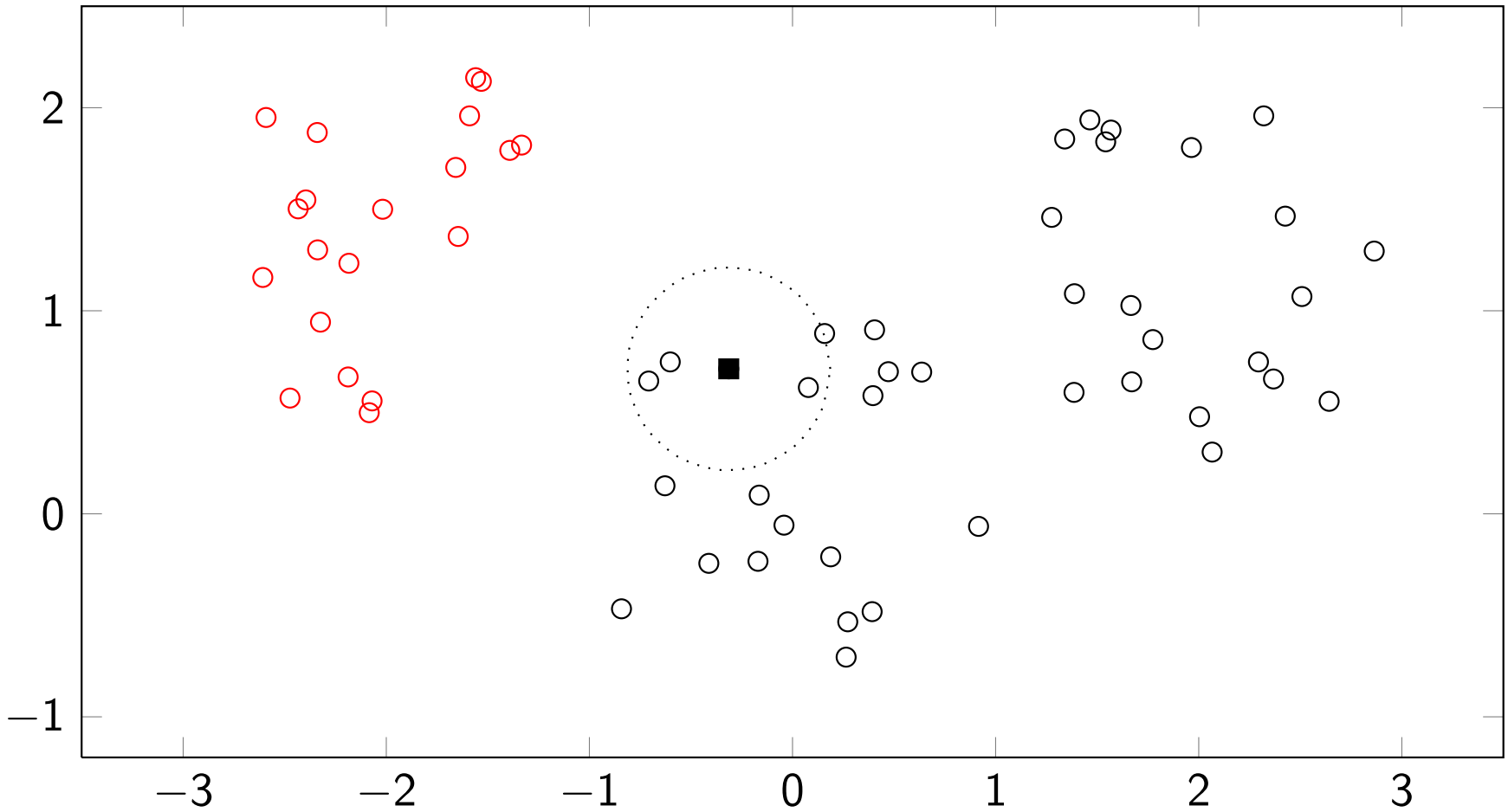
Dokud $X_{\text{un}} \neq \emptyset$ prováděj:

- Libovolně vyber $\vec{x} \in X_{\text{un}}$
- Pokud $\vec{x}$ je nejádrový, označ ho jako šum, a $X_{\text{un}} = X_{\text{un}} \setminus \{\vec{x}\}$
- Pokud $\vec{x}$ je jádrový, $m = m + 1$, najdi všechny hustotně dosažitelné body v X bodu $\vec{x}$.
  Všechny tyto body a $\vec{x}$ tvoří shluk C_m.
  (okrajové body, které předtím byly označeny za šum jsou také přiřazeny do C_m)
  $X_{\text{un}} = X_{\text{un}} \setminus C_m$.

