

April 13, Kyiv

Data Science & Mathematical Modeling Bachelor Program

Course “Basics of Machine Learning”

Lecture 5: Distance-Based Classification & Regression. Clustering



Oleg CHERTOV

Professor, Sc.D. (Doctor Habilitatus),
Head of the Applied Mathematics Department



Applied Mathematics Department
Igor Sikorsky Kyiv Polytechnic Institute
Ukraine



Регресія

- Строгое математичне визначення регресії – це умовне математичне сподівання однієї випадкової величини відносно іншої
- Регресія – розпізнавання чисельної (скалярної або векторної) характеристики об'єкта, наприклад, прогнозування вартості будинку чи курсу долара США (скалярна величина) або параметрів завтрашньої погоди: середня температура вдень і вночі, сила віtru, наявність опадів тощо (векторна величина)
- В багатьох прикладних задачах (розвізнавальна) регресія є регресією математичною

Класифікація

- Класифікація – розпізнавання якісної (дискретної) характеристики об'єкта
- Клас – це множина об'єктів, для яких ця характеристика приймає певне (визначене) значення
- Відповіддю класифікатора для кожного об'єкта краще вважати не номер класу, до якого класифікатор відносить об'єкт, а вектор «впевненості» (confidence) в приналежності об'єкта кожному з класів. Тим самим, класифікація перетворюється в спеціальний випадок регресії

Класифікація і регресія

- Вийшло, що регресія (розділення неперервних величин) ущемлена у порівнянні з класифікацією (розділенням дискретних величин): замість вектору (розподілу) маємо лише одне значення
- Рівноправність можна частково відновити, додавши до простору відповідей додаткові параметри розподілу
- Наприклад дисперсію в разі одновимірної регресії і матрицю коваріації у випадку багатовимірної

Гіпотези компактності та неперервності

Задачі класифікації та регресії:

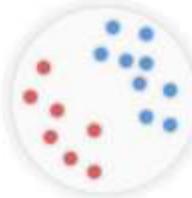
X — об'єкти, Y — відповіді

$X^\ell = (x_i, y_i)_{i=1}^\ell$ — навчальна вибірка

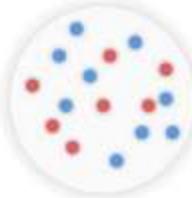
Гіпотеза компактності (для класифікації):

близькі об'єкти, зазвичай, належать одному класу

виконана:



не виконана:



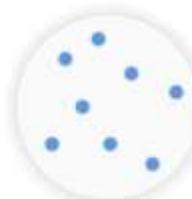
Гіпотеза неперервності (для регресії):

близьким об'єктам, зазвичай, відповідають близькі відповіді

виконана:



не виконана:

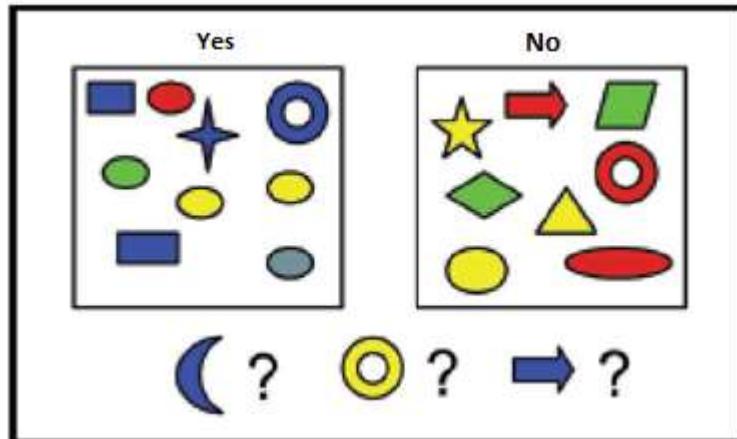


Distance-based vs similarity-based

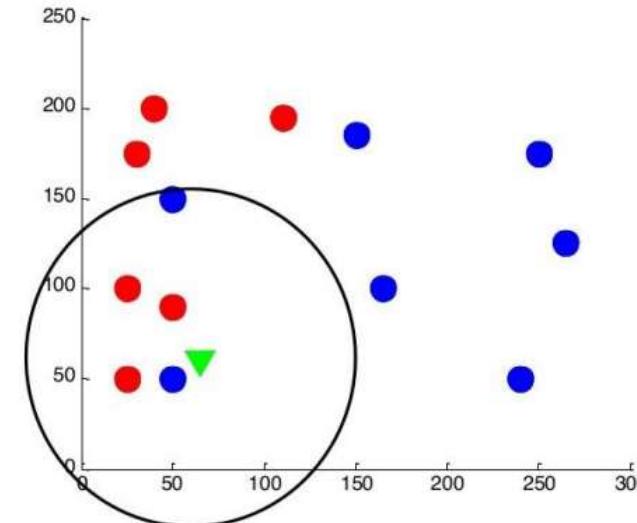
Схожість чи близькість

- Classification is done by relating the unknown to the known according to some distance/similarity function

similarity-based
(e.g. decision trees)



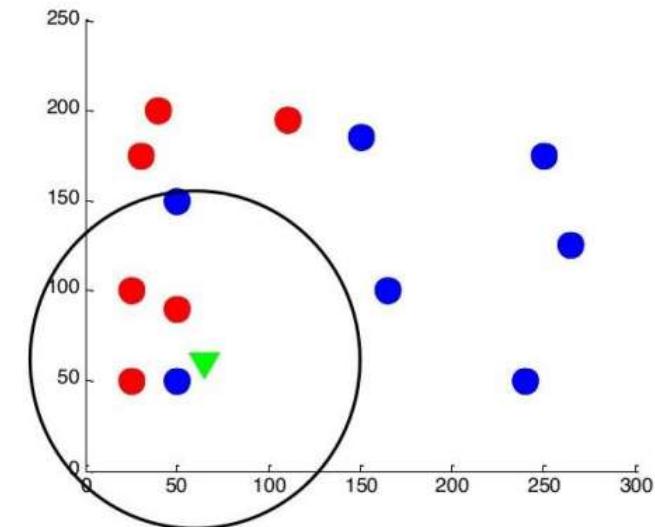
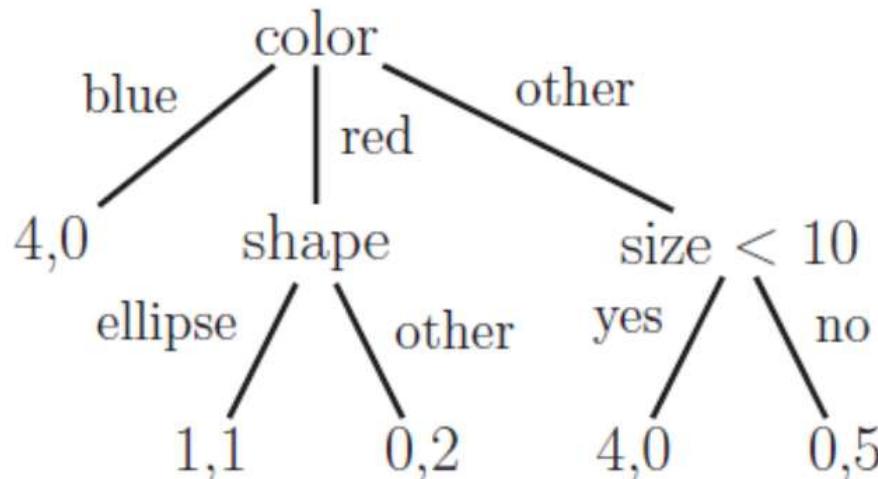
distance-based
(e.g. kNN)



Eager learners vs Lazy learners

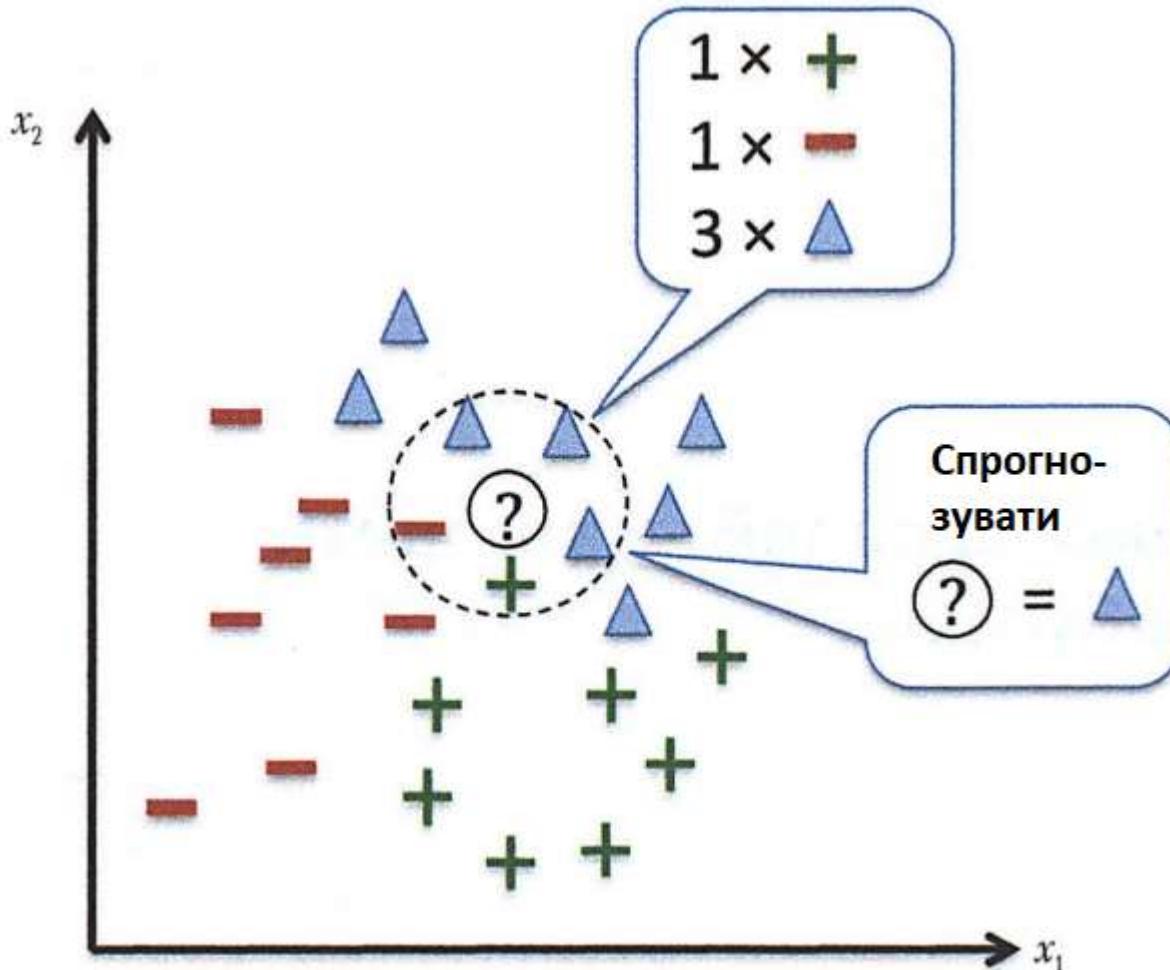
Жадібні / ледачі алгоритми

- **Eager learners**
(e.g. decision trees,
Bayesian classifiers, Neural networks)
- **Lazy learners**
(e.g. k-nearest neighbor,
k-means clustering)



- Different names for **Lazy learning**
 - ❖ Memory-based reasoning
 - ❖ Example-based reasoning
 - ❖ Instance-based reasoning
 - ❖ Case-based reasoning

Алгоритм k найближчих сусідів



1. Вибрати число k і метрику відстані
2. Знайти k найближчих сусідів зразка, який ми хочемо класифікувати
3. Присвоїти мітку класу мажоритарним голосуванням

Алгоритм k найближчих сусідів

Customer	Age	Income	No. credit cards	Class	Distance from John
George	35	35K	3	No	$\sqrt{[(35-37)^2 + (35-50)^2 + (3-2)^2]} = 15.16$
Rachel	22	50K	2	Yes	$\sqrt{[(22-37)^2 + (50-50)^2 + (2-2)^2]} = 15$
Steve	63	200K	1	No	$\sqrt{[(63-37)^2 + (200-50)^2 + (1-2)^2]} = 152.23$
Tom	59	170K	1	No	$\sqrt{[(59-37)^2 + (170-50)^2 + (1-2)^2]} = 122$
Anne	25	40K	4	Yes	$\sqrt{[(25-37)^2 + (40-50)^2 + (4-2)^2]} = 15.74$
John	37	50K	2	YES	

$k = 1$, може вплинути шум (15 і 15,16)

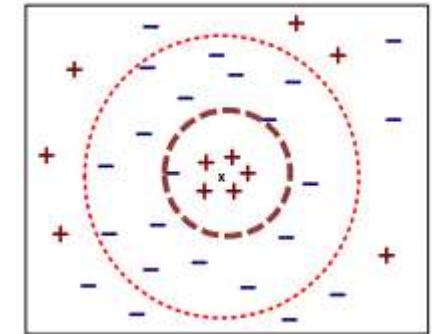
$k = 5$, викривлення, бо «захоплюється» багато «сусідів»

Яке значення k краще?

- Розпізнавання методом $k = 1$ найближчого сусіда не робить жодної помилки на пред'явленому йому вхідному наборі даних (на навчальній вибірці), але може помилятися на невідомих йому векторах ознак
- Розпізнавання методом $k > 1$ найближчих сусідів не обов'язково безпомилково розпізнає точки навчальної вибірки, зате при невеликих k , зазвичай, менше помиляється на невідомих йому векторах
- Перевага методу kNN – **проста інтерпретація** – case-based reasoning, схоже на якийсь відомий випадок

How to choose k ?

- If k is too small it is sensitive to noise points
- Larger k works well. But too large k may include majority points from other classes
- На практиці хороші результати для більшості наборів даних з малою кількістю розмірностей можна отримати за допомогою значення k , що знаходитьться приблизно між 5 і 10
- Значення для k можна також вибрати шляхом перехресної перевірки (cross-validation)



Stuart J. Russell and Peter Norvig.
Artificial Intelligence: A Modern
Approach, 2020 4th ed.

It is used in over 1400 universities
worldwide.

«Ледаче» машинне навчання

- Навчання класифікатора (чи розпізнавача) при методі k найближчих сусідів є тривіальним і зводиться до запам'ятовування навчальної вибірки
- Розпізнавання теж тривіальне, але є дуже ресурсозатратним, і трудомісткість зростає пропорційно обсягу навчальної вибірки
- Погано: на навчальній виборці – швидко, на тестовій виборці – повільно!
- Що робити – розберемо окремо!

Алгоритм k найближчих сусідів

- Для класифікації:

1. Вибрати число k і метрику відстані
2. Знайти k найближчих сусідів зразка, який ми хочемо класифікувати
3. Присвоїти мітку класу мажоритарним голосуванням

- Для регресії:

1. Вибрати число k і метрику відстані
2. Знайти k найближчих сусідів зразка, для якого ми хочемо знайти значення
3. Присвоїти їх середньоарифметичне значення

Алгоритм k найближчих сусідів (формальне визначення)

- Нехай маємо навчальну вибірку $X = (x_i, y_i), i = 1..l$ і функцію відстані ρ
- Потрібно класифікувати новий об'єкт u
- Розташуємо об'єкти навчальної вибірки X у порядку зростання відстані до нового об'єкта u :

$$\rho(u, x_{(1), u}) \leq \rho(u, x_{(2), u}) \leq \dots \leq \rho(u, x_{(l), u}),$$

де через $x_{(i), u}$ позначений i -ий сусіда об'єкта u

- Алгоритм KNN віднесе об'єкт u до того класу, представників котрого виявиться найбільше серед k його найближчих сусідів:

$$a(u, X, k) = \arg \max_y \sum_{i=1}^k w_{(i)} \left[y_{(i), u} = y \right]$$

□ *Оцінка близькості
об'єкта u до i -го сусіда*

- w_i – це невід'ємна функція, ваги, тобто ступінь важливості i -го «сусіда» для об'єкта u

Customer	Age	Income	No. credit cards	Class
George	35	35K	3	No
Rachel	22	50K	2	Yes
Steve	63	200K	1	No
Tom	59	170K	1	No
Anne	25	40K	4	Yes
John	37	50K	2	YES

Distance from John
$\text{sqrt } [(35-37)^2 + (35-50)^2 + (3-2)^2] = 15.16$
$\text{sqrt } [(22-37)^2 + (50-50)^2 + (2-2)^2] = 15$
$\text{sqrt } [(63-37)^2 + (200-50)^2 + (1-2)^2] = 152.23$
$\text{sqrt } [(59-37)^2 + (170-50)^2 + (1-2)^2] = 122$
$\text{sqrt } [(25-37)^2 + (40-50)^2 + (4-2)^2] = 15.74$

$$a(u, X, k) = \arg \max_y \sum_{i=1}^k w_{(i)} \left[y_{(i),u} = y \right]$$

□ *Оцінка близькості об'єкта u до i -го сусіда*

□ w_i – це невід'ємна функція, ваги, тобто ступінь важливості i -го «сусіда» для об'єкта u

Підбір вагів

- Параметр k зазвичай налаштовується за допомогою перехресної перевірки
- У класичному методі KNN всі об'єкти мають одиничні ваги: $w_i = 1$. Однак такий підхід не завжди є обґрунтованим
- Припустимо, що $k = 3$, $\rho(u, x_{(1), u}) = 1$, $\rho(u, x_{(2), u}) = 2$, $\rho(u, x_{(3), u}) = 100$.
- Очевидно, що третій сусіда розмістився дуже далеко і не повинен сильно впливати на відповідь. Це міркування реалізується через ваги:

$$w_i = F(\rho(u, x_{(i), u})),$$

Наведіть приклад функції F .

