ЛЕКЦІЯ № 9
КЛАСТЕРИЗАЦІЯ

Методи кластеризації:

K-Means. Агломеративна кластеризація, DBSCAN. Оцінка якості кластеризації

Завдання кластеризації

У задачі класифікації ми мали справу з відновленням відображення з багатьох об'єктів у кінцевий набір міток класів. При цьому класи були зафіксовані заздалегідь, тобто ми з самого початку приблизно розуміли, які об'єкти повинні ставитися до кожного з них, і ми мали в своєму розпорядженні навчальну вибірку з прикладами об'єктів і класів, до яких вони відносяться. У задачі кластеризації ми теж розбиваємо об'єкти на кінець безліч класів, але в нас немає ні навчальної вибірки, ні розуміння, якою буде природа цих класів. Те, що модель кластеризації якісь об'єкти визнала «схожими», віднісши до одного класу, буде новою інформацією, «відкриттям», зробленим цією моделлю. Навчальної вибірки у нас також не буде: ми ж не знаємо заздалегідь, що за класи вийдуть (а іноді й скільки їх буде). Таким чином, кластеризація – це завдання навчання без учителя. Через загальну схожість постановок завдань у літературі кластеризацію іноді називають unsupervised classification.

Методи кластеризації часто застосовують, коли фактично потрібно вирішити завдання класифікації, але навчальну вибірку зібрати важко (дорого чи довго). При цьому валідаційну вибірку для оцінки результатів кластеризації зібрати значно простіше, тому що для неї потрібно менше прикладів. При цьому варто пам'ятати, що точність роботи supervised-методів значно вища. Тому, якщо навчальну вибірку можна зібрати, краще вирішувати завдання класифікації, ніж завдання кластеризації.

Приклади задач кластеризації:

Наглядний приклад застосування методів кластеризації – аналіз геоданих. У мобільних програмах, що збирають геодані користувачів, часто потрібно зрозуміти, де саме користувач знаходився. GPS-координати відомі з деякою похибкою, користувач також зазвичай рухається, тому замість точного положення часто доводиться мати справу з роєм крапок. Становище посилюється, коли ми намагаємося аналізувати поведінку одразу тисяч людей у якійсь локації - наприклад, визначити, в яких місцях люди найчастіше сідають у таксі біля аеропорту. Може здатися, що достатньо подивитися на дані, і ми побачимо точно потрібні кластери.

Зображення нижче (Рис. 1) показує, як може виглядати ситуація всього для декількох користувачів: за даними GPS, таксі підбирають пасажирів і всередині будівлі аеропорту, і на злітній смузі, і там, де це відбувається насправді:

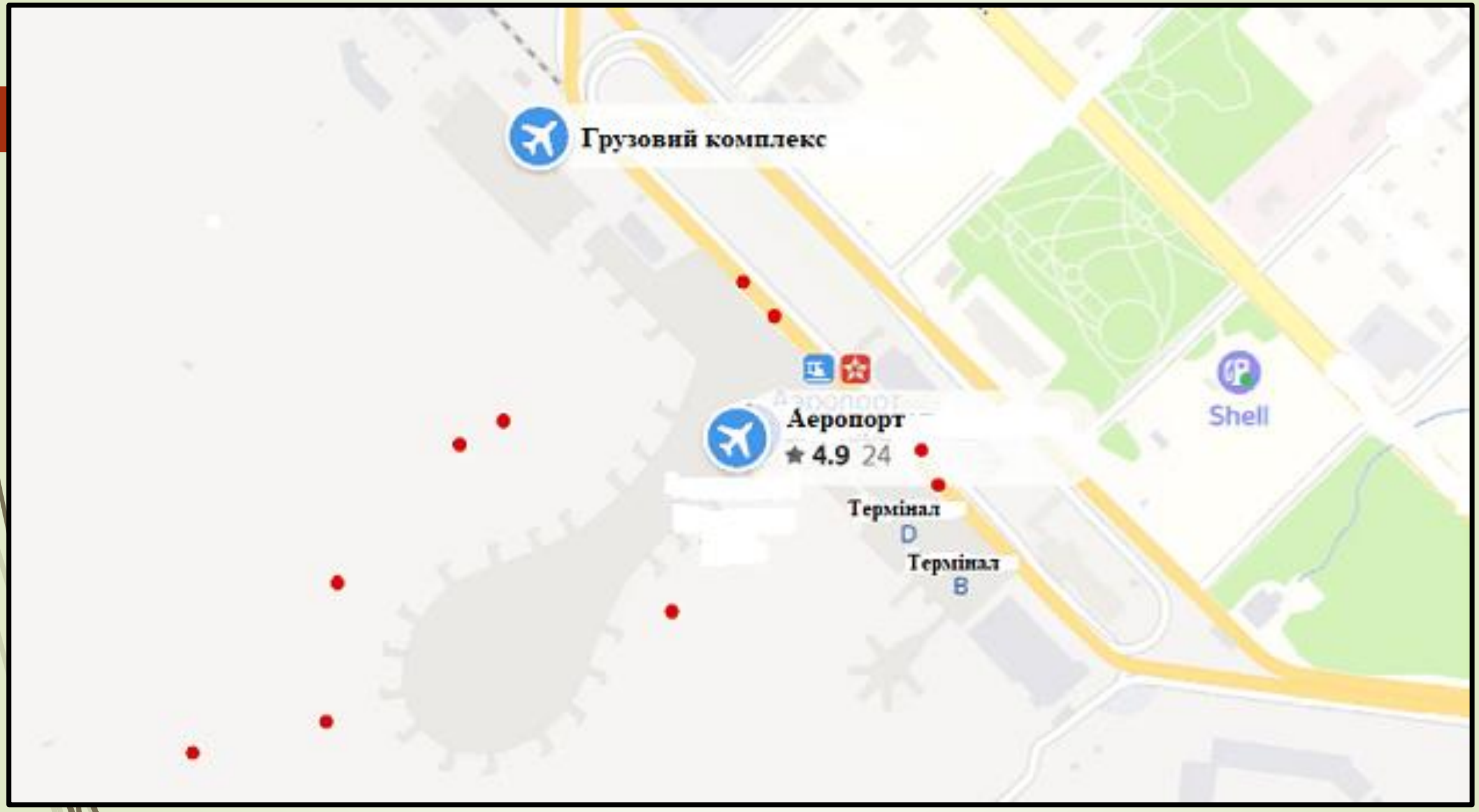

Грузовий комплекс
Аеропорт
4.9 24
Термінал D
Термінал B
Shell

Рис. 1

Найпростіші методи кластеризації за допомогою графів.

Можна наводити приклади не тільки про геоаналітику, але тема геоданих допоможе нам придумати пару найпростіших і найнаочніших методів кластеризації. Припустимо, що перед нами рій геокоординат і нам потрібно запропонувати за цими даними пікап-пойнти для таксі. Розберемо кілька очевидних методів.

Виділення компонент зв'язності.

Логічно спробувати об'єднати точки, які знаходяться одна від одної на відстані двох-трьох метрів, а потім просто вибрати найпопулярніші місця. Для цього збудуємо на відомих нам точках граф: точки, відстань між якими в межах трьох метрів, ми з'єднаємо ребрами. Виділимо у цьому графі компоненти зв'язності, вони й будуть нашими кластерами.

Цей спосіб має пару очевидних недоліків. По-перше, може знайтися скільки завгодно довгий ланцюжок точок, в якому сусідні відстоять один від одного на пару метрів, і весь він потрапить в одну компоненту зв'язності. У результаті наша відсічка за трьома метрами має дуже опосередковане відношення до діаметру кластерів, а самі кластери виходитимуть значно більше, ніж нам хотілося б. По-друге (і з першою проблемою це теж пов'язано), незрозуміло, як ми вибираємо максимальну відстань, при якій з'єднуємо точки ребром.

У цьому завданні ще можна пред'явити хоч якусь логіку, а от якби ми кластеризували не геомітки, а щось багатовимірне, наприклад електронні листи на їхню тематику, придумати відсічку було б вже складніше. Якщо наша мета – не лише вирішити практичне завдання, а й вигадати досить загальний метод кластеризації, зрозуміло, що нам хочеться розуміти, як підбирати параметри цього методу (у даному випадку умова додавання ребер до графа). Ці міркування можуть призвести до іншого рішення.

Мінімальне остове дерево.

Замість проведення ребер у графі, давайте їх видаляти. Побудуємо мінімальне кістякове дерево, рахуючи відстані між точками вагами ребер. Тоді, вилучивши N ребер з найбільшою вагою, ми отримаємо N+1 компоненту зв'язності, які, як і в минулому підході, вважатимемо кластерами. Відмінність у тому, що тепер нам потрібно задавати не відстань, коли проводиться ребро, а кількість кластерів. З одного боку, якщо ми вирішуємо задачу розрахунку пікап-пойнтів у якійсь конкретній локації (аеропорт, торговий центр, житловий будинок), нам може бути зрозуміло, скільки приблизно пікап-пойнтів хочемо отримати.