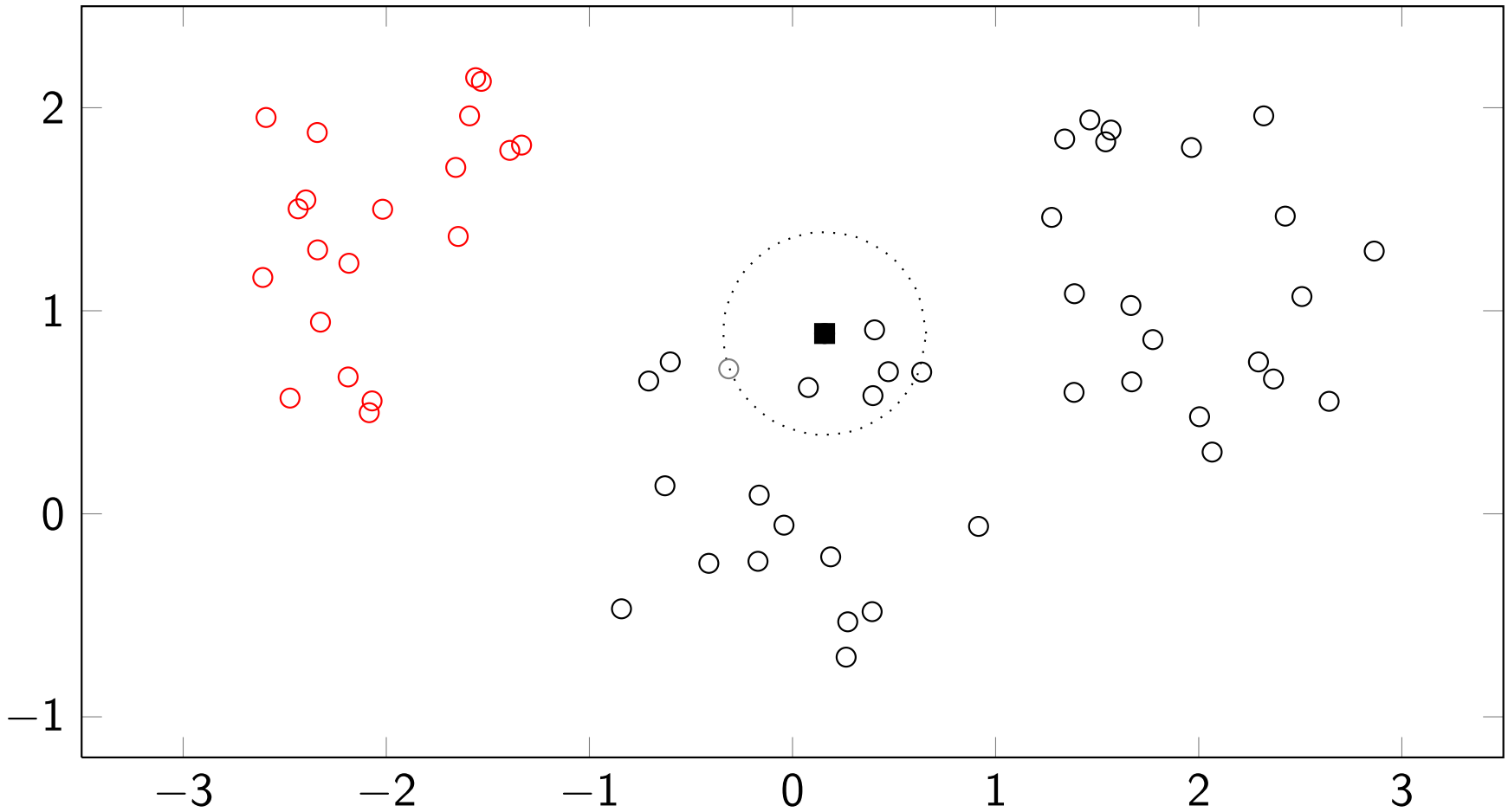
Demonstrace algoritmu DBSCAN



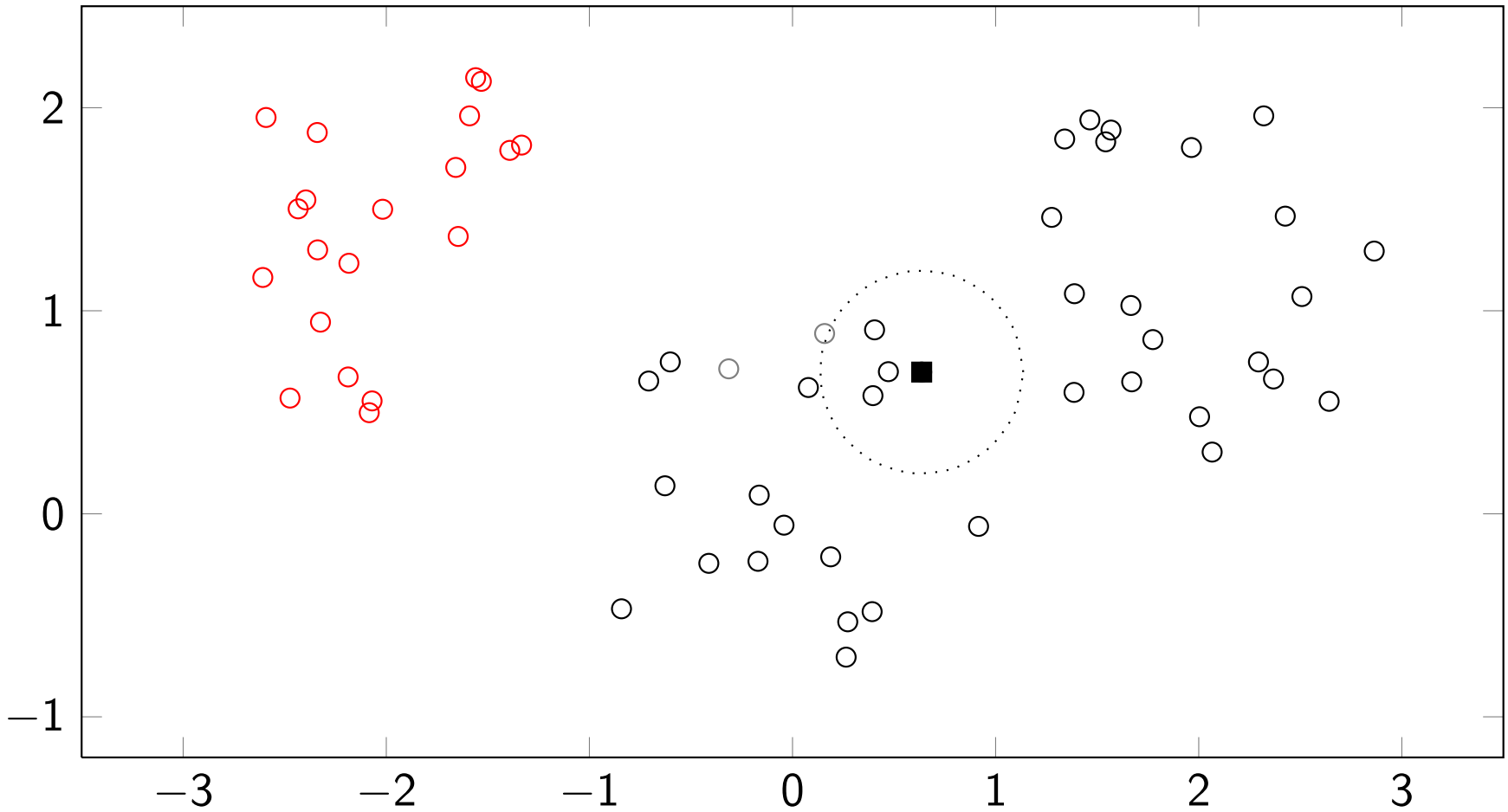
Demonstrace algoritmu DBSCAN



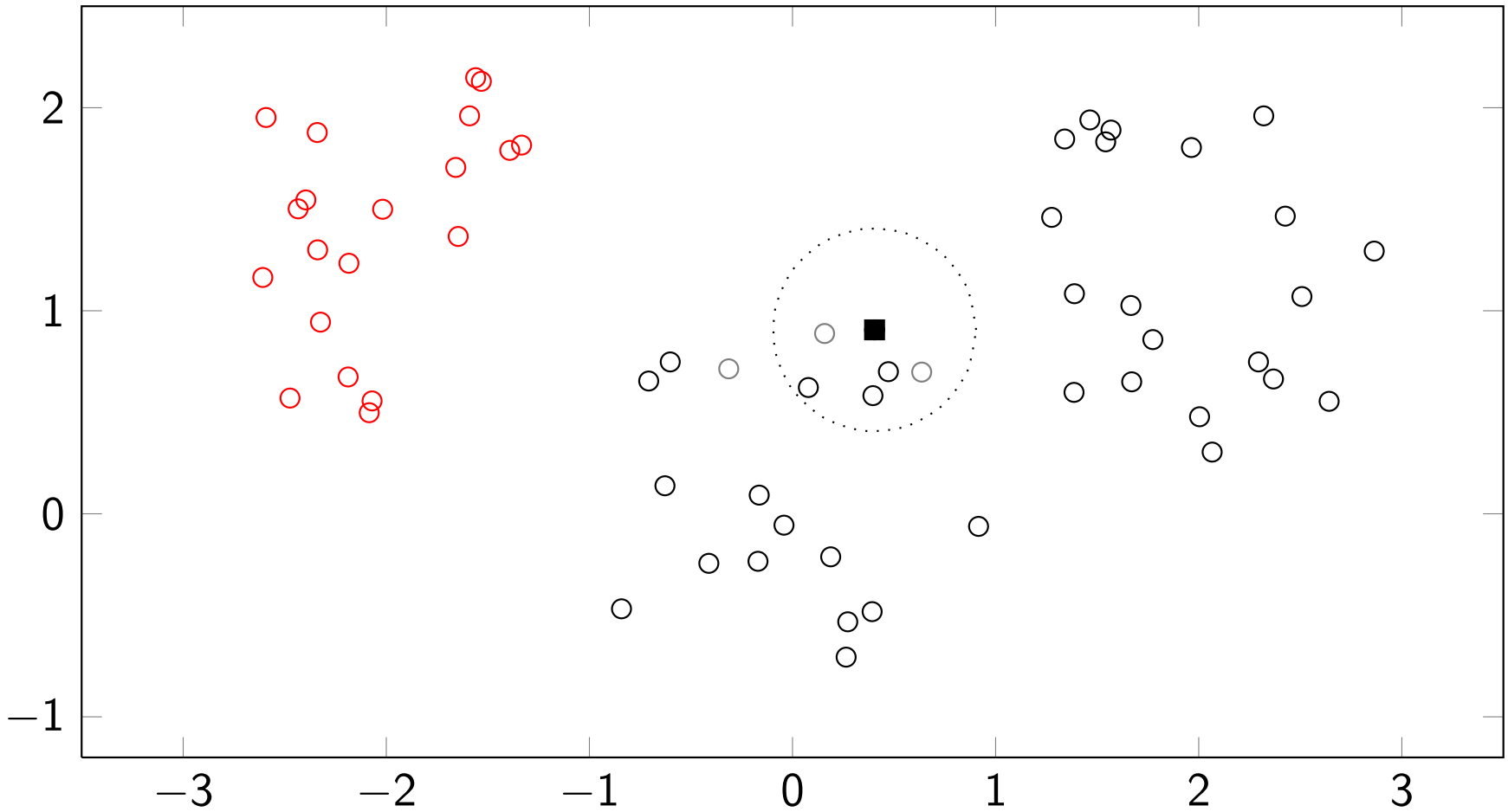
Demonstrace algoritmu DBSCAN



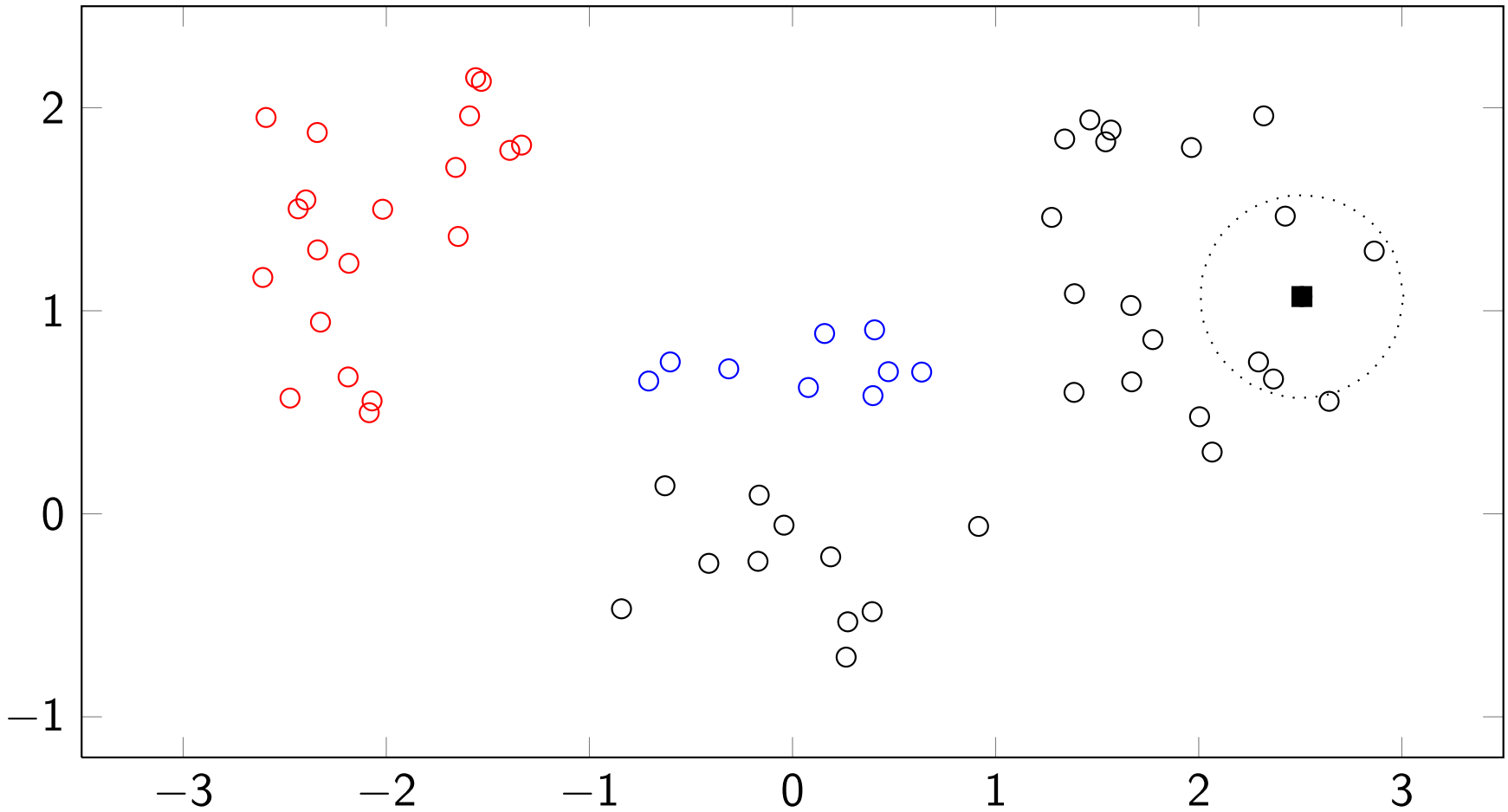
Demonstrace algoritmu DBSCAN 4



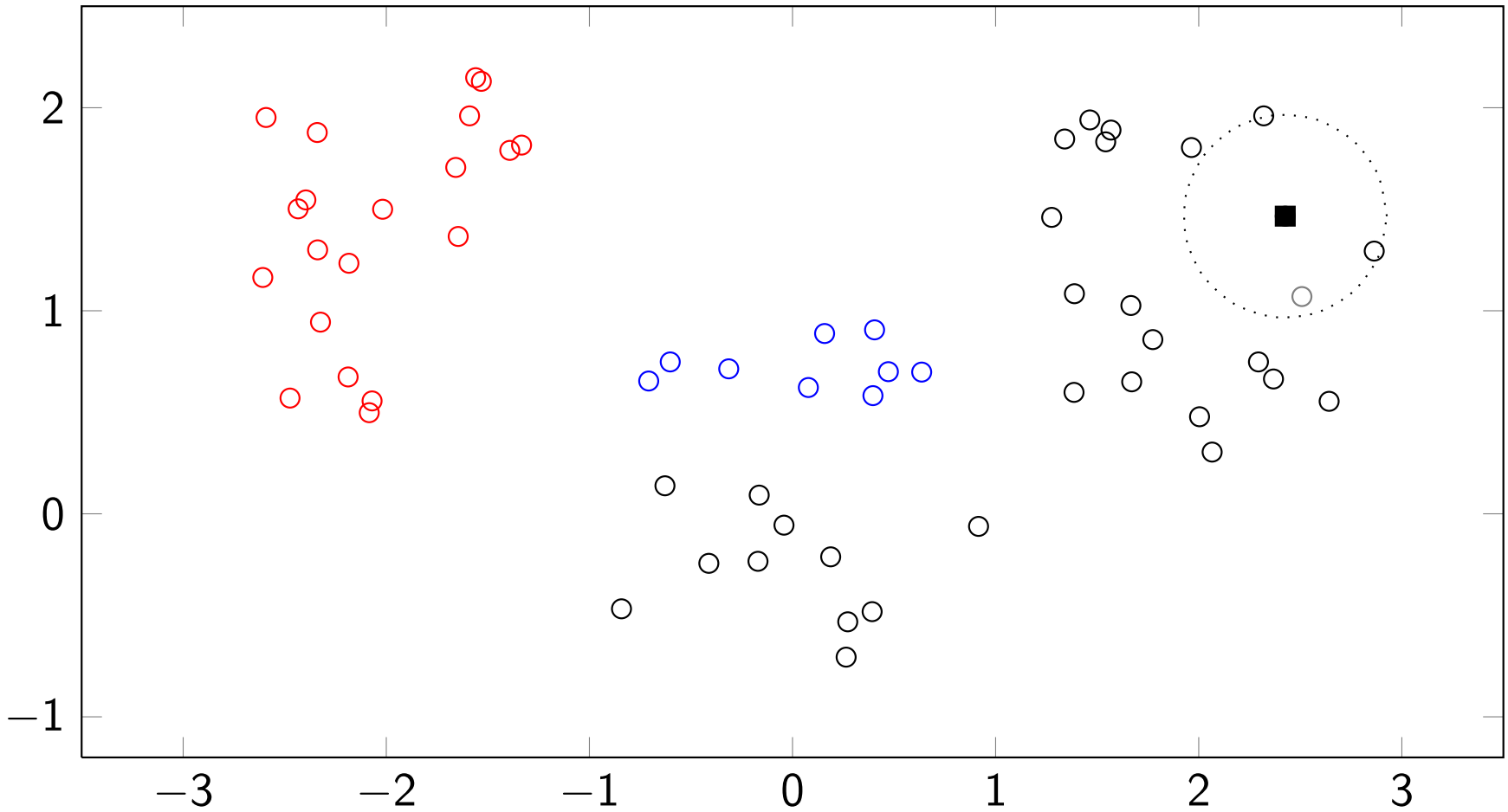
Demonstrace algoritmu DBSCAN



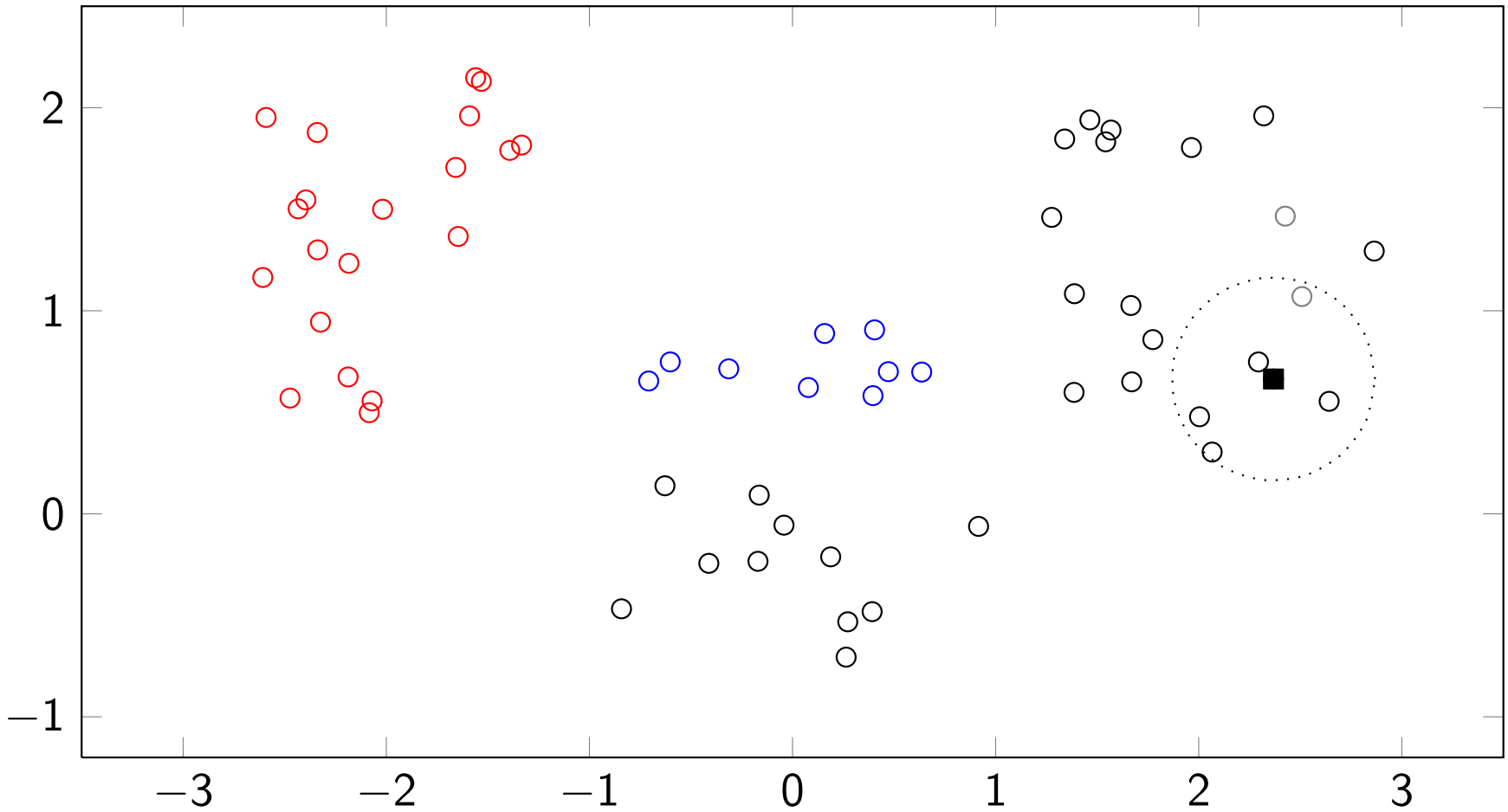