де $F(t)$ –

Підбір вагів

- Параметр k зазвичай налаштовується за допомогою перехресної перевірки
- У класичному методі KNN всі об'єкти мають одиничні ваги: $w_i = 1$. Однак такий підхід не завжди є обґрунтованим
- Припустимо, що $k = 3$, $\rho(u, x_{(1),u}) = 1$, $\rho(u, x_{(2),u}) = 2$, $\rho(u, x_{(3),u}) = 100$.
- Очевидно, що третій сусіда розмістився дуже далеко і не повинен сильно впливати на відповідь. Це міркування реалізується через ваги, **обернено пропорційні відстані**:

$$w_i = F(\rho(u, x_{(i),u})),$$

Наведіть приклад функції F .

де $F(t)$ – будь-яка монотонно спадна функція

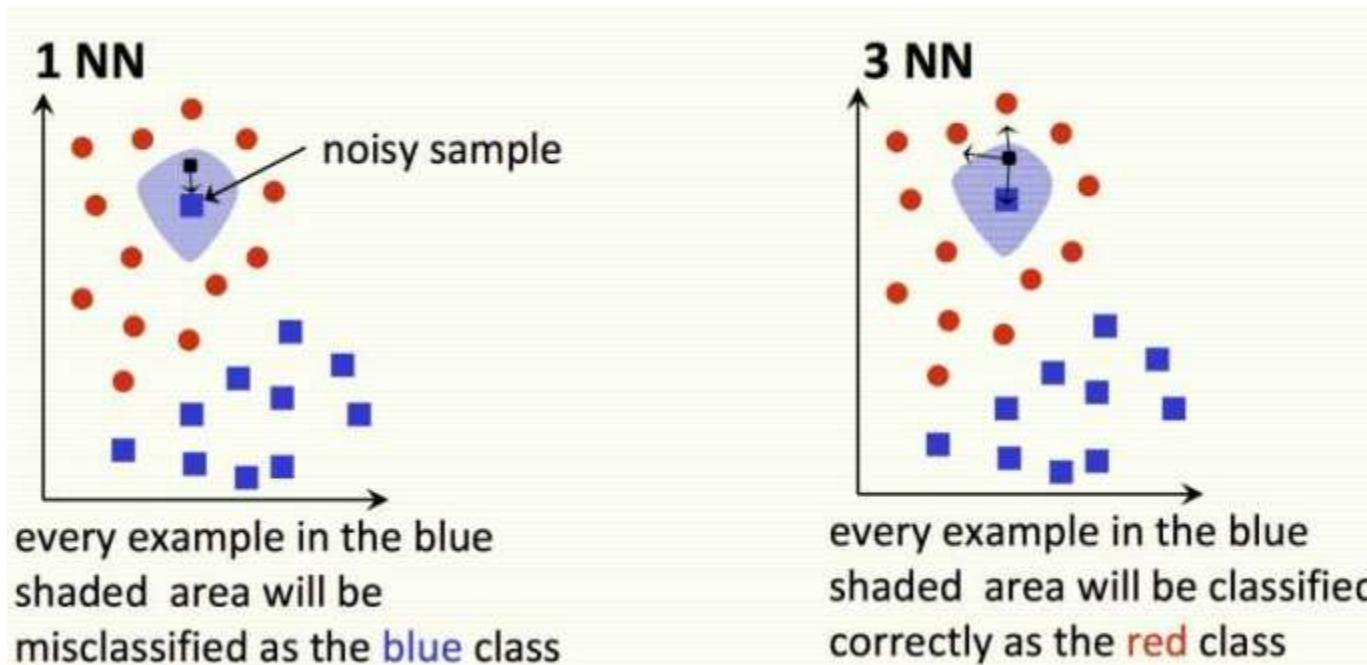
Наприклад, $F = 1/\rho$.

- За допомогою метода можна розв'язувати і задачі регресії.

- Для цього потрібно усереднити значення цільової функції по сусідам з урахуванням їхніх ваг:

$$a(u, X, k) = \frac{\sum_{i=1}^k w_{(i)} y_{(i),u}}{\sum_{i=1}^k w_{(i)}}$$

Чутливість до шумів



- Класифікатор **одного** найближчого сусіда є вкрай чутливим до шумових об'єктів і викидів, і межа між класами може виявитися дуже складною
- Зі збільшенням k межа згладжується за рахунок «усереднення» по декількох об'єктах

Зашумлені характеристики

- Зашумлені характеристики можуть зробити сильний вплив на метрику
- Виявити такі характеристики можна, видаляючи по черзі всі характеристики і дивлячись на помилку на тестовій вибірці. Більш складні методи відбору інформативних характеристик будуть розібрані на наступних лекціях

Нормування характеристик

- Помножимо одну з характеристик (наприклад, першу) на константу С
- Евклідова відстань прийме наступний вид

$$\rho_2(x, y) = \sqrt{(C(x_1 - y_1))^2 + \sum_{i=2}^d (x_i - y_i)^2}.$$

- Таким чином, відмінність за першою характеристикою буде вважатися в С раз більш значущою, ніж відмінності за всіма іншими. При цьому розташування об'єктів один відносно іншого не змінилося — змінився лише масштаб (тобто міряємо у метрах чи кілометрах, наприклад)!

Customer	Age	Income	No. credit cards	Class	Distance from John
George	35	35K	3	No	$\sqrt{[(35-37)^2 + (35-50)^2 + (3-2)^2]} = 15.16$
Rachel	22	50K	2	Yes	$\sqrt{[(22-37)^2 + (50-50)^2 + (2-2)^2]} = 15$
Steve	63	200K	1	No	$\sqrt{[(63-37)^2 + (200-50)^2 + (1-2)^2]} = 152.23$
Tom	59	170K	1	No	$\sqrt{[(59-37)^2 + (170-50)^2 + (1-2)^2]} = 122$
Anne	25	40K	4	Yes	$\sqrt{[(25-37)^2 + (40-50)^2 + (4-2)^2]} = 15.74$
John	37	50K	2	YES	

Врахування тисяч (K) в розрахунках може кардинально змінити результат порівняння

- Таким чином, відмінність за першою характеристикою буде вважатися в С раз більш значущою, ніж відмінності за всіма іншими. При цьому розташування об'єктів один відносно іншого не змінилося — змінився лише масштаб!

Нормування характеристик

- ❑ Щоб уникнути подібних проблем, ознаки потрібно нормувати
- ❑ Широко застосовуються такі способи:
 - нормування на середньо квадратичне відхилення (СКВ):

$$x_{\text{норм. } i} = \frac{x_i - \bar{x}_i}{\text{СКВ}(x_i)}$$

- нормування на відрізок [0; 1] шляхом ділення на розмах:

$$x_{\text{норм. } i} = \frac{x_i - \min(x_i)}{\max(x_i) - \min(x_i)}$$

- у цих формулах x_i – це вектор, складений із i -тих ознак усіх об'єктів (тобто це i -та колонка матриці «об'єкти-ознаки»)

Нормування характеристик

- Щоб уникнути подібних проблем, ознаки потрібно нормувати
- Широко застосовуються такі способи:
 - нормування на середньо квадратичне відхилення (СКВ):

$$x_{\text{норм. } i} = \frac{x_i - x_{\text{сер } i}}{\text{СКВ}(x_i)}$$

- нормування на відрізок [0; 1] шляхом ділення на розмах:

$$x_{\text{норм. } i} = \frac{x_i - \min(x_i)}{\max(x_i) - \min(x_i)}$$

Якщо це фінансові дані, то краще, можливо, нормувати на відрізок [-1; 1]. Як змінити формулу?

- у цих формулах x_i – це вектор, складений із i -тих ознак усіх об'єктів (тобто це i -та колонка матриці «об'єкти-ознаки»)

Нормування характеристик

- ❑ Щоб уникнути подібних проблем, ознаки потрібно нормувати
- ❑ Широко застосовуються такі способи:
 - нормування на середньо квадратичне відхилення (СКВ):

$$x_{\text{норм. } i} = \frac{x_i - \bar{x}_i}{\text{СКВ}(x_i)}$$

- нормування на відрізок [0; 1] шляхом ділення на розмах:

$$x_{\text{норм. } i} = \frac{2x_i - \min(x_i) - \max(x_i)}{\max(x_i) - \min(x_i)}$$

Якщо це фінансові дані, то краще, можливо, нормувати на відрізок [-1; 1]. Як змінити формулу?

- у цих формулах x_i – це вектор, складений із i -тих ознак усіх об'єктів (тобто це i -та колонка матриці «об'єкти-ознаки»)

Метрика Мінковського

$$\rho_p(x, y) = \left(\sum_{i=1}^d |x_i - y_i|^p \right)^{1/p}$$

□ Окремі випадки метрики Мінковського:

- ($p=2$) **Евклідова метрика**, визначає відстань як довжину прямої, що з'єднує дві визначені точки в одно-, два-, ..., багатовимірному просторі
- ($p=1$) **Манхеттенська відстань**, визначає відстань як мінімальну довжину шляху між визначеними точками за умови, що можна рухатися лише паралельно осям координат
- ($p=\infty$) **Метрика Чебишева**, визначає відстань як максимальну із відстаней між координатами цих двох точок:

$$\rho_\infty(x, y) = \max_{i=1, \dots, d} |x_i - y_i|.$$

Метрика Мінковського

□ Окремі випадки

- ($p=2$) **Евклідова** прямої, що з'єднує дві точки в багатовимірному просторі
- ($p=1$) **Манхеттен** або **норма** вимірює мінімальну довжину підпідходу під умови, що можна перескочити тільки по одній осі
- ($p=\infty$) **Метрика Чебишева** вимірює максимальну із відстаней від будь-якої точки до будь-якої з координатних осей



$i=1, \dots, d$

Метрика Мінковського

$$\rho_p(x, y) = \left(\sum_{i=1}^d |x_i - y_i|^p \right)^{1/p}$$

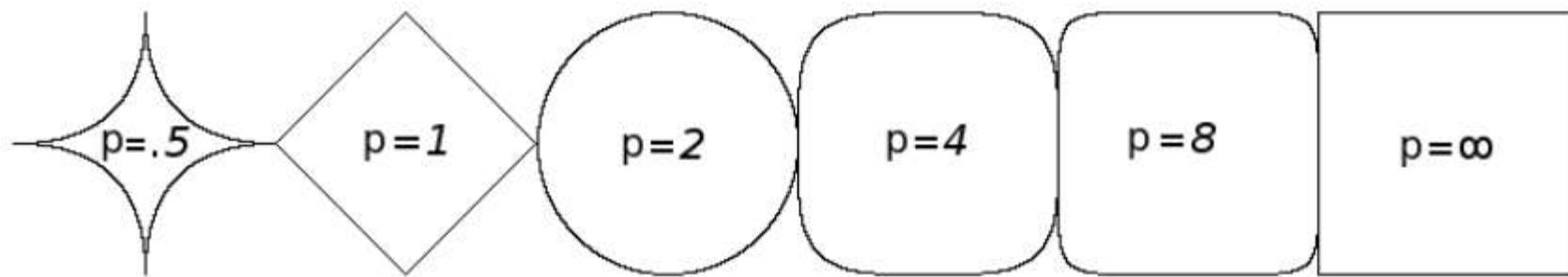
□ Окремі випадки метрики Мінковського:

- ($p=2$) **Евклідова метрика**, визначає відстань як довжину прямої, що з'єднує дві визначені точки в одно-, два-, ..., багатовимірному просторі
- ($p=1$) **Манхеттенська відстань**, визначає відстань як мінімальну довжину шляху між визначеними точками за умови, що можна рухатися лише паралельно осям координат
- ($p=\infty$) **Метрика Чебишева**, визначає відстань як максимальну із відстаней між координатами цих двох точок:

$$\rho_\infty(x, y) = \max_{i=1, \dots, d} |x_i - y_i|.$$

Метрика Мінковського

$$\rho_p(x, y) = \left(\sum_{i=1}^d |x_i - y_i|^p \right)^{1/p}$$



- По мірі збільшення параметра p метрика слабкіше штрафує невеликі відмінності між векторами і сильніше штрафує значні відмінності

Косинусна міра

- Скалярний добуток двох векторів x і y , між якими кут θ , рахується як

$$(x, y) = \|x\| \|y\| \cos \theta$$

- Відповідно косинусна відстань між цими векторами визначається як

$$\rho_{\cos}(x, y) = \arccos\left(\frac{(x, y)}{\|x\| \|y\|}\right) = \arccos\left(\sum_{i=1}^d x_i y_i \Bigg/ \left(\sqrt{\sum_{i=1}^d x_i^2} \sqrt{\sum_{i=1}^d y_i^2}\right)\right)$$

- Косинусна міра використовується для вимірювання схожості між текстами
- Кожен документ описується вектором, кожна компонента якого відповідає слову зі словника
- Компонента дорівнює одиниці, якщо відповідне слово зустрічається в тексті, і нулю в іншому випадку

Косинусна міра

- Скалярний добуток двох векторів x і y , між якими кут θ , рахується як

$$(x, y) = \|x\| \|y\| \cos \theta$$

- Відповідно косинусна відстань між цими векторами визначається як

$$\rho_{\cos}(x, y) = \arccos\left(\frac{(x, y)}{\|x\| \|y\|}\right) = \arccos\left(\sum_{i=1}^d x_i y_i \Bigg/ \left(\sqrt{\sum_{i=1}^d x_i^2} \sqrt{\sum_{i=1}^d y_i^2}\right)\right)$$

- Тоді чим більше слів зустрічаються в цих двох документах одночасно, тим кут між двома векторами буде менше, тобто відстань між документами буде менше, вони є більш схожими між собою
- Косинусна міра не залежить, наприклад, від розмірів порівнюваних текстів, вимірюючи лише обсяг їх схожості
- Правильніше було б косинусну міру називати арккосинусною 😊

Відстань Жаккара (Paul Jaccard)

- Якщо об'єктами є множини (наприклад, кожен об'єкт — це текст, представлений множиною слів), то їх відмінність/схожість можна також вимірювати за допомогою відстані Жаккара:

$$\rho_J(A, B) = 1 - \frac{|A \cap B|}{|A \cup B|}$$

Jaccard, P. (1901) Distribution de la flore alpine dans le bassin des Dranses et dans quelques régions voisines. *Bulletin de la Société Vaudoise des Sciences Naturelles* 37, 241-272.

Hamming Distance

"karolin" and "kathrin" is 3

"karolin" and "kerstin" is 3

1011101 and 1001001 is 2

2173896 and 2233796 is 3

Для вимірювання подібності між двома рядками (наприклад, послідовностями ДНК) можна використовувати редакторську відстань, що дорівнює мінімальному числу вставок і вилучень символів, за допомогою яких можна перетворити перший рядок на другий

Алгоритм k найближчих сусідів. Як знайти найближчих сусідів?

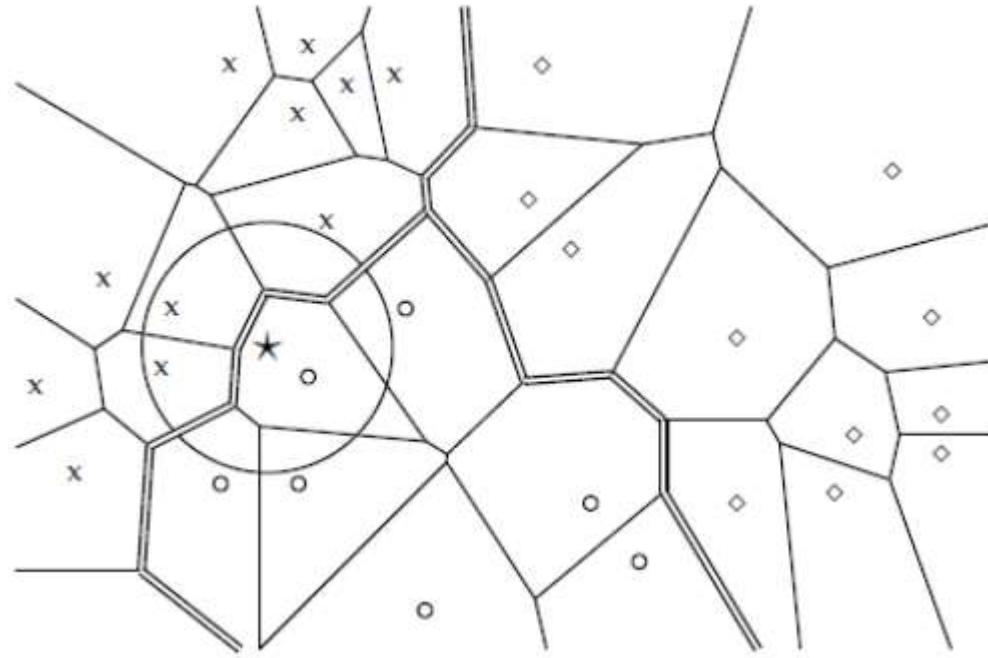
- Для класифікації:

1. Вибрати число k і метрику відстані
2. Знайти k найближчих сусідів зразка, який ми хочемо класифікувати
3. Присвоїти мітку класу мажоритарним голосуванням

- Для регресії:

1. Вибрати число k і метрику відстані
2. Знайти k найближчих сусідів зразка, для якого ми хочемо знайти значення
3. Присвоїти їх середньоарифметичне значення

Евклідова метрика і діаграма Вороного



$$\rho(x, y) = \left(\sum_{i=1}^d |x_i - y_i|^2 \right)^{1/2}$$

- Діаграма Вороного, відповідна вибірці X_ℓ — це таке розбиття простору на області, що кожна область складається з точок, для яких одна і та ж точка з вибірки є найближчою

Діаграма Вороного

- Більш формально, діаграма Вороного для вибірки X_ℓ складається з ℓ областей R_1, \dots, R_ℓ , що визначаються як

$$R_i = \{x \in \mathbb{R}^d \mid \rho(x, x_i) < \rho(x, x_j), j \neq i\}$$

- Очевидно, що при використанні класифікатора найближчого сусіда ($k = 1$) межа між класами є підмножиною меж між такими областями

Точні методи пошуку найближчих сусідів

- Обмежимось евклідовою метрикою
- Візьмемо $k = 1$, тобто будемо розглядати задачу пошуку одного найближчого сусіда (всі методи нескладно узагальнюються на випадок з $k > 1$) **А як це зробити?**

Точні методи пошуку найближчих сусідів

- Обмежимось евклідовою метрикою
- Візьмемо $k = 1$, тобто будемо розглядати задачу пошуку одного найближчого сусіда (всі методи нескладно узагальнюються на випадок з $k > 1$)
- Якщо просто перебирати всі об'єкти навчальної вибірки, шукаючи найбільш близький до нового об'єкту, то отримуємо складність $O(\ell d)$,
де ℓ – розмір навчальної вибірки,
 d – кількість факторів (= характеристик, ознак)
- **Може якось евристично можна спростити цю переборну задачу?**

Точні методи пошуку найближчих сусідів

- Прості евристики не дуже допомагають

Можна вибрати підмножину ознак, і спочатку обчислити відстань тільки по цих координатах

Вона є нижньою оцінкою на повноцінну відстань, і якщо вже вона більше, ніж поточний найкращий результат, то даний об'єкт можна більше не розглядати як кандидата в найближчого сусіда

Але за великої розмірності задачі експериментально було показано, такий підхід також дає $O(\ell d)$

Weber, R., Schek, H. J., Blott, S. (1998). A Quantitative Analysis and Performance Study for Similarity-Search Methods in High-Dimensional Spaces. Proc. of the 24th VLDB conf., New York, 194–205 pp.