З іншого боку, навіть без локального розгляду можна зробити досить багато кластерів, щоб було з чого вибирати, але при цьому досить мало, щоб у кожен кластер потрапляла репрезентативна кількість точок. Аналогічна логіка буде справедлива і в багатьох інших завданнях кластеризації: кількість кластерів - досить загальний і досить інтерпретований параметр, щоб налаштовувати його вручну, тому в багатьох методах кластеризації кількість кластерів виступає як гіперпараметр.

Далі розглядатимемо деяке узагальнене завдання кластеризації без прив'язки до нашого прикладу з аналізом геоданих. Ми наведемо три найбільш популярні методи кластеризації: k-середніх, ієрархічну кластеризацію та DBSCAN, а потім розглянемо питання оцінки якості кластеризації.

Метод K середніх.

Мабуть, один із найпопулярніших методів кластеризації – це метод K-середніх (K-means). Основна ідея методу – ітеративне повторення двох кроків:

- розподіл об'єктів вибірки за кластерами;
- перелік центрів кластерів.

На початку роботи алгоритму вибираються *K* випадкових центрів у просторі ознак (Рис.2). Кожен об'єкт вибірки відносять до кластера, до центру якого об'єкт виявився ближче. Далі центри кластерів перераховують як середнє арифметичне векторів ознак усіх об'єктів, що увійшли до цього кластеру (Рис.3) (тобто центр мас кластера). Як тільки ми оновили центри кластерів, об'єкти знову перерозподіляються по них, а потім можна знову уточнити положення центрів. Процес триває доти, доки центри кластерів не перестануть змінюватися.