Demonstrace algoritmu DBSCAN



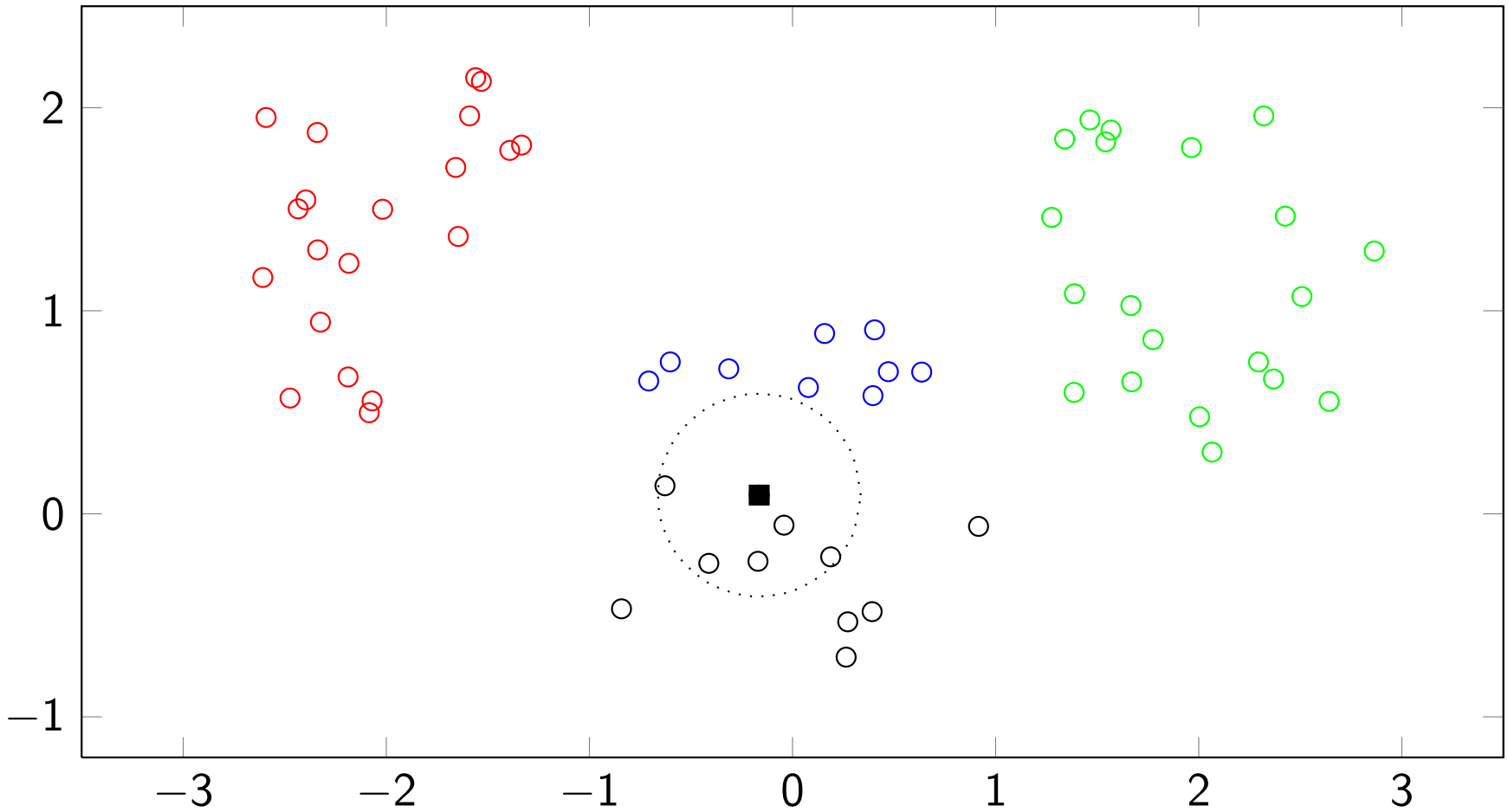
Demonstrace algoritmu DBSCAN



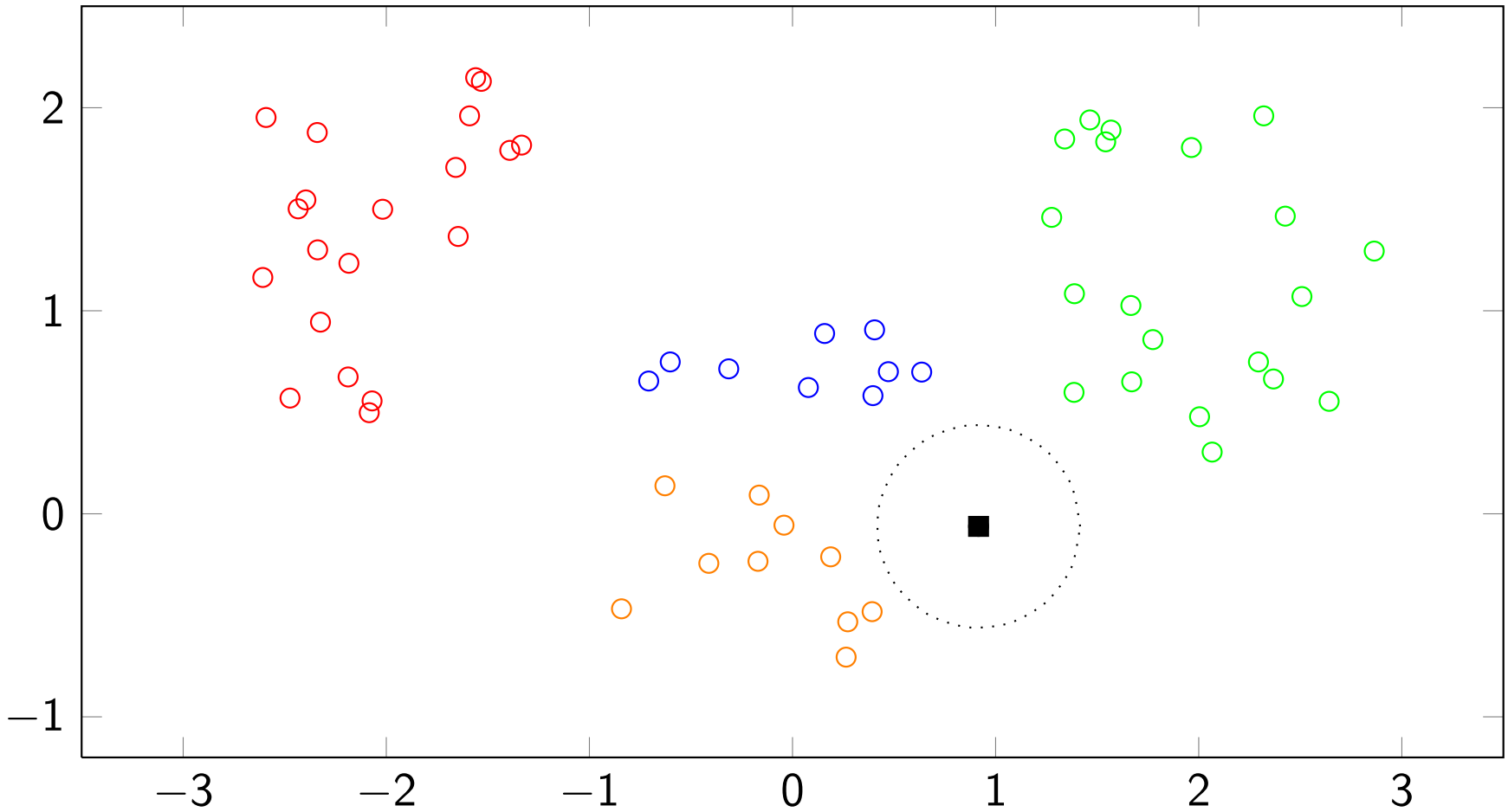
Demonstrace algoritmu DBSCAN



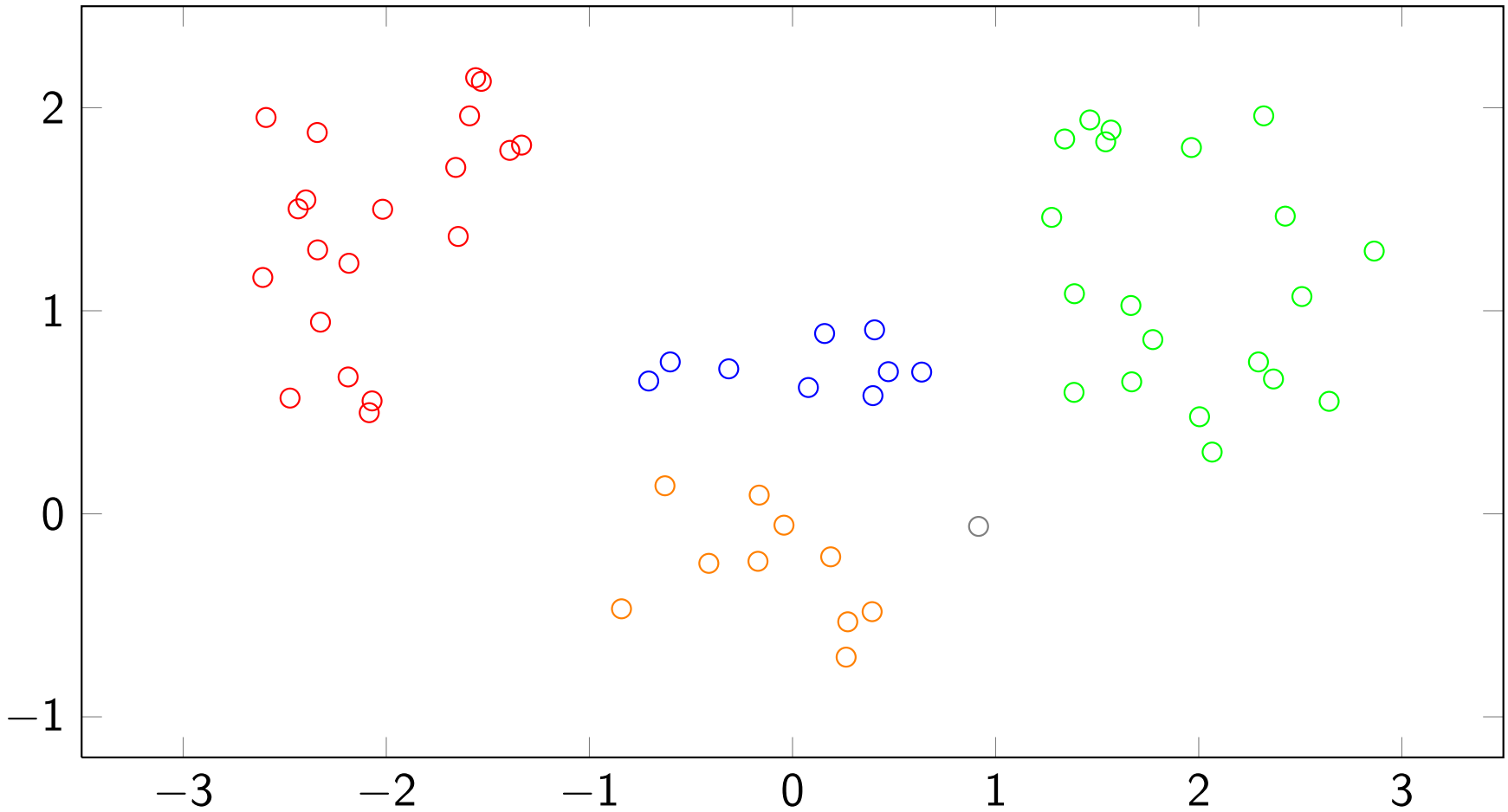
Demonstrace algoritmu DBSCAN



Demonstrace algoritmu DBSCAN



Demonstrace algoritmu DBSCAN

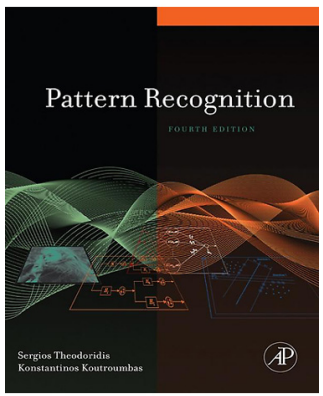


Pozn.:

- Velmi závisí na volbě parametrů q a ε.