Наближені методи пошуку найближчих сусідів

- Два способи боротьби з високою складністю пошуку найближчих сусідів при великій кількості ознак:
 1. Запам'ятувати не всю навчальну вибірку, а лише її представницьку підмножину, вирану за певною евристикою (наприклад, **алгоритм STOLP**)
 2. Шукати к найближчих сусідів наблизено, тобто дозволяти результату пошуку бути **трохи** далі від нового об'єкта, ніж к його справжніх сусідів (наприклад, **метод Locality Sensitive Hashing, LSH**)

Усі ці методи розглянемо в магістерському курсі «Машинне навчання»!

Кластерний аналіз = кластеризація (=Clustering)= стратифікація = таксономія

- **Кластерний аналіз** призначений для розбиття множини об'єктів на визначене чи невідоме число класів (= кластерів) згідно з певним критерієм якості класифікації
- Цей критерій тією чи іншою мірою повинен відображати такі нeформальні вимоги:
 - а) всередині кластерів об'єкти повинні бути тісно пов'язані між собою
 - б) об'єкти різних кластерів повинні бути далекими одне від одного
 - в) за інших рівних умов розподіли об'єктів по кластерам повинні бути рівномірними

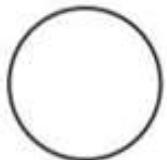
Кластерний аналіз (=Clustering)

- **Кластерний аналіз** призначений для розбиття множини об'єктів на визначене чи невідоме число класів згідно з певним критерієм якості класифікації
 - Цей критерій тією чи іншою мірою повинен відображати такі неформальні вимоги:
 - а) всередині кластерів об'єкти повинні бути тісно пов'язані між собою
 - б) об'єкти різних кластерів повинні бути далекими одне від одного
 - в) за інших рівних умов розподіли об'єктів по кластерам повинні бути **рівномірними**
- 
- концепція (гіпотеза)
компактності
класів розбиття
- щоб не нав'язувати
об'єднання в
окремі групи

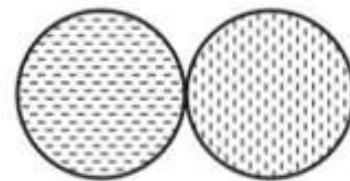
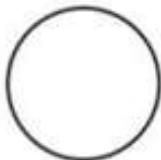
Приклади конкретних задач кластеризації

- Сегментація цільової аудиторії сайту для наступної маркетингової реклами (targeted advertising)
- Ідентифікація груп сімей – споживачів певного товару для розробки стратегії позиціонування бренду
- Тематичне (topic) моделювання електронних листів
- Групування фільмів в рекомендаційній системі
- Кластеризація символів в незалежності від їх шрифту, розміру тощо (для подальшого розпізнавання)

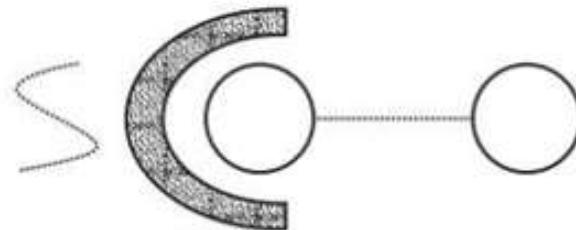
Type of clusters



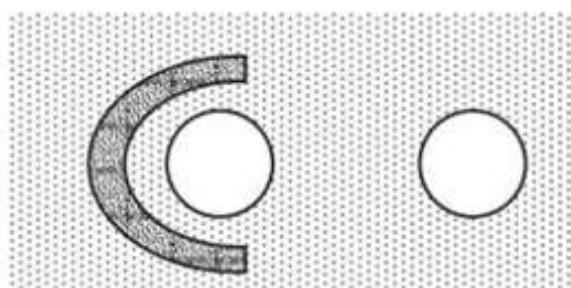
(a) Well-separated clusters. Each point is closer to all of the points in its cluster than to any point in another cluster.



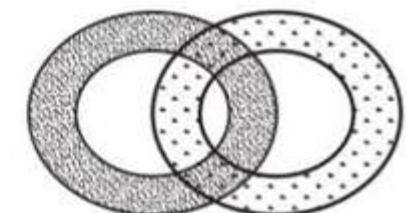
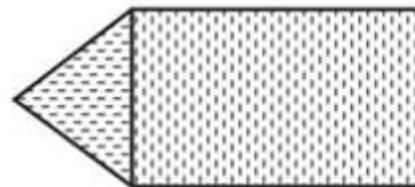
(b) Center-based clusters. Each point is closer to the center of its cluster than to the center of any other cluster.



(c) Contiguity-based clusters. Each point is closer to at least one point in its cluster than to any point in another cluster.

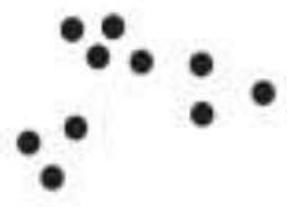


(d) Density-based clusters. Clusters are regions of high density separated by regions of low density.



(e) Conceptual clusters. Points in a cluster share some general property that derives from the entire set of points. (Points in the intersection of the circles belong to both.)

20 points but how many clusters?



(a) Original points.



(b) Two clusters.



(c) Four clusters.



(d) Six clusters.

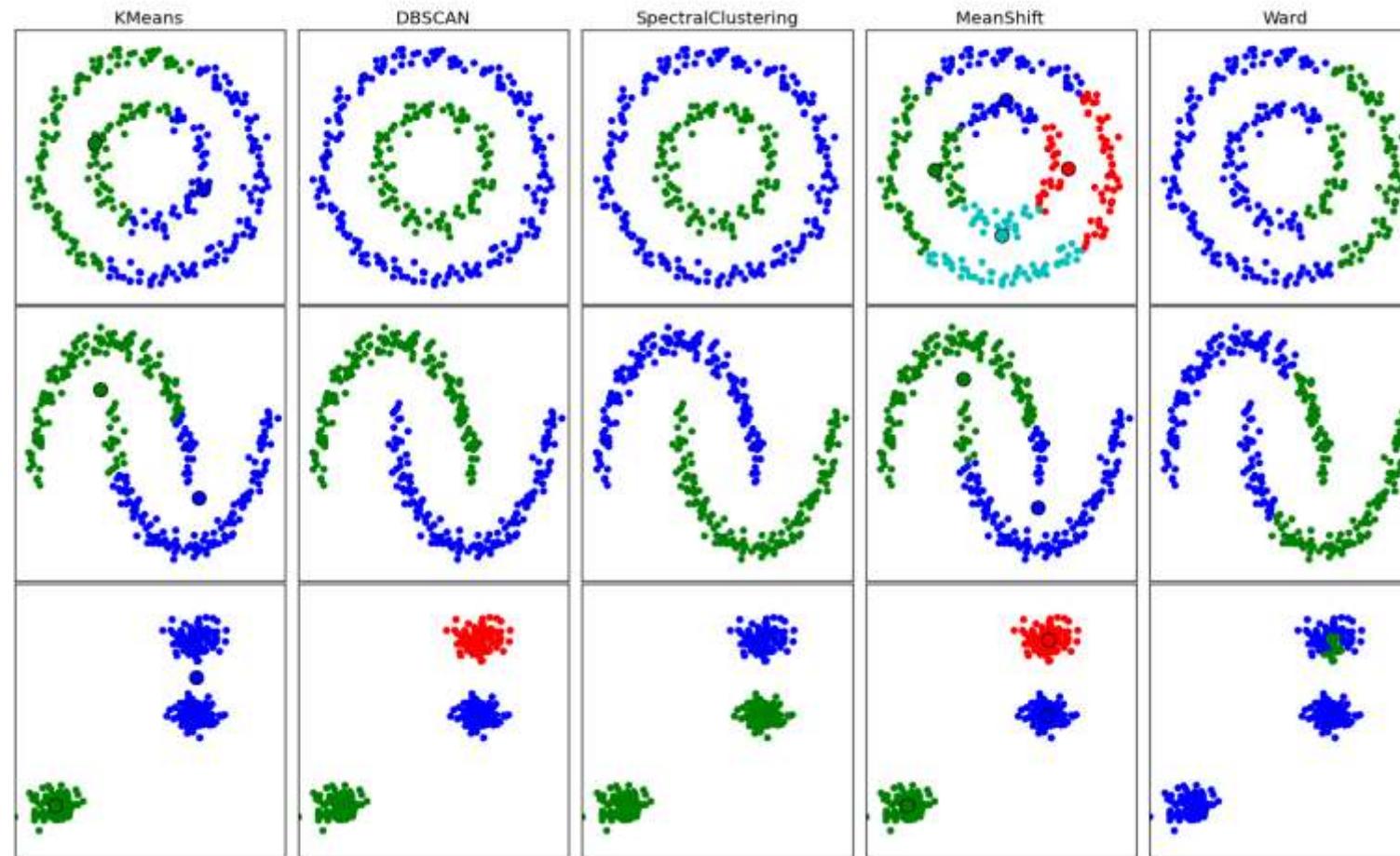
Задача кластеризації ставиться нечітко

- Невідомі властивості кластерів
- Невідома їх кількість
- Невідомо, чи є вони взагалі?
- Відсутня навчальна вибірка
- Відсутні очевидні критерії (метрики) якості
- **Але на відміну від класифікації не потрібні мітки (не потрібен учитель). Тому можна обробляти суттєво більшу кількість об'єктів!**
- Скільки кластерів?
- Як рахувати відстань між ними (чи їх близькість)?
- За якими правилами об'єднувати елементи в кластери?

Мета (результат) кластеризації

- Розбиття об'єктів на групи
- Знаходження типових точкових представників класів:
 - центроїдів (для неперервних значень)
 - медоїдів=кластроїдів (для дискретних значень)
- Знаходження нетипових представників класів (викидів)
- Побудова повної ієрархії груп об'єктів (таксономія)
- Стиснення даних
- Розуміння даних

Різні методи кластеризації



Обираємо відповідний метод в залежності від природи даних і нашого їх розуміння: чи відомі приблизна форма і розмір кластерів? Чи можуть вони бути вкладеними? Чи можуть об'єкти належати одночасно декільком кластерам?

Типи кластерних алгоритмів

- **Пласкі алгоритми (Partitioning Clustering):**

- ❖ починають роботу розділенням елементів по групах випадковим чином
- ❖ ітеративно покращують результат
- ❖ головний алгоритм: K-середніх (K-means)

- **Алгоритми ієрархічної кластеризації (Hierarchical Clustering):**

- ❖ створюють ієрархію
- ❖ знизу-вгору (агломеративні)
- ❖ зверху-вниз (розділяючі)

- **Жорстка кластеризація:**

- ❖ кожен елемент належить строго одному кластеру

- **М'яка кластеризація:**

- ❖ елемент може належати кільком кластерам

- **Ймовірнісна кластеризація:**

- ❖ елемент може належати кільком кластерам з однаковою чи різною ймовірністю

Clustering

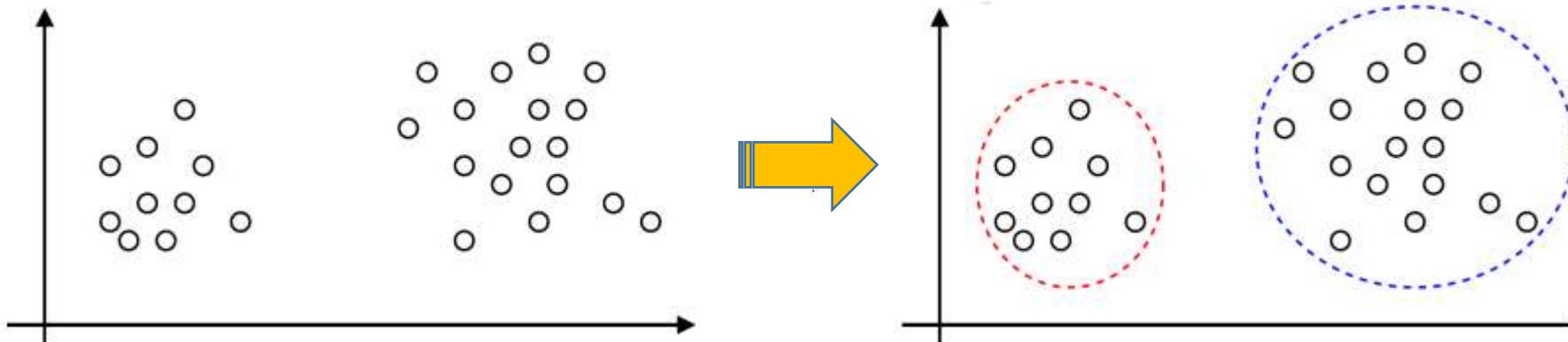
- **Clustering** is a type of **unsupervised learning** in which the goal is to partition a set of examples into groups called **clusters**
- Intuitively, the examples within a cluster are more similar to each other than to examples from other clusters
- In order to measure the similarity between examples, clustering algorithms use various **distortion** or **distance measures** (=міри відмінності та метричні міри)

Зведемо задачу до геометричної

- Кожен об'єкт – точка
- Функції відстані – Евклідово, косинусне, Манхетенське, Жаккара, Хеммінга, ...
- Схожі об'єкти розташовані «близько» одні до одного
- Розрізняються об'єкти, що розташовані «далеко»
- Скупчення точок – кластер

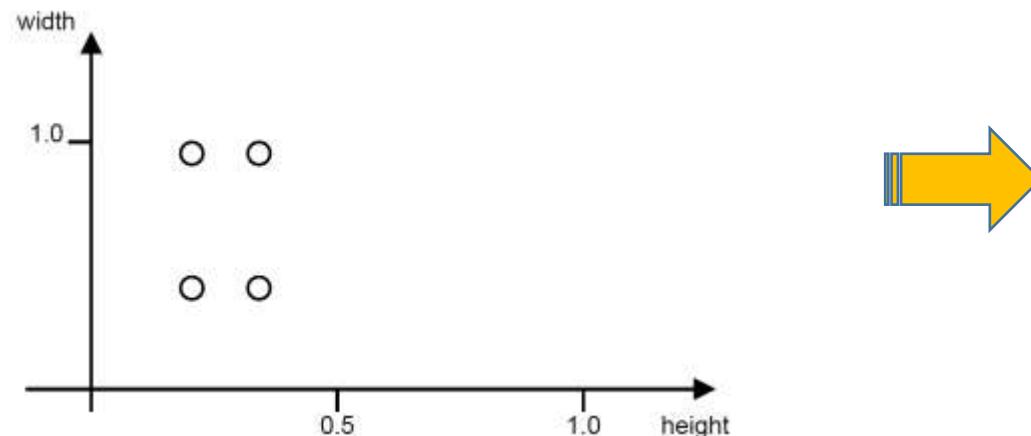
Кластерний аналіз (=Clustering)

- **Кластер** – це множина об'єктів, які схожі між собою і одночасно несхожі на об'єкти інших кластерів
- Гарний метод кластеризації породжує кластери з високою схожістю всередині кластера (high intra-class similarity) і низькою міжкластерною схожістю (low inter-class similarity)
- Якщо мірилою схожості є відстань між об'єктами, то маємо метричну кластеризацію (distance-based clustering)



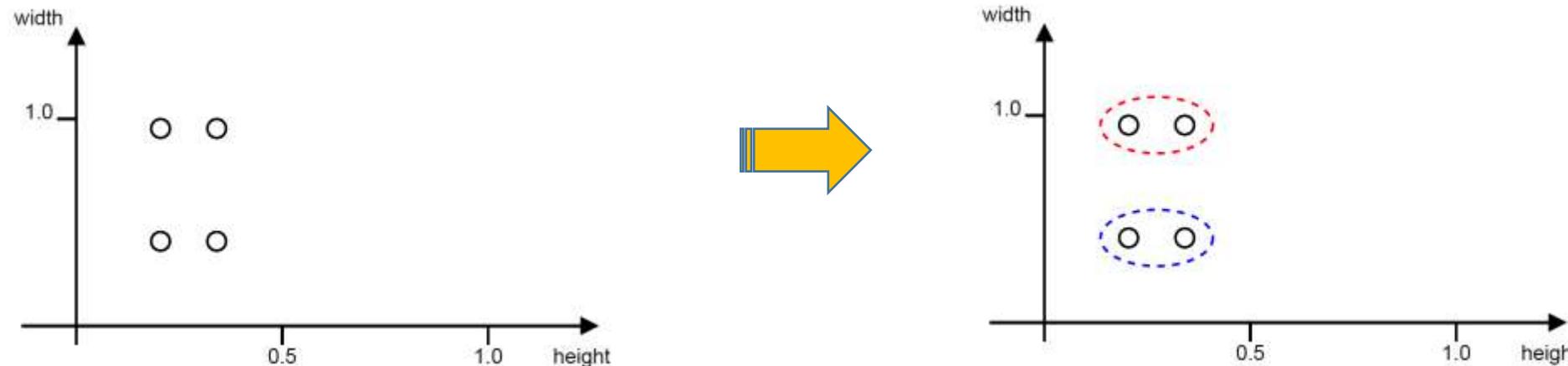
Метрична кластеризація

- Очевидно, що метрична кластеризація залежить від обраної метрики (евклідова, Манхетенська, косинусна, ...)
- Не так очевидно, що вона, залежить від ...