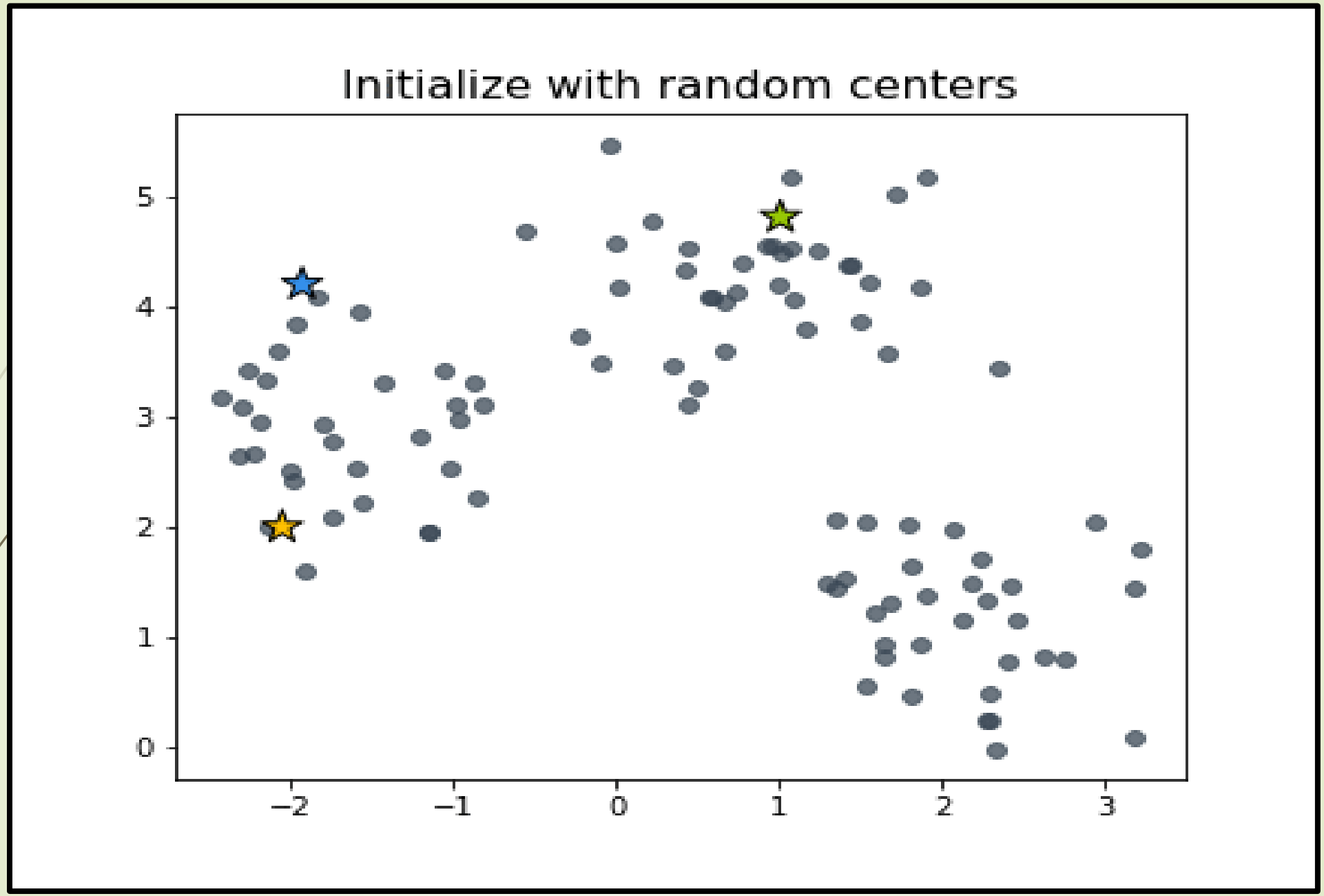


Рис.2 *K* випадкових центрів у просторі ознак

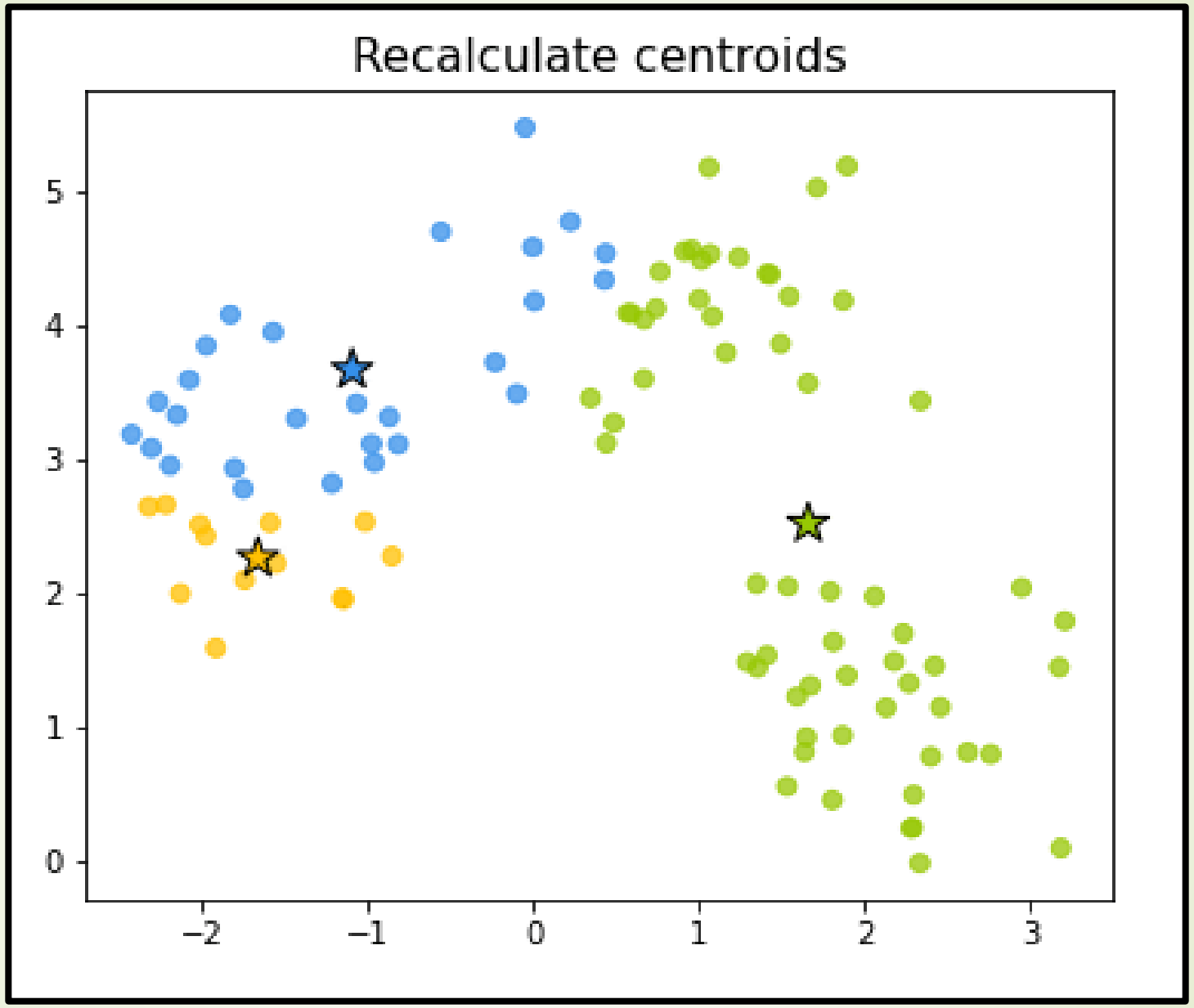


Рис.3 Перерахунок центрів кластерів.