- Implementováno-li pomocí R*-stromů, lze dosáhnout složitosti $\mathcal{O}(N \log N)$.

KMI/ALS1 – Algoritmy a složitost 1

Chcete vědět víc?



Theodoridis S., Koutroumbas K.,
Pattern Recognition,
Academic Press, 2008

Sekvenční shlukovací algoritmy

Tyto algoritmy jsou nízké složitosti ($\mathcal{O}(N)$).
Výsledek záleží na tom, v jakém pořadí jsou datapointy algoritmu předávány a na uživatelem definovaných parametrech.

BSAS – **B**asic **S**equential **A**lgorithmic **S**cheme

Nechť $\mathrm{d}(\vec{x}, C)$ označuje vzdálenost mezi vektorem $\vec{x}$ a shlukem C.

Uživatelem definované parametry:

- Θ – práh nepodobnosti (vzdálenosti),
- q – maximální počet shluků.

Datapointy jsou předkládány algoritmu postupně (sekvenčně), každý z nich je přiřazen existujícímu shluku, nebo nově vytvořenému shluku (vzávislosti na už existujících shlucích).

BSAS – Basic Sequential Algorithmic Scheme
Inicializuj $m = 1, C_m = \{\vec{x}_1\}$.

Pro $i = 2$ až N

- Najdi $C_k : \mathrm{d}(\vec{x}_i, C_k) = \min\limits_{i \le j \le m} \mathrm{d}(\vec{x}_i, C_j)$.
- Pokud $\mathrm{d}(\vec{x}_i, C_k) > \Theta$ a $(m < q)$:
  $m = m + 1$
  $C_m = \{\vec{x}_i\}$
- Jinak