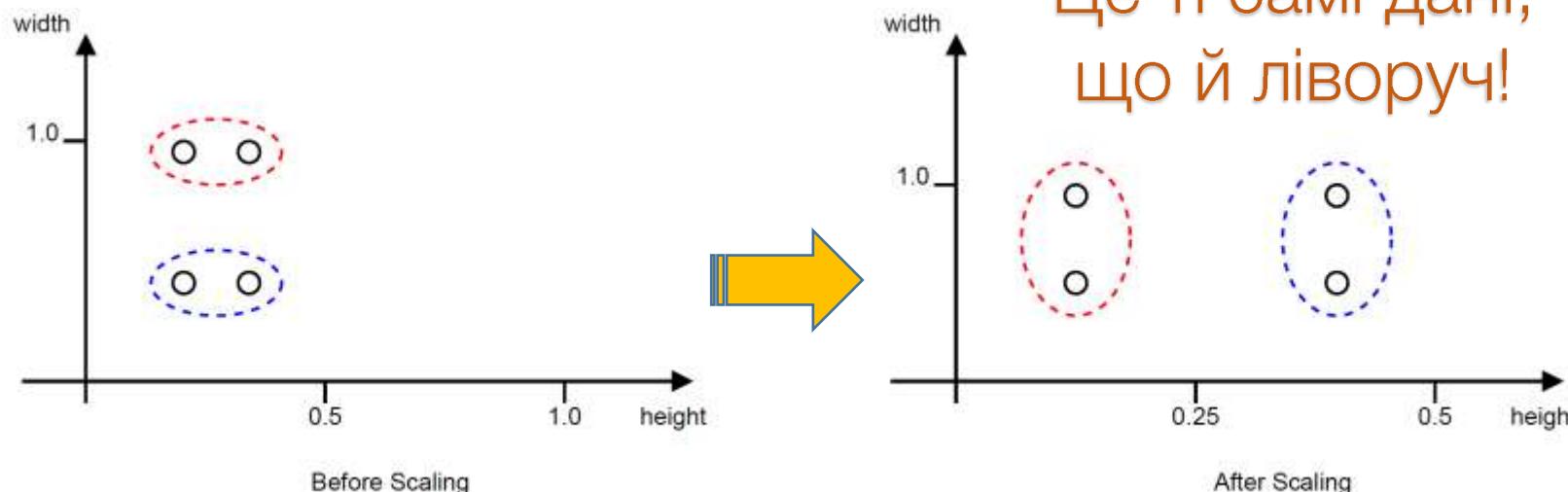
Метрична кластеризація

- Очевидно, що метрична кластеризація залежить від обраної метрики (евклідова, Манхетенська, косинусна, ...)
- Не так очевидно, що вона, залежить від ...



Метрична кластеризація

- Очевидно, що метрична кластеризація залежить від обраної метрики (евклідова, Манхетенська, косинусна, ...)
- Не так очевидно, що вона, залежить **від масштабування** даних (так само, як і розглянуті раніше метричні методи класифікації)



Можливі характеристики кластерів

- **Діаметр:** максимальна відстань між будь-якими двома точками в кластері
- **Радіус:** максимальна відстань від якогось «центру» до будь-якої з точок кластера
- **Щільність:** кількість точок в кластері, поділена на «обсяг», тобто на радіус в якісь степені
- **Міжкластерна відстань:** відстань між центрами, між найближчими точками, середня відстань між усіма парами, ...

Критерії зупинки кластеризації

- **А які можуть бути критерії зупинки кластеризації?**

Критерії зупинки кластеризації

- Кластери не змінилися під час останньої ітерації
- Характеристики кластерів (діаметр, щільність, ...) досягли граничних значень
- Побудована потрібна кількість кластерів

Центри кластерів

- В евклідовому просторі є центр ваги (центроїд) – середнє арифметичне координат точок кластера
- У неевклідовому просторі (наприклад, в просторі слів) центроїда немає. Центром (кластроїдом=медоїдом) вибирається одна з точок кластера, що мінімізує
 - ❖ максимальну відстань до інших точок
 - ❖ суму відстаней (чи суму квадратів відстаней)

Partitioning Clustering

- Partitioning methods construct a partition of N objects into K clusters
- K is known, and the partition optimizes a partitioning criterion
- Global optimal: exhaustively enumerate all partitions select the best one:
 - Unfeasible due to the high number of possible partitions: C_N^K
- Heuristic methods: select cluster representatives and optimize those
 - **k-means** (MacQueen'67): clusters are represented by the points' centroids
 - **k-medoids** or PAM (Partition around medoids) (Kaufman & Rousseeuw'87): each cluster is represented by one of the objects in the cluster

Пласкі методи (Partitioning Clustering).

Метод k-середніх (k-means)

- Кожен кластер визначається своїм **центроїдом**
- Критерій кластеризації**: мінімізувати усереднену відстань точок кластера від його центроїда (*inertia*, or **within-cluster sum of squares**)

$$\sum_{i=0}^n \min_{\mu_j \in C} (||x_i - \mu_j||^2)$$

- Визначення центроїда:

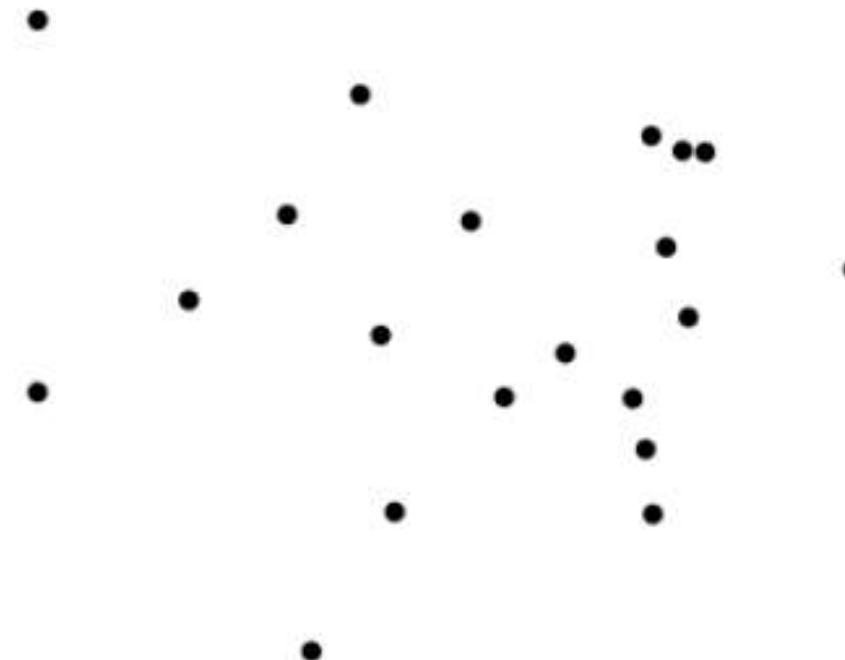
де w позначає кластер

$$\vec{\mu}(\omega) = \frac{1}{|\omega|} \sum_{\vec{x} \in \omega} \vec{x}$$

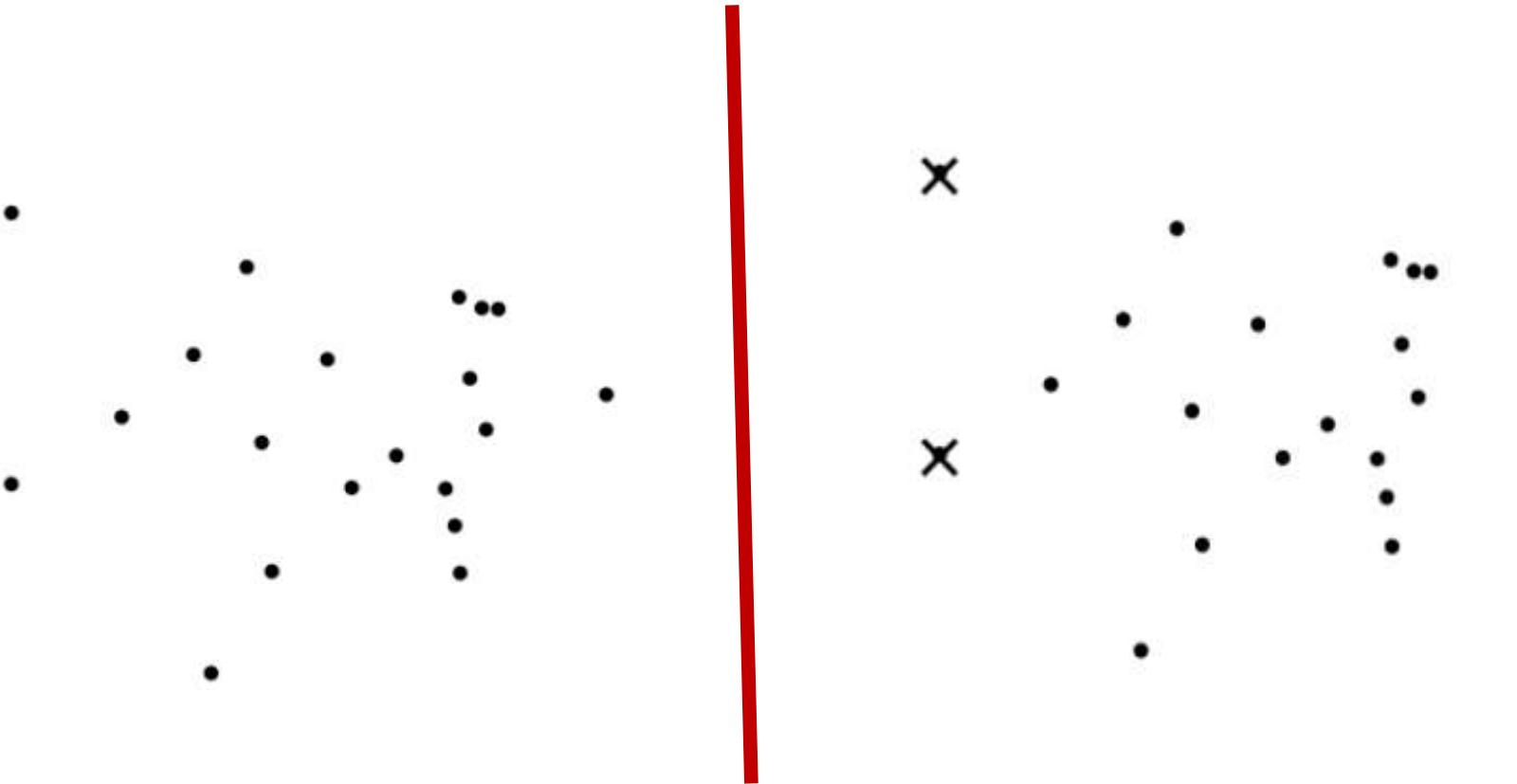
- Ітеративно застосовуємо два кроки алгоритму:

- ❖ **перерозподіл**: зараховуємо кожен об'єкт до найближчого центроїду
- ❖ **перерахунок**: заново розраховуємо кожен центроїд як середнє об'єктів, віднесених до кластера на попередньому кроці