Вибір початкового наближення.

Перше питання при виборі початкового положення центрів – як, вибираючи центри з деякого випадкового розподілу, не потрапити до простору ознак, де немає точок вибірки. Базове рішення – просто вибрати як центри якісь з об'єктів вибірки.

Друга потенційна проблема – купчасте розміщення центрів. І тут початкове становище з великою ймовірністю виявиться далеким від підсумкового становища центрів кластерів. Наприклад, для таких початкових положень центрів (Рис.4).

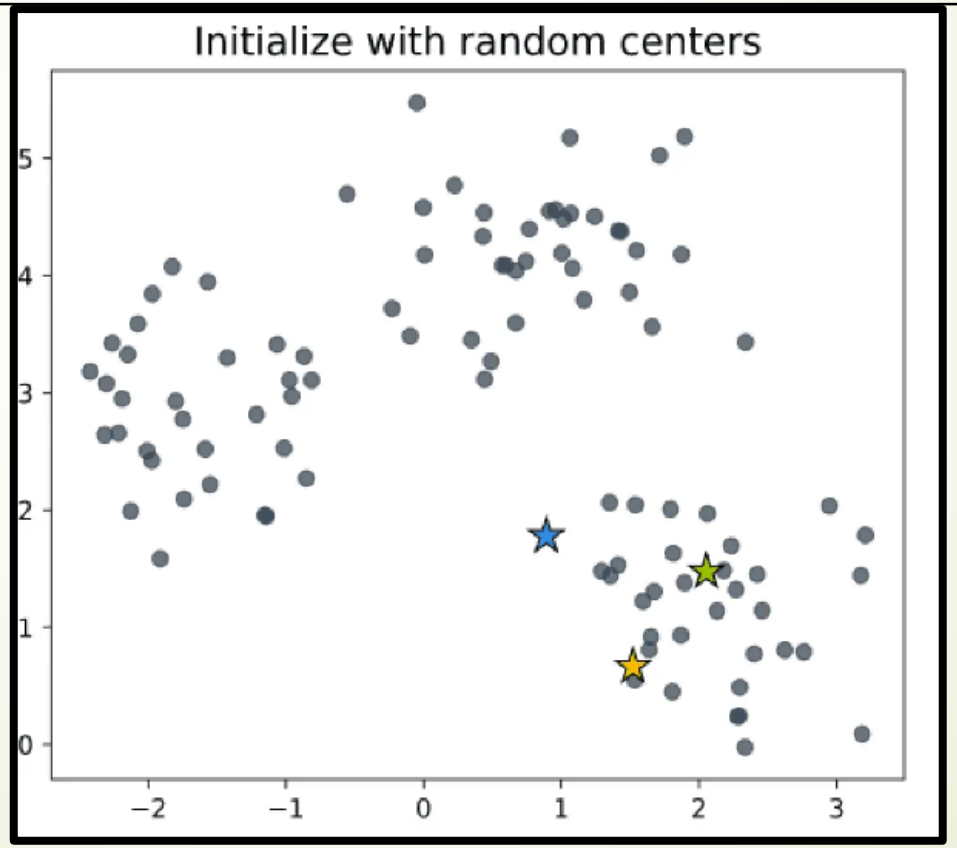


Рис.4

В такій ситуації, ми матимемо неправильну кластеризацію.