  $C_k = C_k \cup \{\vec{x}_i\}$
  Tam, kde je to potřeba, uprav reprezentanty.

Různými volbami $\mathrm{d}(\vec{x}_i, C)$ dostáváme různé algoritmy

Pokud C je reprezentován jedním vektorem, z $\mathrm{d}(\vec{x}, C)$ se stane

$$\mathrm{d}(\vec{x}, C) = \mathrm{d}(\vec{x}, \vec{r}_c),$$

kde $\vec{r}_c$ je reprezentant C

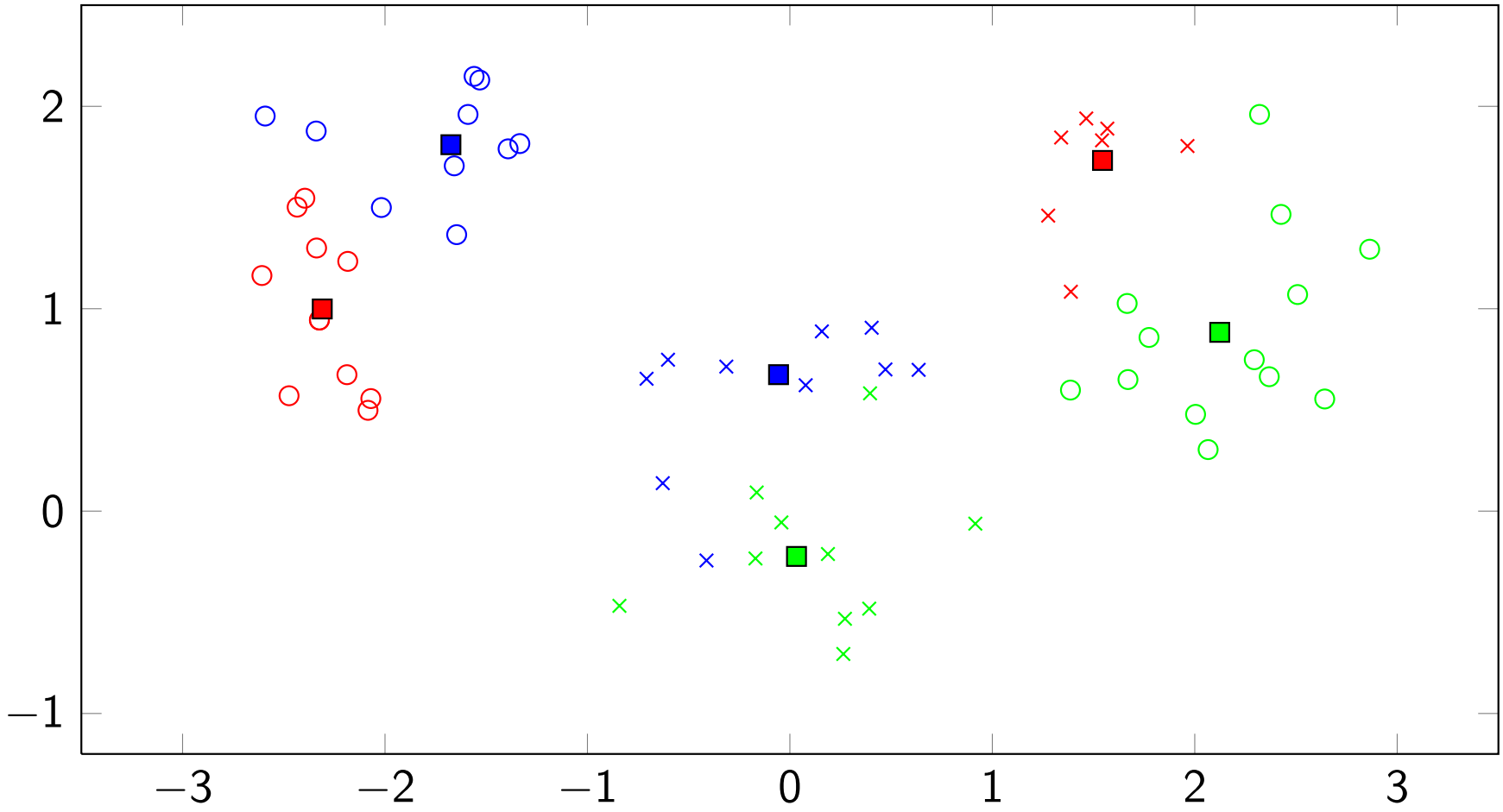
Úprava reprezentanta může být třeba:

$$\vec{r}_{C_k}^{\text{new}} = \frac{(n_{C_k}^{\text{new}} - 1)\vec{r}_{C_k}^{\text{old}} + \vec{x}}{n_{C_k}^{\text{new}}}$$

kde

- $n_{C_k}^{\text{new}}$ je kardinalita shluku C_k po přiřazení $\vec{x}$,
- $\vec{r}_{C_k}^{\text{new}}$ je reprezentant C_k po přiřazení $\vec{x}$,
- $\vec{r}_{C_k}^{\text{old}}$ je reprezentant C_k před přiřazením $\vec{x}$.

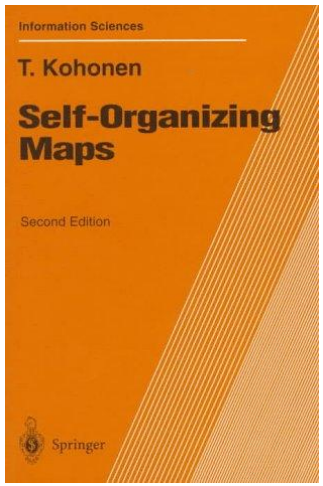
Demonstrace BSAS ($q = 6$, $\Theta = 1.2$)



samoorganizační se sítě (SOM)

Přístup založený na NN; nepotřebuje předem znát počet shluků.

KMI/UNS – Umělé neuronové sítě



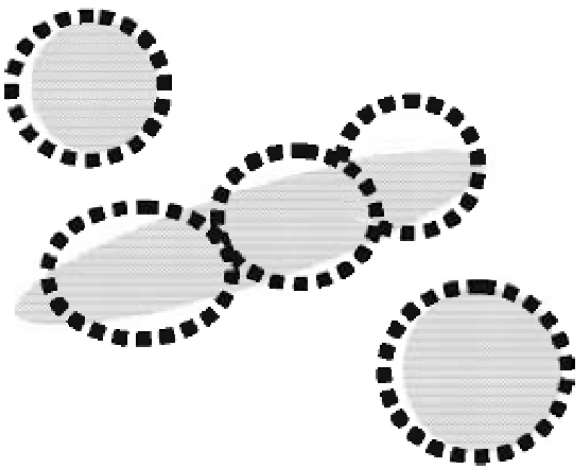
Kohonen T.,
Self-Organizing Maps,
Springer., 2001

Platnost shlukování

Vyjadřují vygenerované shluky pravou podstadu dat?

Přestože jde o unsupervised metoty, jsou v (téměř) každém algoritmu dvě komponenty, které jsou v jádru supervised:
(A) *Měření podobnosti/vzdálenosti*

- to určuje tvar shluků ,
- např. Euklidovská vzdálenost při k-means povede ke kulatým shlukům,



(B) *Počet shluků*

Normálně by uživatel odhadl rozsah počtu shluků a pak použil *měření platnosti shlukování*, aby zjistil, který počet nejlépe vyjadřuje strukturu dat.

Existuje mnoho měření, které se nazývají *indexy platonosti shluku* Jejich hodnoty se vztahují k počtu shluků, Jsou používány k posouzení shluků nalezených v datech a ohodnocení kvality struktury, ktera byla takto nalezena.

Dva přirozené požadavky na shluky jsou:

Kompaktnost – jak blízko jsou prvky ve shluku.

Separabilita – jak vzájemně odlišné shluky jsou.

Davies-Bouldinův index

Mějme

$$s_i = \frac{1}{N_i} \sum_{\vec{x} \in C_i} \|\vec{x} - \vec{v}_i\|^2,$$

kde

- $\vec{v}_i$ je reprezentant (např. centroid) shluku,
- N_i je kardinalita shluku,

a mějme vzálenost mezi dvěma reprezentanty

$$d_{ij} = \|\vec{v}_i - \vec{v}_j\|^2$$

Nyní můžeme definovat poměr

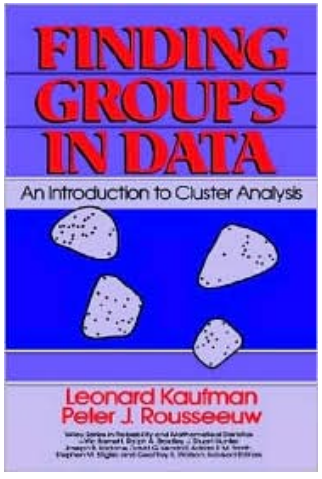
$$r_i = \max_{j, j \neq i} \frac{s_i + s_j}{d_{ij}}$$

a průměr jeho hodnot:

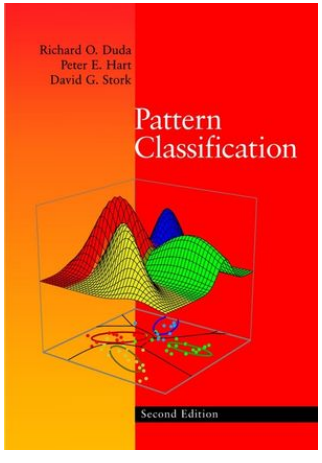
$$r = \frac{1}{c} \sum_{i=1}^{c} r_i$$

„Optimální“ počet c shluků je ten, kde je r minimální.

Chcete vědět víc?



Kaufman L., Rousseeuw P. J.,
Finding Groups in Data: An Introduction to Cluster Analysis,
Wiley & Sons, Inc., 1990



Duda R. O., Hart P. E., Stork D. G.,
Pattern Classification,
Wiley, 2000