Приклад: кластеризація набору даних методом k-середніх

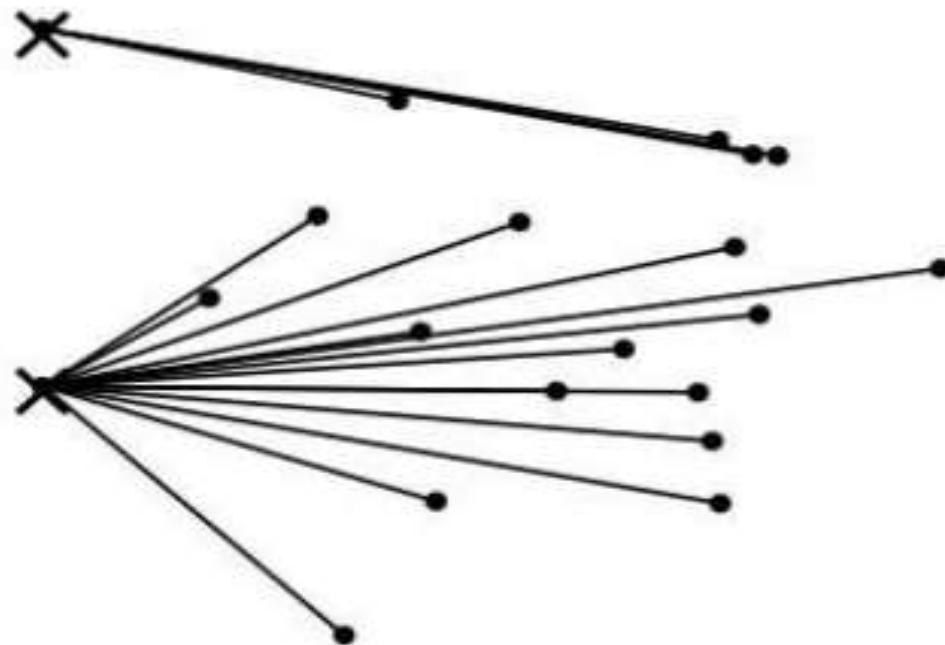


Приклад: кластеризація набору даних методом k-середніх

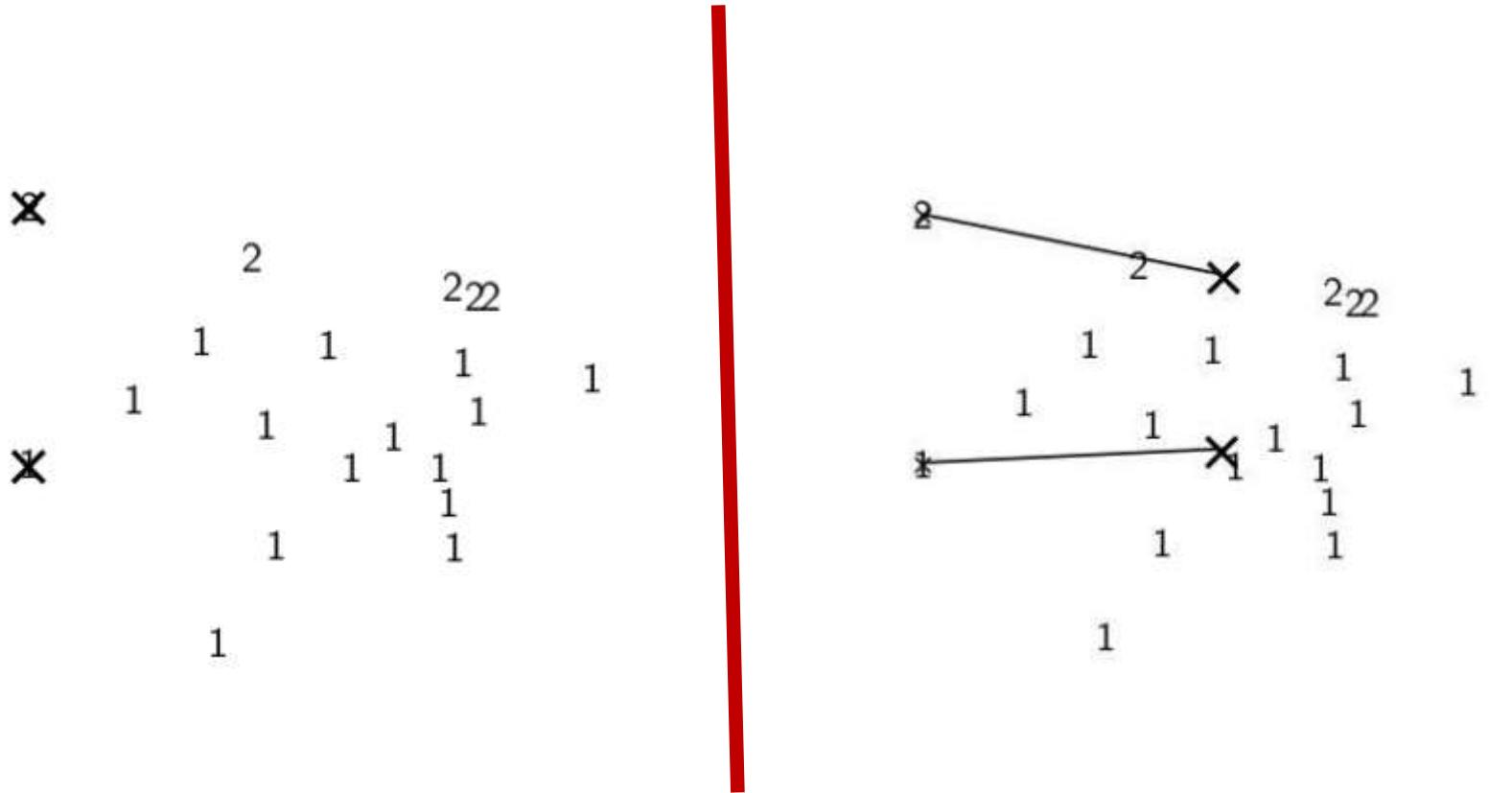


- Вибираємо випадковим чином два центроїда

Ітерація 1. Розподіляємо кожну точку до найближчого центроїда



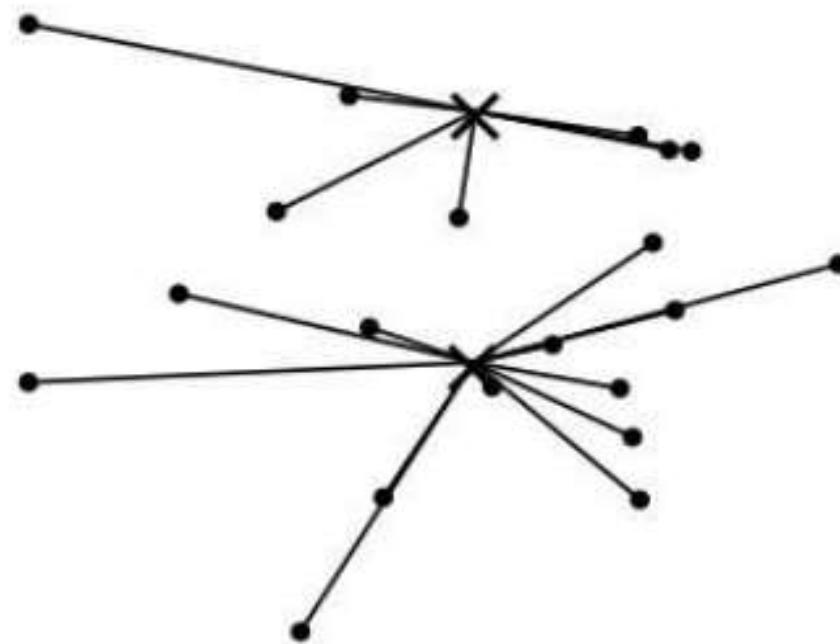
Ітерація 1



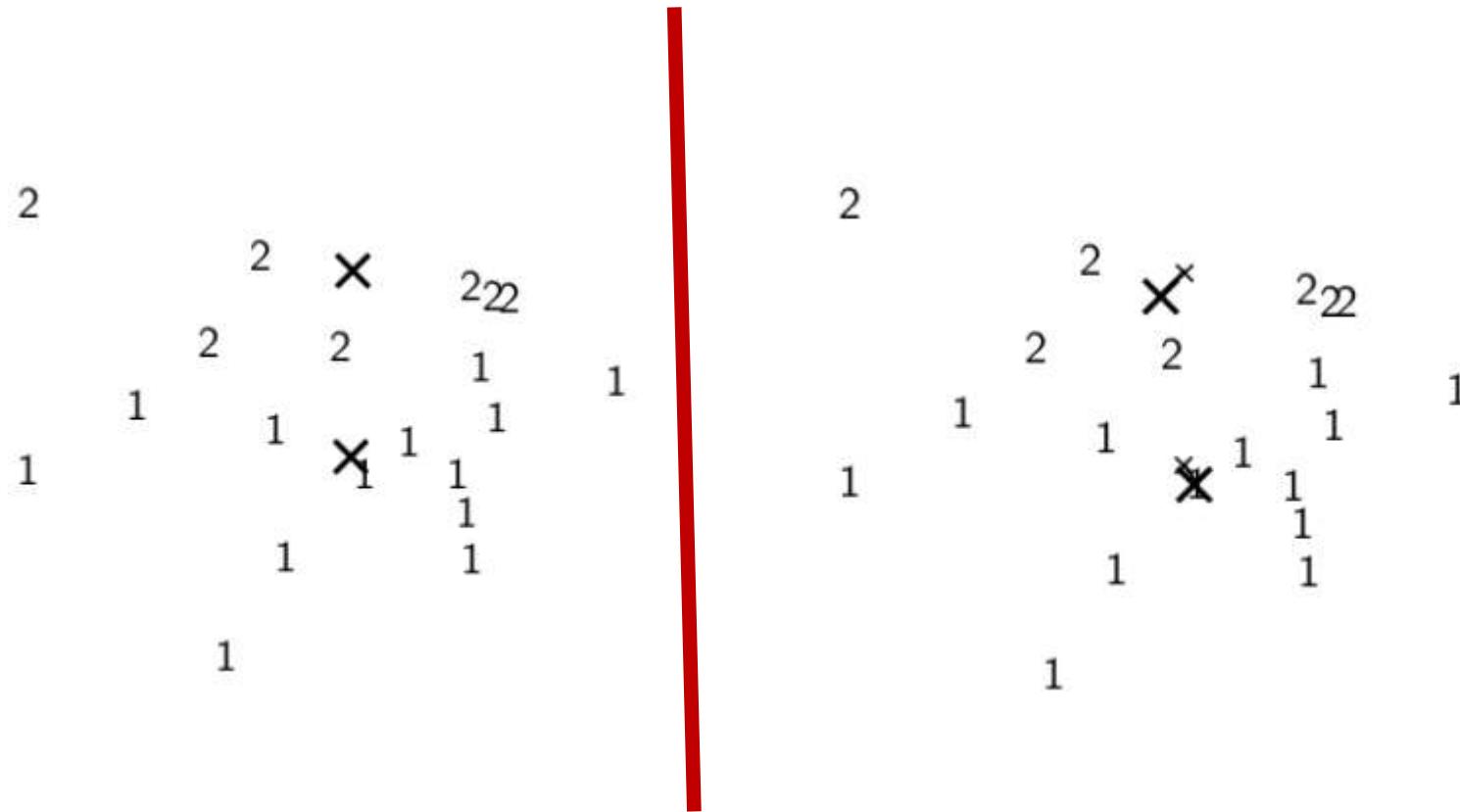
□ Результат розподілу

□ Перераховуємо
центроїди кластерів

Ітерація 2. Розподіляємо кожну точку до найближчого центроїда



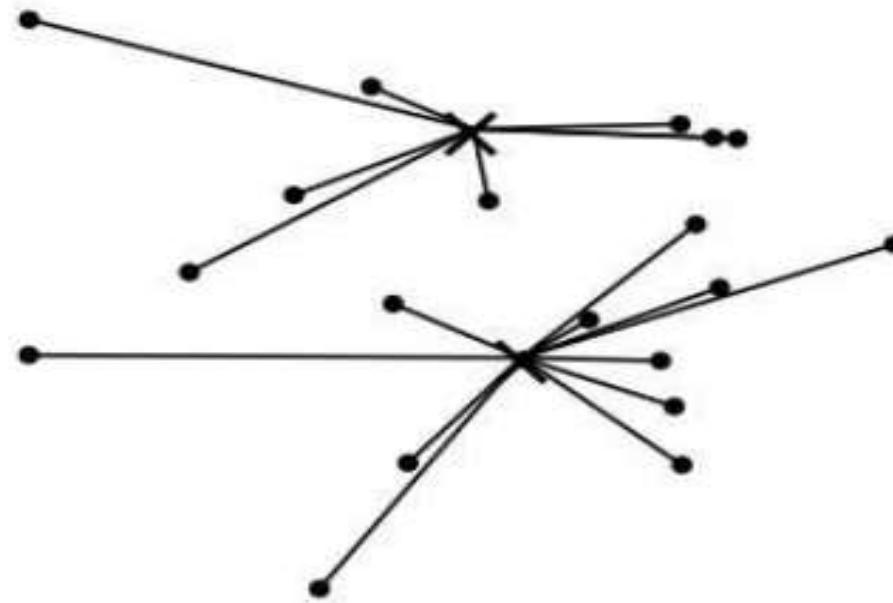
Ітерація 2



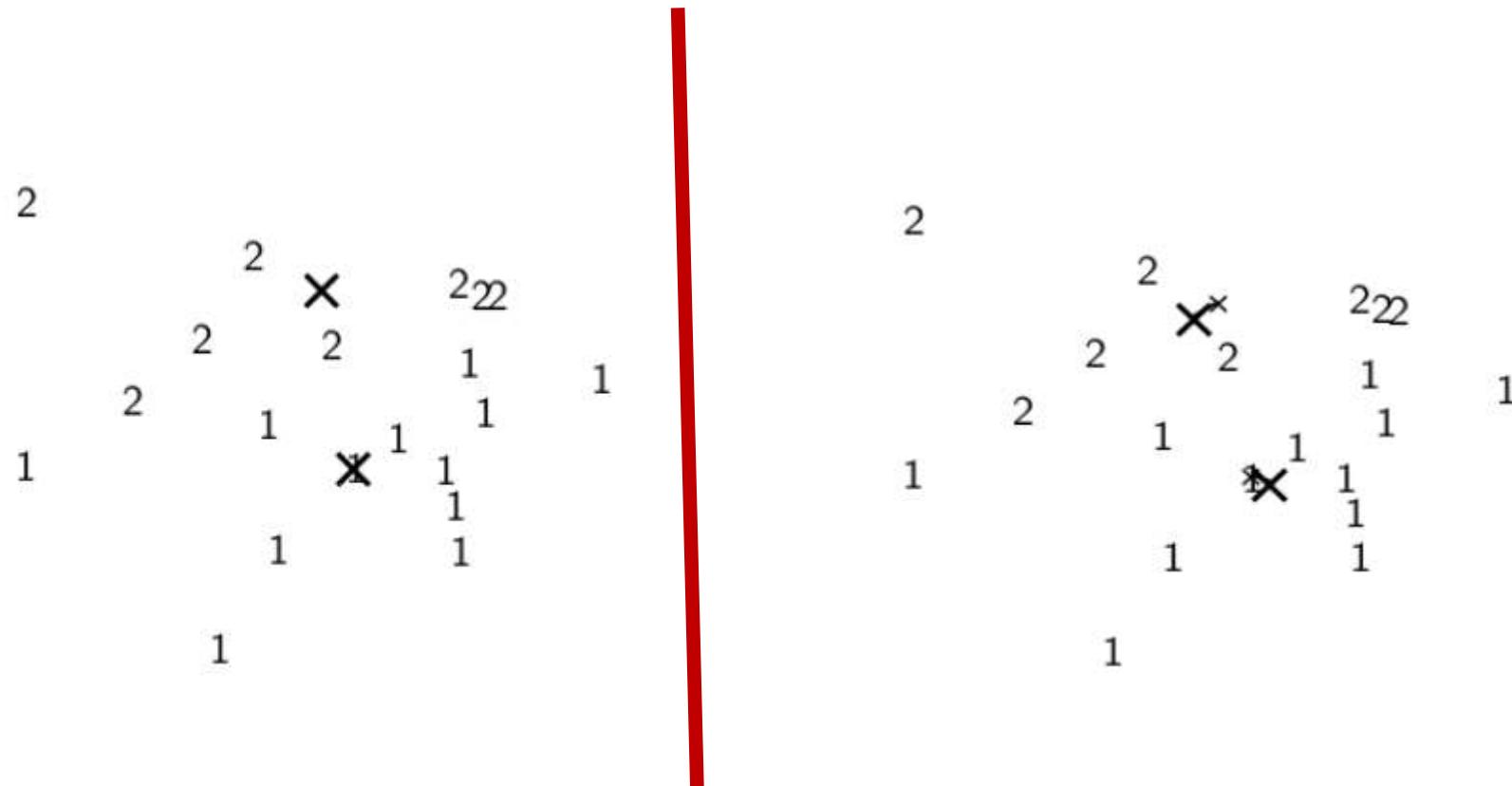
□ Результат розподілу

□ Перераховуємо
центроїди кластерів

Ітерація 3. Розподіляємо кожну точку до найближчого центроїда

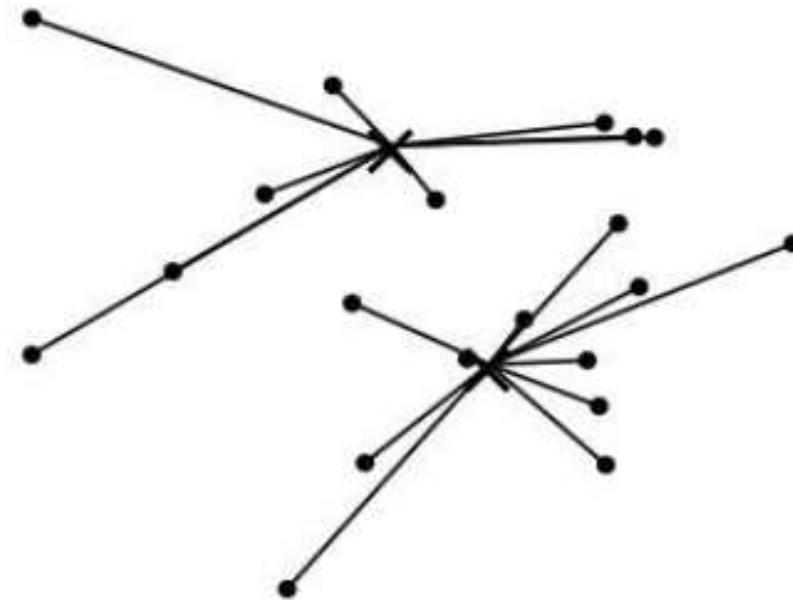


Ітерація 3

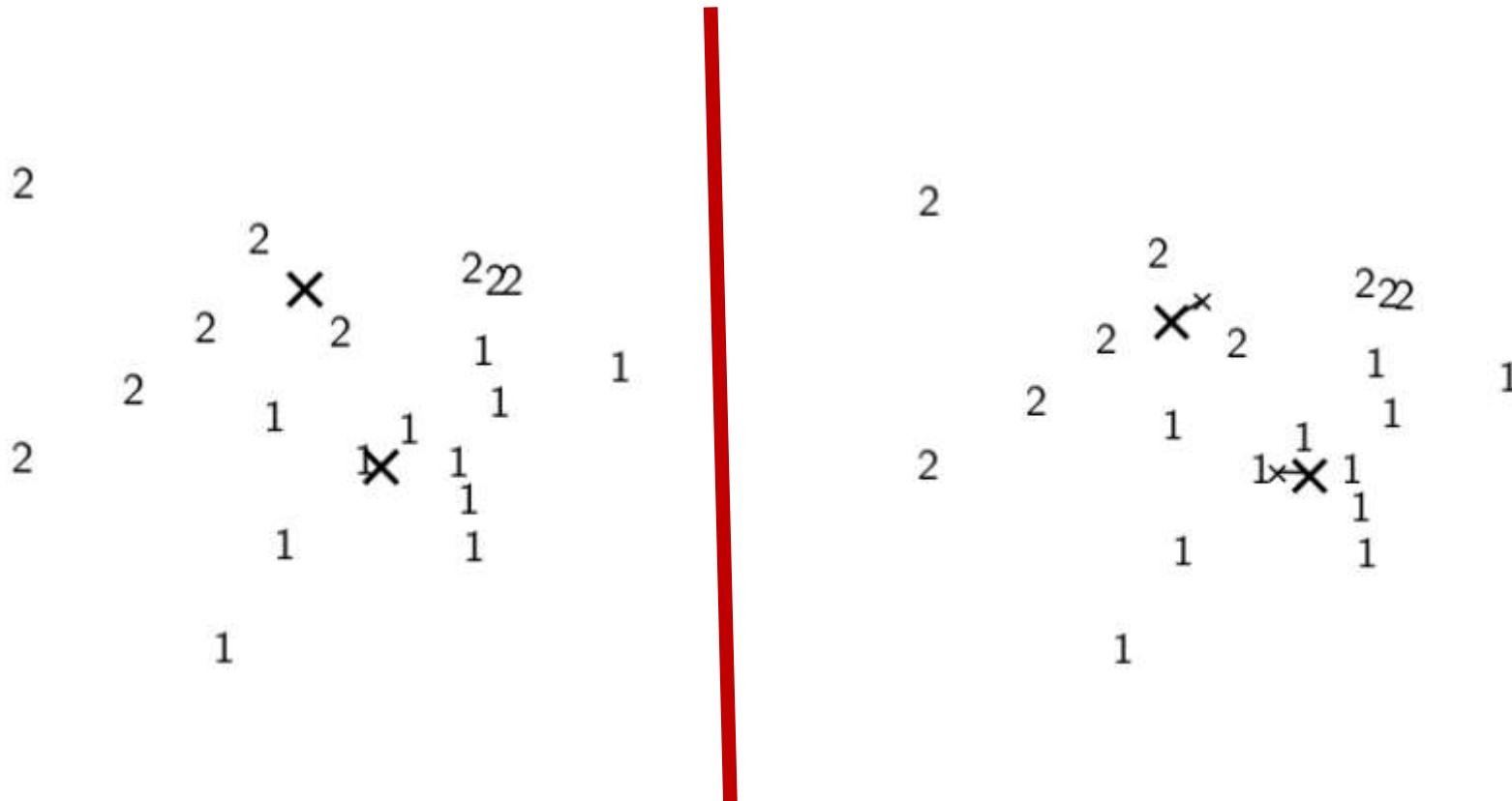


- Результат розподілу
 - Перераховуємо центроїди кластерів (уже не сильно змістилися!)