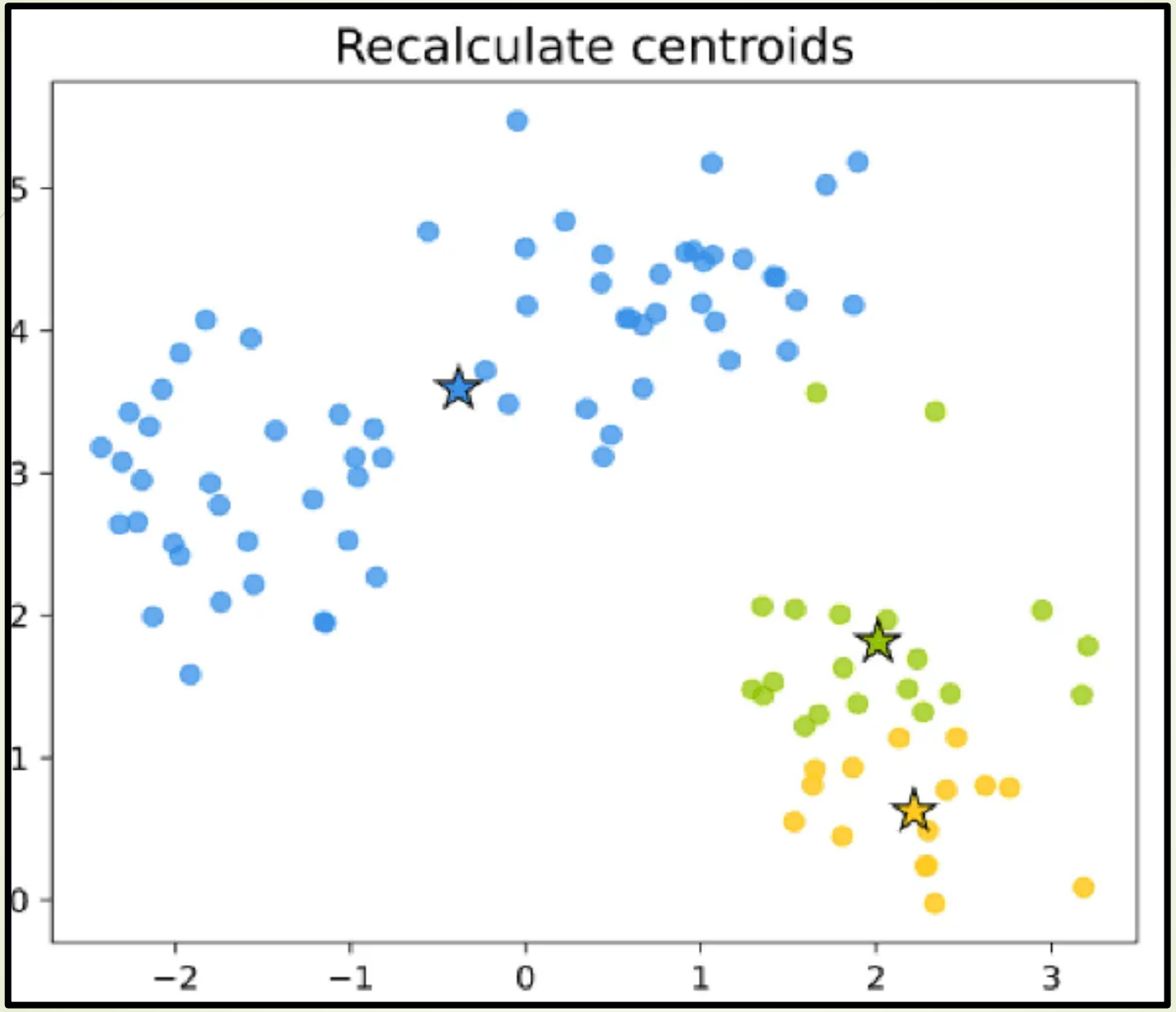


Рис.5

Щоб боротися із цим явищем, вигідно брати максимально віддалені один від одного центри.

На практиці працює наступна евристика:

Перший центр вибираємо випадково із рівномірного розподілу на точках вибірки; кожен наступний центр вибираємо з випадкового розподілу на об'єктах вибірки, у якому ймовірність вибрати об'єкт пропорційна квадрату відстані від нього до найближчого до нього центру кластера.

Модифікація K - means, яка використовує цю евристику для вибору початкових наближень, називається K - means++.

Вибір метрики.

Так як робота методу *K* - середніх складається з послідовного повторення до збіжності двох кроків, обґрунтованість застосування різних метрик (відстаней між точками, а не метрик якості або функцій близькості пов'язана з тим, чи вони «ламають» який-небудь з цих кроків чи ні).

Перший крок із віднесенням об'єктів до найближчих центрів не залежить від виду метрики.

Другий крок передбачає перерахунок центрів як середнього арифметичного точок, що входять у кластер, і ось тут буде підступ: до оптимальності вибору центрів у середньому арифметичному наводить саме евклідова метрика

Однак на практиці ніхто не заважає використовувати метод і без належного обґрунтування, тому можна експериментувати з будь-якими відстанями, з тим лише застереженням, що не буде жодних теоретичних гарантій, що метод спрацює. Найбільш поширена альтернатива евклідовій метриці - це косинусна міра близькості векторів (вона особливо популярна в задачах аналізу текстів) (1):

$$CosineSimilarity(\mu_k, x_i) = \frac{< \mu_k, x_i >}{\|\mu_k\|_2 \cdot \|x_i\|_2} \quad (1)$$

При її використанні варто не забувати, що косинусна міра - це функція близькості, а не відстань, тому чим більше її значення, тим ближче один до одного вектори.

Mini-batch K-means.

Якщо вважати K та розмірність простору ознак константами, обидва кроки алгоритму працюють за O(n), де n – кількість об'єктів навчальної вибірки. Звідси виникає ідея прискорення алгоритму. У mini-batch K-means ми не рахуємо кроки відразу на всій вибірці, а на кожній ітерації вибираємо випадкову підвибірку (міні-батч) та працюємо на ній. Якщо вихідна вибірка дуже велика, перехід до пакетної обробки не призводить до великої втрати якості, зате значно прискорює роботу алгоритму.

Зниження розмірності

З іншого боку, обчислення відстаней та середніх робиться за O(d) де d – розмірність простору ознак, так що інша ідея прискорення K-means – це попередньо знизити розмірність простору ознак (наприклад, ембеддинг). Особливо вдало ця ідея працює в задачах кластеризації текстів, коли K-means застосовують на ембеддінгах слів: виходить виграти не тільки в швидкості роботи, але і в інтерпретації результатів кластеризації.

До речі, сам алгоритм кластеризації теж можна використовувати як метод зниження розмірності. Якщо ви вирішуєте завдання навчання з учителем і простір ознак дуже різноманітний (тобто навчальна вибірка не дає вам достатньо статистики за такої великої кількості ознак), можна виконати кластеризацію об'єктів вибірки на 500 або 1000 кластерів і оперувати попаданням об'єктів у якийсь кластер як ознакою . Такий підхід називається квантуванням простору ознак (feature space quantization) і часто допомагає на практиці, коли потрібно укрупнити ознаки, додати їм інтерпретованості або навпаки знеособити.

приклад такого використання кластеризації – метод bag of visual words, що розширює bag of words з аналізу текстів на роботу із зображеннями. Ідея методу в тому, щоб будувати ознаковий опис зображень на основі фрагментів, що входять до нього: так, зображення з особами будуть містити фрагменти з носом, очима, ротом, а зображення з машинами – колеса, дзеркала, двері. Але проблема тут у тому, що нарізати такі фрагменти з навчальної вибірки та шукати точні збіги у нових прикладах зображень, які потрібно класифікувати, неможливо. У житті фрагменти зображень не повторюються інших зображеннях з попиксельной точністю. Вирішення цієї проблеми виявилося можливим за допомогою алгоритмів кластеризації (історично використовувався саме K-means): фрагменти зображень із навчальної вибірки кластеризували на 100 – 1000 кластерів («візуальних слів»), а проходячи по нових зображеннях, також нарізали їх на фрагменти та відносили до одному з цих кластерів.

При цьому, як нові зображення, і зображення з навчальної вибірки можна було описати кількістю входжень в них фрагментів із різних кластерів («візуальних слів»), як і у аналізі текстів описують текст кількістю входження до нього слів зі словника.

Параметри оптимізації K-means:

Обидва кроки алгоритму працюють на зменшення середнього квадрата евклідової відстані від об'єктів до центрів їх кластерів:

$$\Phi_0 = \frac{1}{nK} \sum_{k=1}^{K} \sum_{i=1}^{n} (\mu_k - x_i)^2 \mathbb{I}[a(x_i) = k] \qquad (2)$$

На кроці віднесення об'єктів до одного з кластерів ми вибираємо кластер з найближчим центроїдом, тобто мінімізуємо кожний доданок у Φ_0: всі потенційно великі складові ми занулюємо, а залишаємо ненульовими лише найменші з можливих (за умови фіксування центрів кластерів).

На кроці перерахунку центрів кластерів ми вибираємо центр таким чином, щоб при фіксованому наборі об'єктів, що належать до кластера, для всіх k мінімізувати вираз, що стоїть під сумою по k:

$$\sum_{i=1}^{n}(\mu_k - x_i)^2 \mathbb{I}[a(x_i) = k] \qquad (3)$$

Тут вже стає принципово, що ми визначаємо квадрат відстані як квадрат різниці векторів, тому що саме звідси при диференціюванні по μ_k та запису необхідної умови екстремуму виходить, що центри кластерів потрібно перераховувати як середні арифметичні
x_i, що належать кластеру. Цих міркувань, звісно, недостатньо, щоб стверджувати, що ми знайдемо мінімум Φ_0. Більше того, гарантії того, що ми знайдемо глобальний мінімум, взагалі, немає. Однак, витративши трохи більше зусиль, можна довести, що процес зійдеться в один з локальних мінімумів.
x_i, на вибірці фіксованого розміру є лише кінцева множина потенційних центрів кластерів. Якщо припустити, що в ході роботи *K-means* не зациклюється, звідси випливає, що рано чи пізно центри кластерів не зміняться на наступному кроці та алгоритм зійдеться. При цьому фактична збіжність, звичайно, відбувається задовго до повного перебору всіх можливих центрів кластерів.