Ітерація 4. Розподіляємо кожну точку до найближчого центроїда



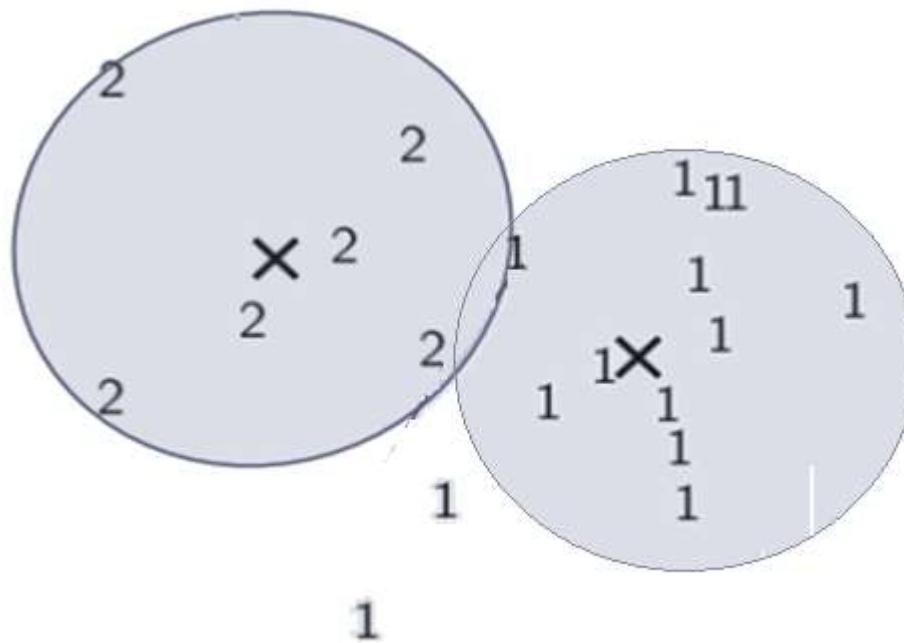
Ітерація 4



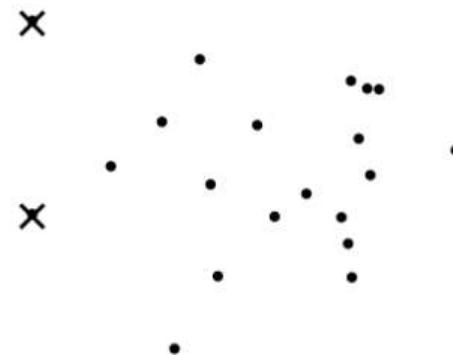
□ Результат розподілу

□ Перераховуємо
центроїди кластерів

Після 8-ї ітерації



А починали з



Обчислювальна складність методу k-середніх

- Обчислення відстані між двома об'єктами $O(M)$, де M - розмірність об'єктів (=векторів)
- Перерозподіл N об'єктів між K кластерами: $O(KN)$ обчислень відстаней, тобто $O(KNM)$
- Обчислення центроїдів: кожен об'єкт одного разу під'єднується до центроїду $O(NM)$
- Якщо у нас I ітерацій, отримуємо загальну складність: $O(IKNM)$

K-means pros and cons

□ Переваги:

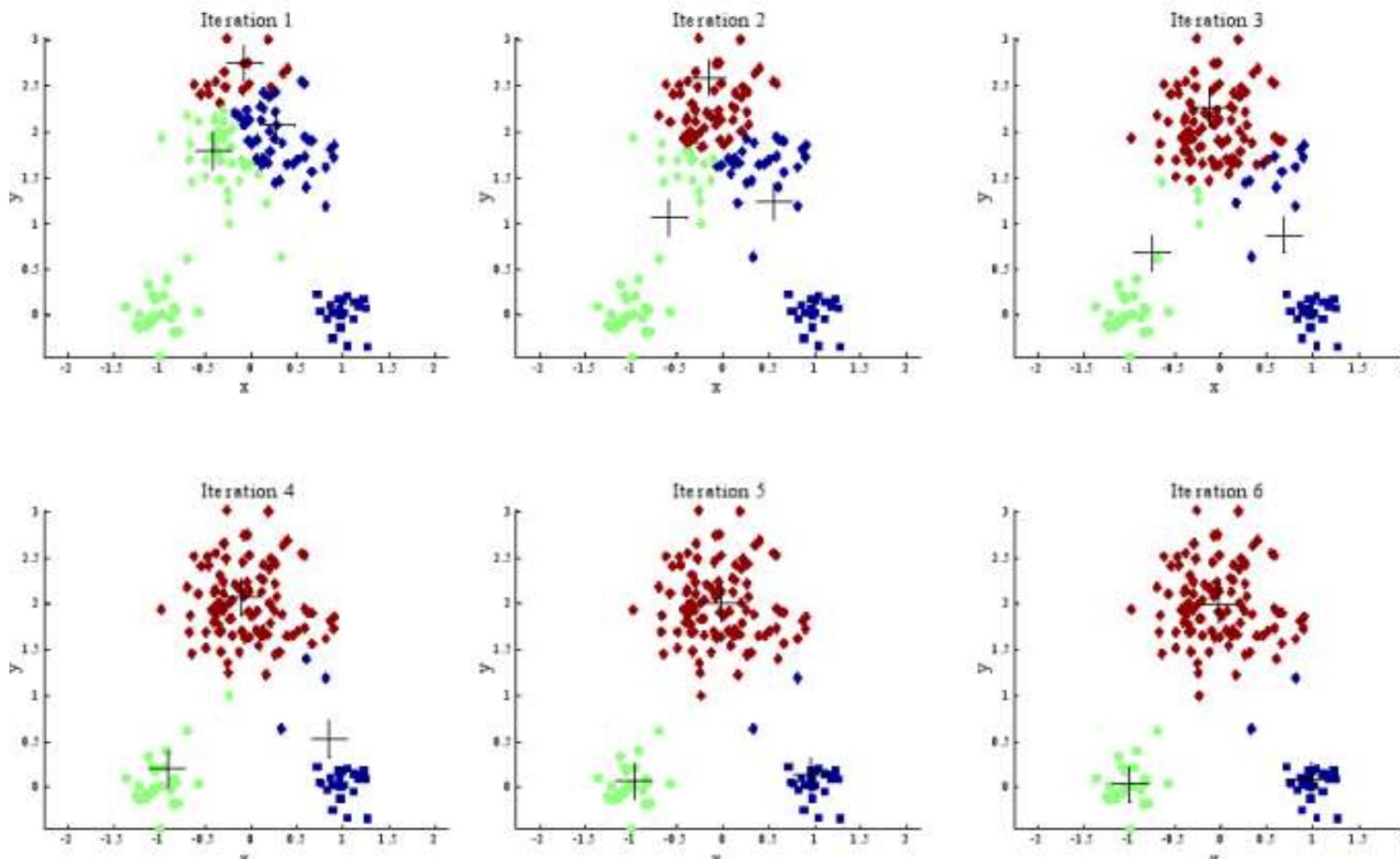
- загальна складність алгоритму: $O(IKNM)$. Зазвичай, кількість кластерів (K), розмірність об'єктів (M) і число ітерацій (I) \ll кількість об'єктів (N), тому метод є ефективним
- алгоритм є простим і зрозумілим, легко інтерпретується
- об'єкти автоматично розподіляються між кластерами

□ Недоліки:

- необхідно заздалегідь визначити кількість кластерів
- кластеризація може завершитись на локальному оптимумі, тому для високоякісного результата **необхідна багаторазова початкова ініціалізація**
- неможливо будувати кластери неопуклої форми (non-convex shape)
- чутливість до шумів та викидів, які завжди включаються до кластерів
- можливе застосування тільки для числових даних, тобто для яких визначено середнє

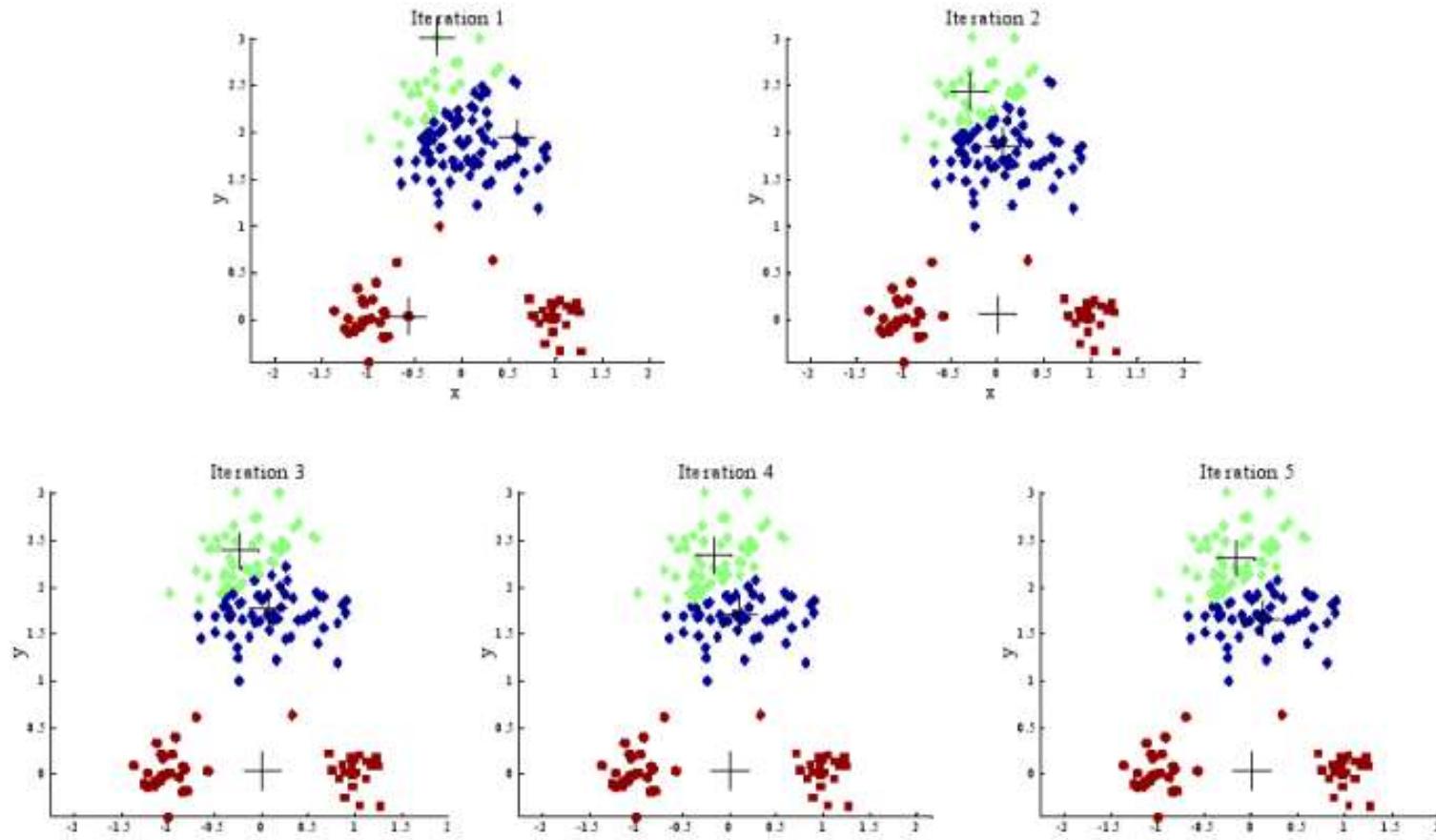
Ілюстрація недоліків алгоритму k-середніх. Локальний оптимум

- Гарна кластеризація, хоча, здається, вибір початкових центроїдів був невдалим



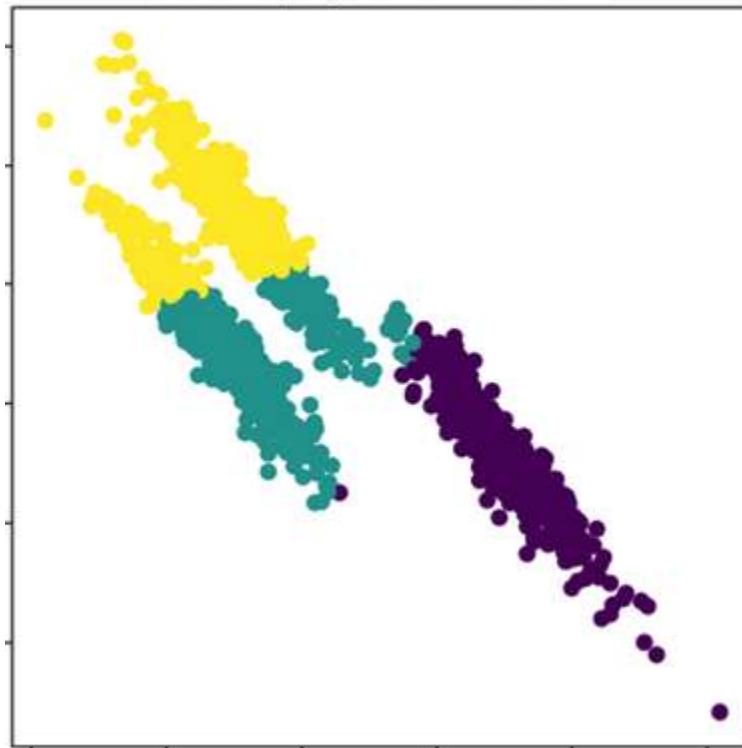
Ілюстрація недоліків алгоритму k-середніх. Локальний оптимум

- Погана кластеризація, хоча, здається, вибір початкових центроїдів був вдалим

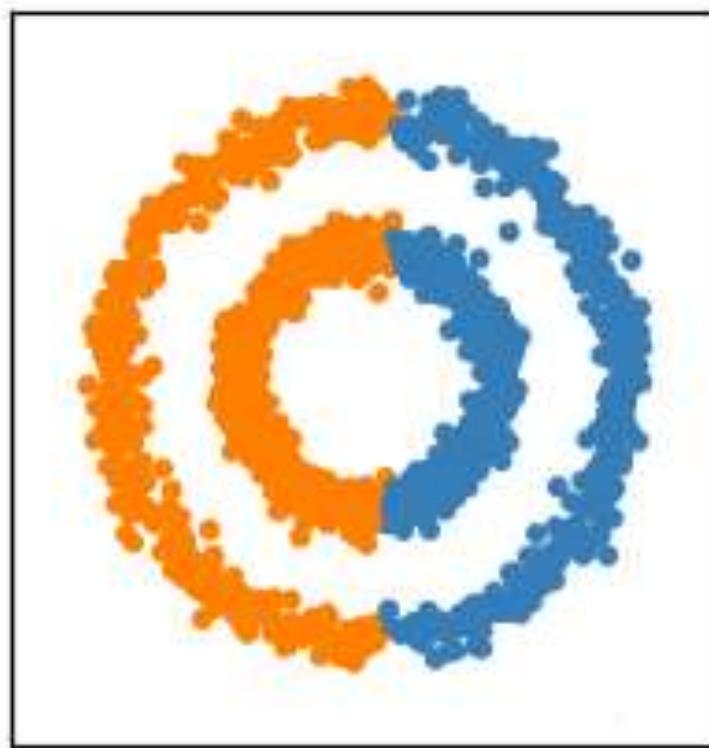


Ілюстрація недоліків алгоритму k-середніх. Кластери тільки круглої форми

- Anisotropicly distributed clusters



- Hierarchical (nested) clusters

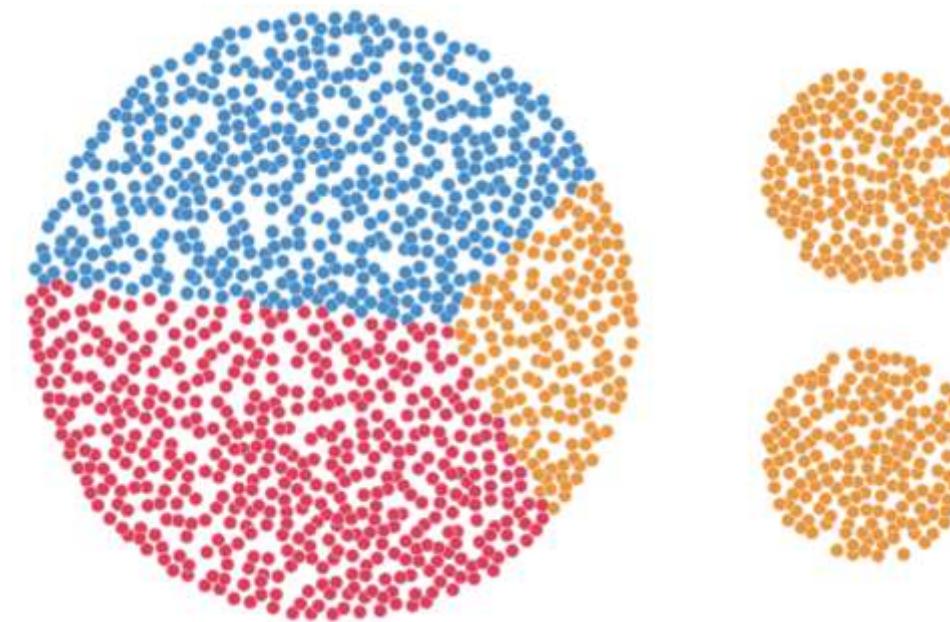
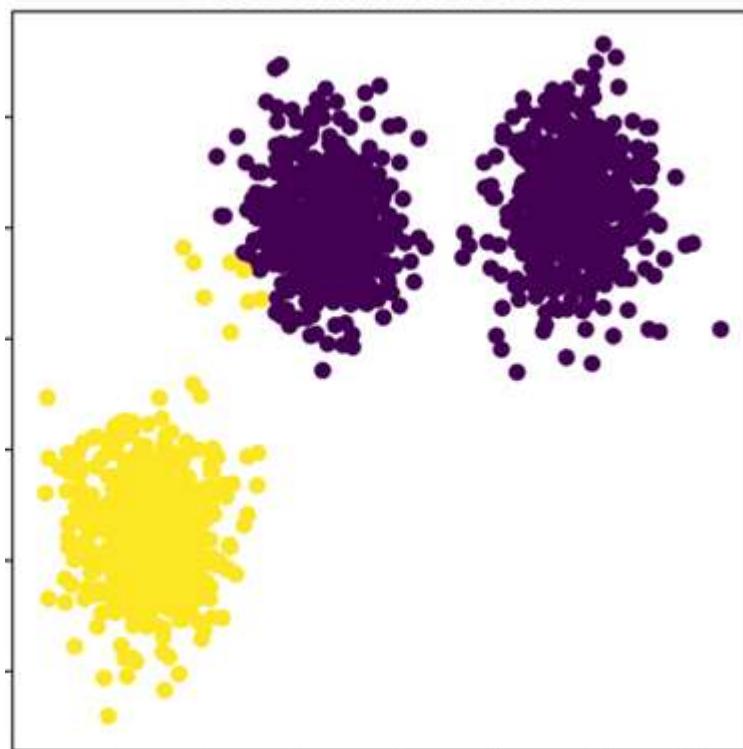


- This example is meant to illustrate situations where k-means will produce unintuitive and possibly unexpected clusters
- The input data does not conform to some implicit assumption that k-means makes and undesirable clusters are produced as a result

Ілюстрація недоліків алгоритму k-середніх.

Неправильний вибір кількості кластерів. Надання переваги близьким за розміром кластерам

- Incorrect number of clusters
- Splitting a large cluster by neighboring small ones



Зменшення впливу викидів: K-medoids & K-medians algorithms

- **K-means** algorithm is too sensitive to outliers
 - ❖ An object with an extremely large value may substantially distort the distribution of the data and thus the clustering result
- **K-medians** algorithm
 - ❖ Idea: use medians, instead of means, to describe the representative point
 - Mean of 1, 3, 5, 7, 9 is 5
 - Mean of 1, 3, 5, 7, 1009 is 205
 - Median of 1, 3, 5, 7, 1009 is 5
- **K-medoids** algorithm uses the most centrally located point in a cluster, as a representative point of the cluster

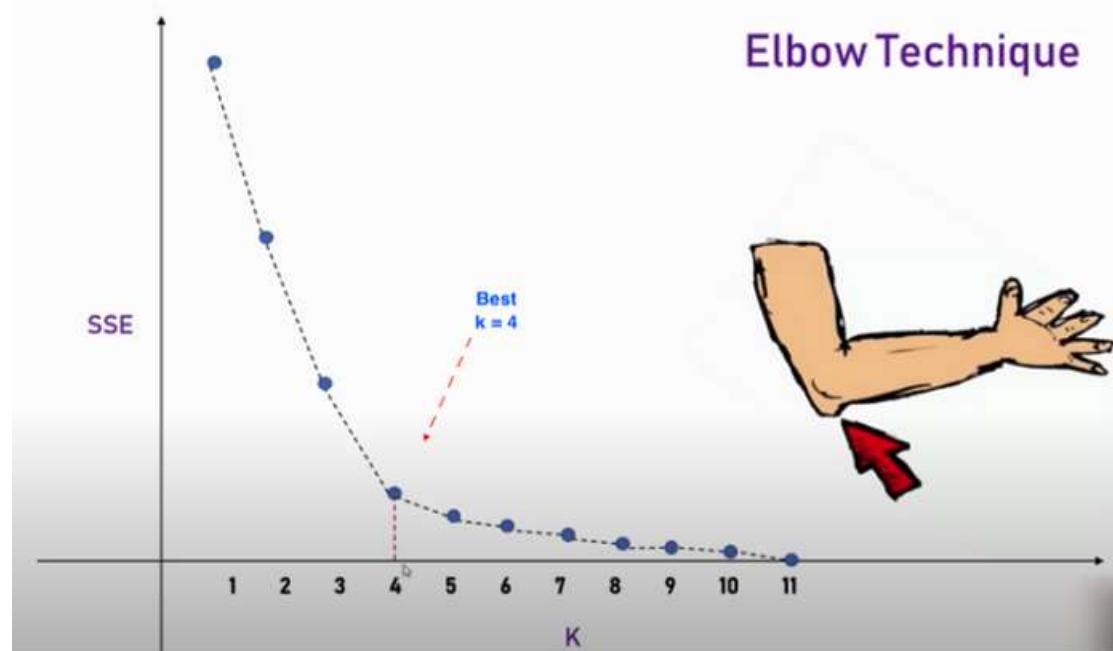
Як визначити кількість кластерів?

- Число кластерів K має бути визначено заздалегідь
 - ❖ Евристика: наприклад, знаючи характер об'єктів, припустимо «прийнятне» число кластерів
- Просто, щоб середня відстань від об'єктів до свого центроїда була мінімальною??
- Проста цільова функція для пошуку K
 - ❖ Починаємо з одного кластера ($K = 1$)
 - ❖ Продовжуємо додавати кластери (= збільшуємо K)
 - ❖ Нараховуємо штраф за кожен новий кластер
 - ❖ Балансуємо штрафи за нові кластери і вигоду від меншої середньої дистанції від центроїду
 - ❖ Вибираємо K з найкращим балансом

Як визначити кількість кластерів?

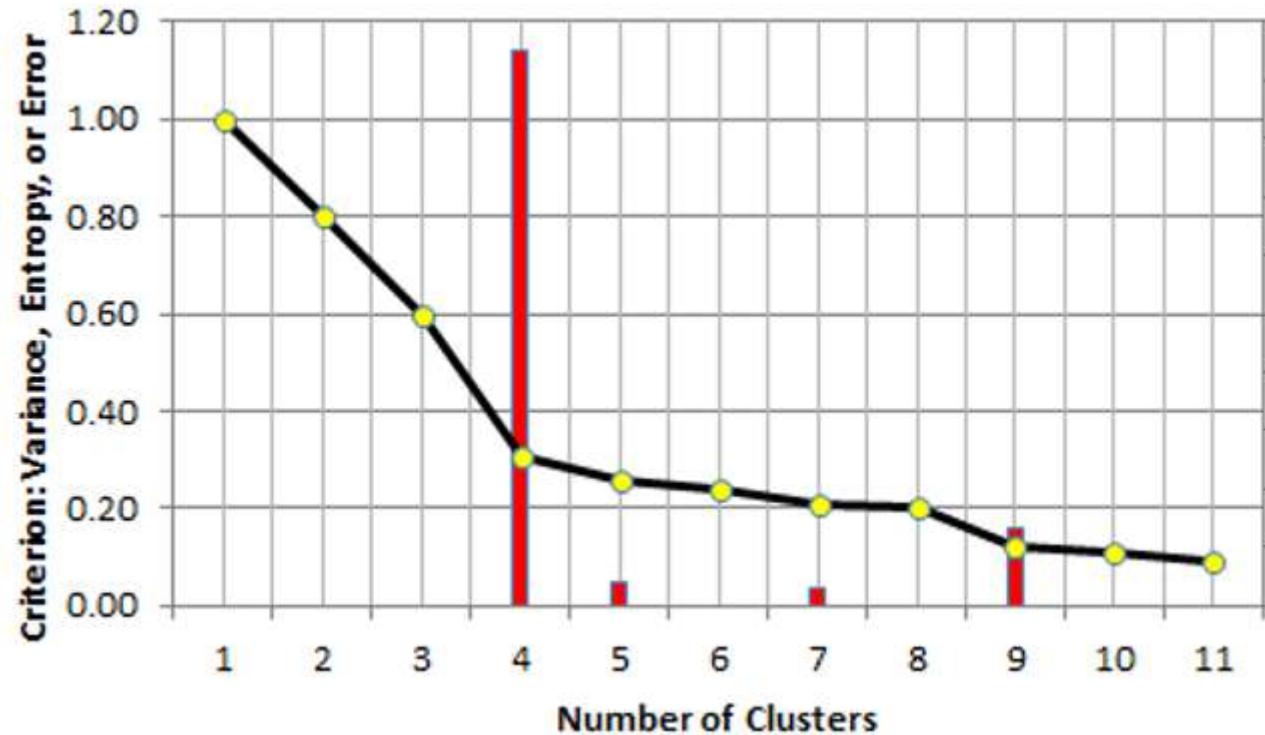
- Проста цільова функція для пошуку K
 - ❖ ...
 - ❖ Для даної кластеризації, визначте вартість штрафу для об'єкта як квадрат відстані до центроїда
 - ❖ Загальний штраф для кластера розрахуйте як суму штрафів всіх об'єктів у кластері $RSS(K)$ (Residual Sum of Squares)
 - ❖ Кожен кластер додатково штрафується фіксованим параметром λ
 - ❖ Цільова функція: мінімізувати $RSS(K) + K\lambda$
 - ❖ Залишається проблемою як знайти оптимальне значення **параметру регуляризації λ**

Пошук «ліктя» («elbow») на кривій



- У роботі Vincent Granville. How to Automatically Determine the Number of Clusters in your Data - and more (<https://www.datasciencecentral.com/profiles/blogs/how-to-automatically-determine-the-number-of-clusters-in-your-dat>) запропонований алгоритм пошуку такого «ліктя»

Пошук «ліктя» («elbow») на кривій



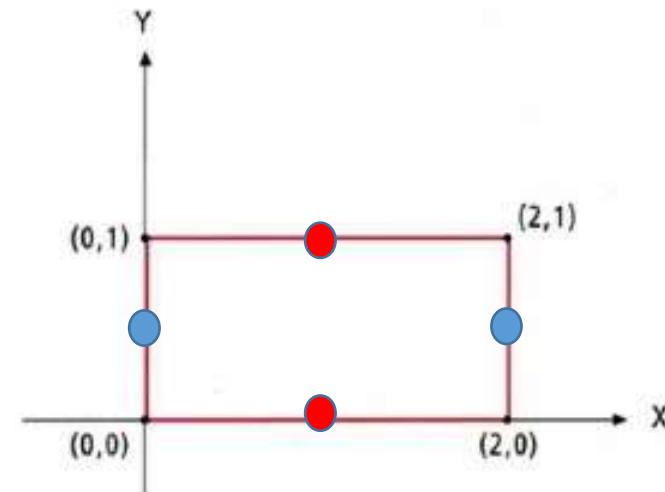
- У роботі Vincent Granville. How to Automatically Determine the Number of Clusters in your Data - and more (<https://www.datasciencecentral.com/profiles/blogs/how-to-automatically-determine-the-number-of-clusters-in-your-dat>)

запропонований алгоритм пошуку такого «ліктя»

- Виберіть кількість кластерів, при якій крива стає більш «пласкою» - як згин у лікті
- У даному випадку: 4
- Пакет kneed містить функцію для підрахунку «ліктя» («коліна»)

Метод k-means++

- Кластеризація методом k-середніх може завершитись на локальному оптимумі, тому для високоякісного результату необхідна **багаторазова початкова ініціалізація**
- Більше того, знайдене наближення може бути довільно поганим щодо цільової функції порівняно з оптимальною кластеризацією
 - - довільно поганий вибір центроїдів
 - - оптимальний вибір центроїдів



Метод k-means++

- Ідея методу:
 - перший центр кластера вибирається рівномірно випадковим чином з точок даних, які кластерізуються
 - після чого кожний наступний центр кластера вибирається з решти точок даних з ймовірністю, пропорційною квадрату відстані від точки до найближчого існуючого центру кластера
- k-means++ is an algorithm for choosing the initial values (or "seeds") for the k-means clustering algorithm
- The authors tested their method with real and synthetic datasets and obtained typically 2-fold improvements in speed, and for certain datasets, close to 1000-fold improvements in error
- Scikit-learn has a K-Means implementation that uses k-means++ by default

Arthur, D.; Vassilvitskii, S. (2007). "k-means++: the advantages of careful seeding" (PDF). Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms. Society for Industrial and Applied Mathematics Philadelphia, PA, USA. pp. 1027–1035

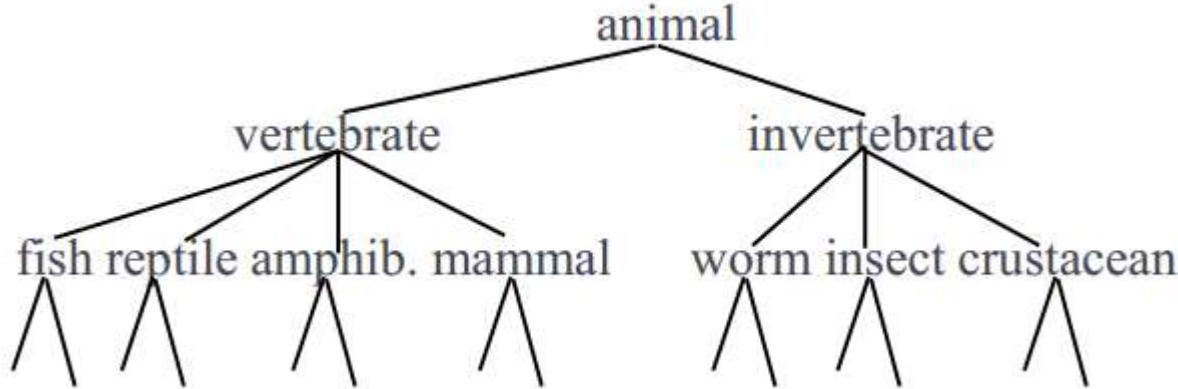
Fuzzy C-means cluster analysis (узагальнення методу k-середніх на нечіткий випадок)

- Метод запропонований Jim Bezdek в його PhD дисертації: J. C. Bezdek (1973). *Fuzzy Mathematics in Pattern Classification*, PhD Thesis, Cornell University, Ithaca, NY
- В нечіткій кластеризації кожен об'єкт належить різним кластерам з певною вагою у діапазоні від 0 (абсолютно не належить) до 1 (абсолютно належить)
- Але накладається обмеження, що сума вагів кожного об'єкта має дорівнювати 1
- Метод нечіткої кластеризації C-середніх має обмежене застосування через істотний недолік - неможливість коректного розбиття на кластери, в разі, коли вони мають різну дисперсію по різним вісям елементів (наприклад, кластер має форму еліпса)
- Даний недолік усунуто в [алгоритмі GMM \(Gaussian mixture models\)](#)

Метод GMM розглянемо в магістерському курсі «Машинне навчання»!

Ієрархічна кластеризація

- Завдання ієрархічної кластеризації – побудувати ієрархію кластерів



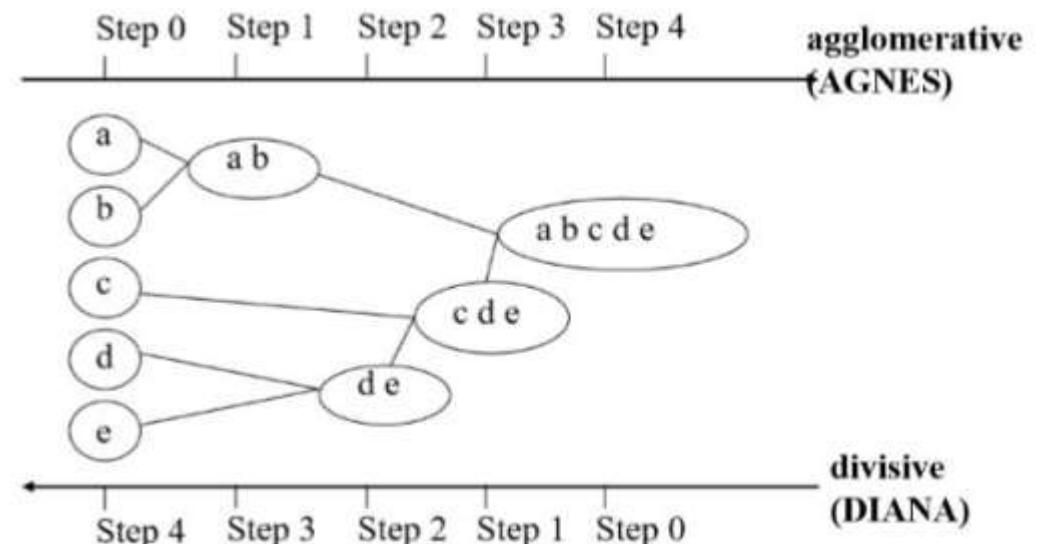
- Ієрархія будується автоматично:

- або згори-вниз: **агломеративні алгоритми** - AGNES (AGglomerative NESting), приклади: ROCK, CURE, CHAMELEON
- або знизу-вгору: **алгоритми розділення** - DIANA (DIvisive ANAlysis), приклади: BIRCH, MST

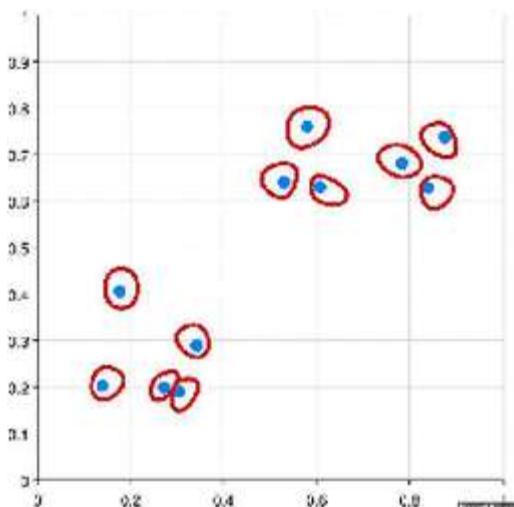
- In both agglomerative and divisive hierarchical clustering, users need to specify the desired number of clusters as a termination condition (when to stop merging)

Ієрархічна кластеризація

- На початку роботи агломеративного алгоритму кожна точка розглядається як кластер, потім алгоритм намагається об'єднати найближчі сусідні точки в один більший кластер і так далі, щоб зрештою об'єднати всі кластери в один великий кластер
- Алгоритм розділення спочатку розглядає всі точки множини як один кластер; на подальших кроках деякі кластери вищого рівня рекурсивно розщеплюються для побудови діаграми
- Ці підходи протилежні один одному

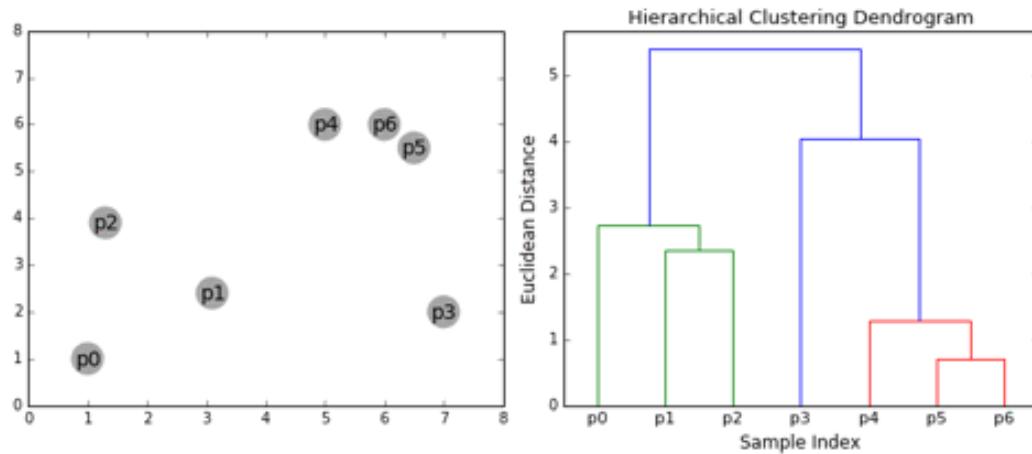


Ієрархічна агломеративна кластеризація



- Найвідоміший метод побудови знизу-вгору: ієрархічна агломеративна кластеризація
- Будує ієрархію у вигляді двійкового дерева
- Використовує **міру близькості** для визначення подібності двох кластерів
- Алгоритм:
 - ❖ Спочатку кожен об'єкт розглядається як окремий кластер
 - ❖ По черзі об'єднуємо два найбільш схожих кластера
 - ❖ До тих пір поки не залишиться один кластер
 - ❖ Історія об'єднань формує дерево ієрархії
 - ❖ Така історія зображується **дендограмою**

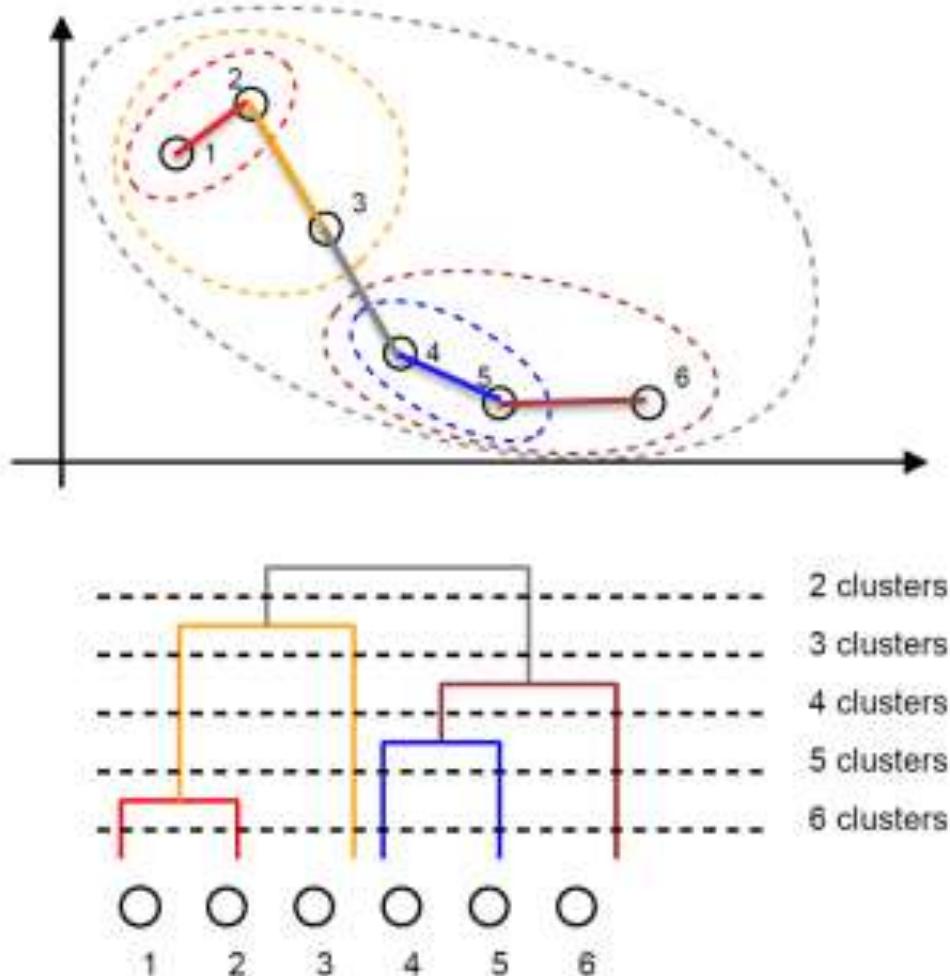
Ієрархічна агломеративна кластеризація



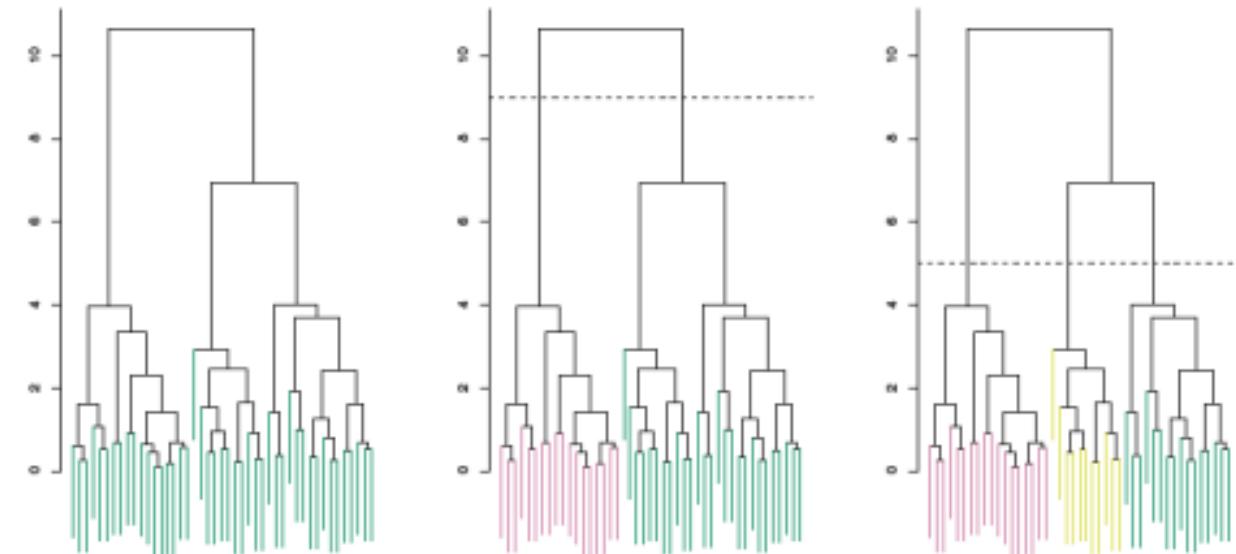
- A dendrogram is a type of tree diagram showing hierarchical relationships between different sets of data
- Dendrogram contains the memory of hierarchical clustering algorithm, so just by looking at the dendrogram you can tell how the cluster is formed

1. Distance between data points represents dissimilarities
2. Height of the blocks represents the distance between clusters

Денограма



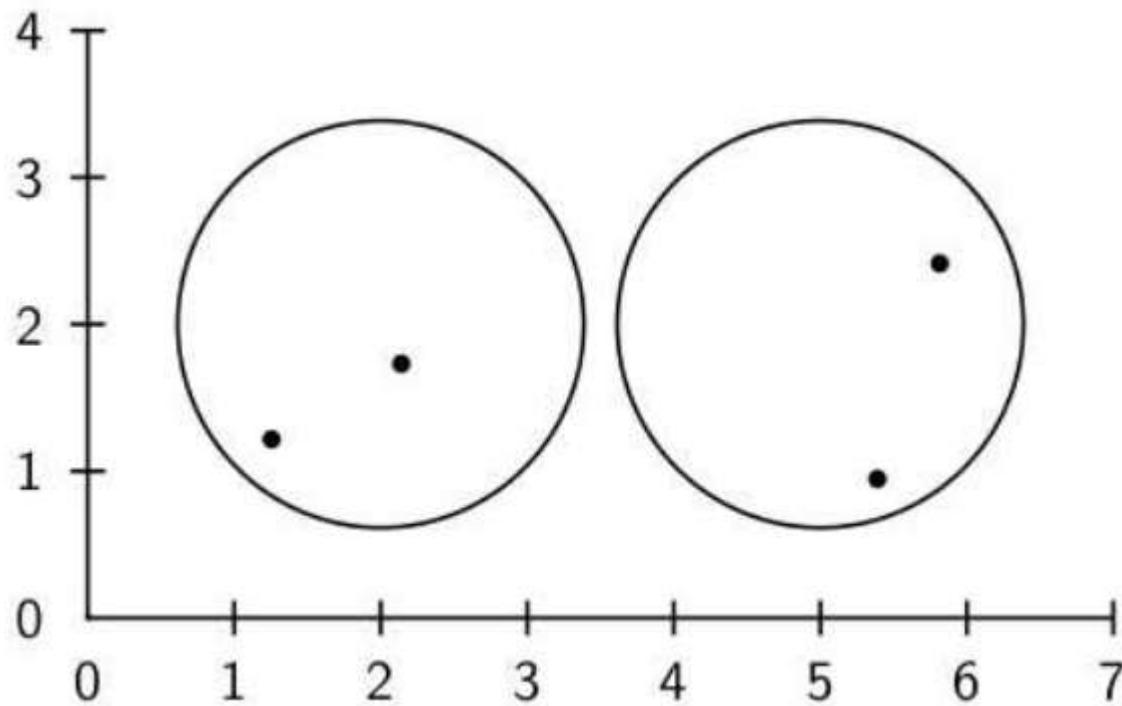
- Ми можемо відсісти денограму на будь-якому етапі для отримання пласкої кластеризації



Як обчислити близькість кластерів (Linkage Methods)?

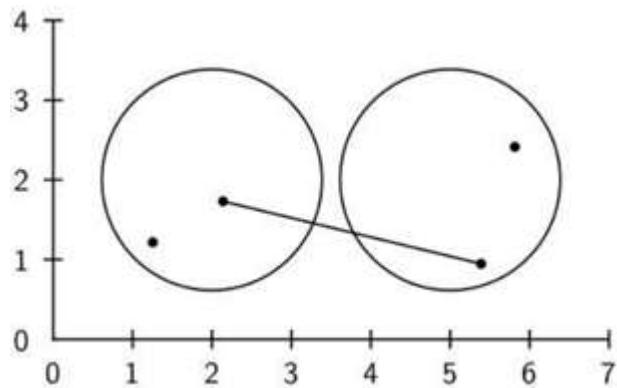
- Одиночний зв'язок (Single linkage):
 - ❖ максимальна близькість будь-яких двох об'єктів
 - Повний зв'язок (Complete linkage):
 - ❖ мінімальна близькість будь-яких двох об'єктів
 - Центроїдний зв'язок (Centroid linkage):
 - ❖ середня близькість всіх пар об'єктів (**не** включаючи пари об'єктів всередині кластерів)
 - ❖ рівносильно близькості центроїдів
 - Групове-середнє (Average linkage):
 - ❖ середня близькість всіх пар об'єктів, включаючи пари всередині кластерів
- 
- Попарна близькість
- Центроїдна близькість

Близькість кластерів на прикладі

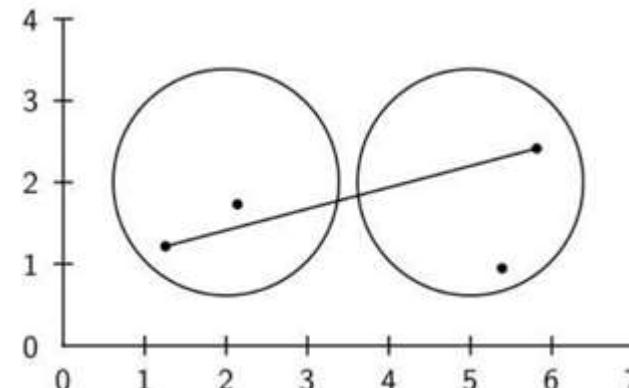


Близькість попарна

- Одиночний зв'язок (Single linkage):
максимальна близькість



- Повний зв'язок (Complete linkage): мінімальна близькість

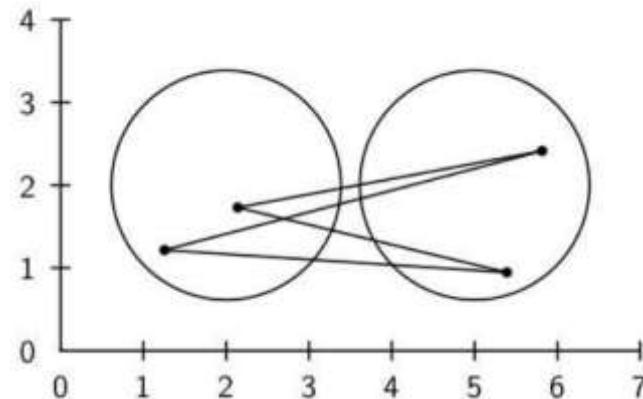


- подібність двох кластерів – це подібність між їх найбільш подібними членами (найближчий сусід)
- приділяється увага найближчим точкам, ігнорується структура кластера
- можливість будувати кластери неправильної форми
- такий вид зв'язку чутливий до даних з шумами та викидами

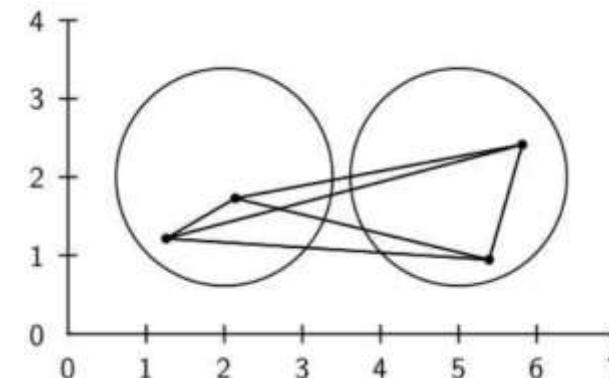
- подібність двох кластерів рахується як подібність їх найменш подібних членів
- два кластери, об'єднуючись, формують кластер з найменшим діаметром
- на виході – кластери компактної форми
- чутливий до викидів

Близькість між центроїдами

- Середня міжкластерна близькість (міжцентроїдна близькість)



- Групове-середнє: середня внутрішньокластерна близькість



- багато обчислень
- дозволяють об'єднувати в кластери дані без істотних змін через викиди та шуми

Обчислювальна складність ієрархічної кластеризації

- Обчислюємо близькість всіх $N \times N$ пар об'єктів
- Потім, на кожній ітерації:
 - ❖ Скануємо $O(N \times N)$ близькостей для знаходження максимальної
 - ❖ Об'єднуємо два кластери
 - ❖ Обчислюємо близькість між створеним кластером і всіма рештою
 - ❖ Всього $O(N)$ ітерацій, кожна вимагає $O(N \times N)$ сканувань
 - ❖ Загальна складність: $O(N^3)$
 - ❖ Існує більш раціональна модифікація алгоритму зі складністю $O(N^2)$

Пласка чи ієрархічна кластеризація?

- Пласка кластеризація значно швидше ($O(N \log N)$) і $O(N^2)$), добре підходить для великих обсягів даних
- Для стабільного передбачуваного результату використовують ієрархічну кластеризацію
- Ієрархічну кластеризацію також застосовують тоді, коли потрібна структура кластерів
- Іноді ієрархічна кластеризація використовується для визначення числа K , а в подальшому використовується пласка кластеризація

Density Based Spatial Clustering of Applications with Noise (DBSCAN)

- It's a clustering algorithm which was proposed in 1996. In 2014, the algorithm was awarded the 'Test of Time' award at the leading Data Mining conference, KDD.
- Why DBSCAN? Partitioning methods (like K-means) and hierarchical clustering work for finding spherical-shaped clusters or convex clusters. In other words, they are suitable only for compact and well-separated clusters. Moreover, they are also severely affected by the presence of noise and outliers in the data.
- Real life data may contain irregularities, like –
 - i) Clusters can have different shapes.
 - ii) Data may contain noise.
- Clusters are dense regions in the data space, separated by regions of the lower density of points. The DBSCAN algorithm is based on this intuitive notion of “clusters” and “noise”.
- The key idea is that for each point of a cluster, the neighborhood of a given radius has to contain at least a minimum number of points.

**Метод DBSCAN розглянемо
в магістерському курсі
«Машинне навчання»!**

Semi-Supervised Learning

- На сучасному етапі кластеризація часто виступає першим кроком при аналізі даних
- Після виділення схожих груп застосовуються інші методи, дляожної групи будується окрема модель (Semi-Supervised Learning)