Ієрархічна агломеративна кластеризація.

Інший класичний метод кластеризації – це ієрархічна кластеризація. Іноді додатково уточнюють: ієрархічна агломеративна кластеризація.

По-перше, є розподіл алгоритмів кластеризації на агломеративні (agglomerative) і дивізійні, або дивізійні (divisive). кластерів (зазвичай – з одного єдиного кластера) та поступово ділять на менші кластери.

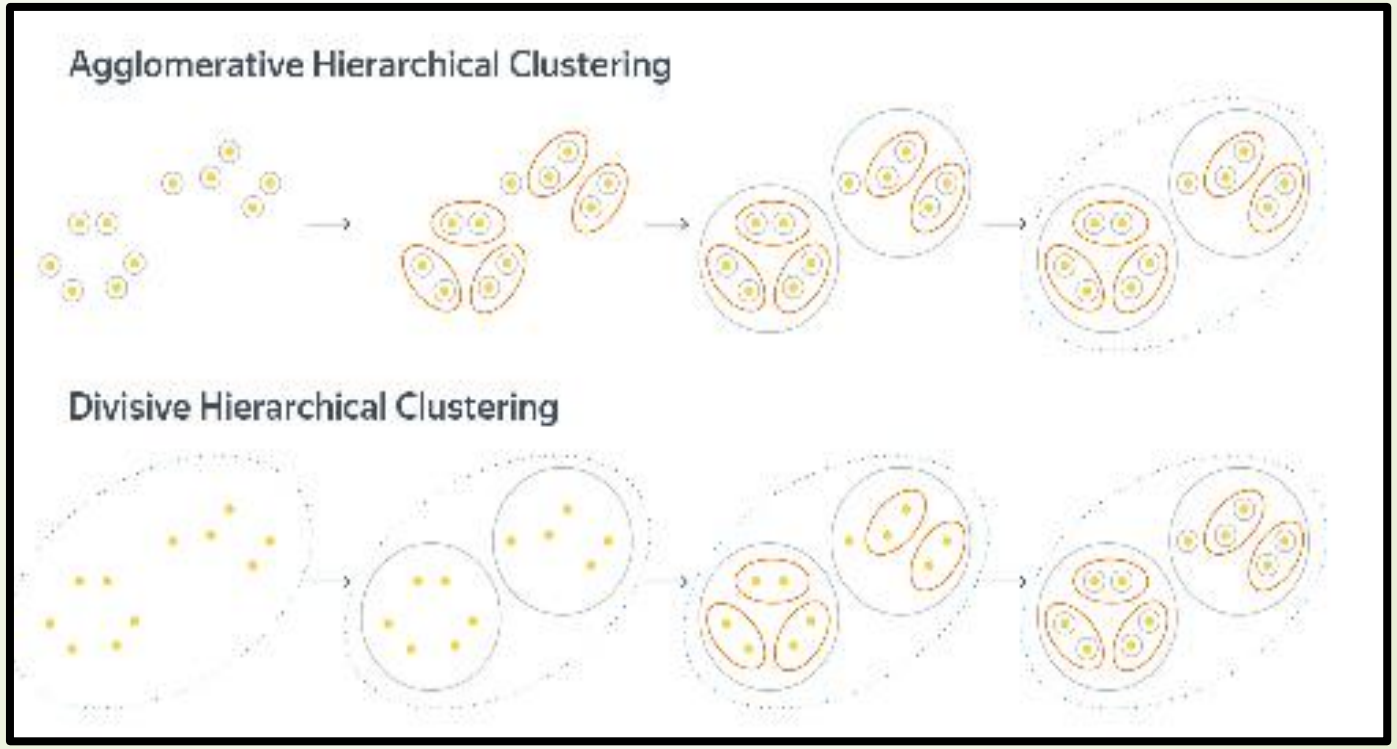


Рис. 6

По-друге, кластеризація буває, за аналогією з оргструктурою в організаціях, плоскою (коли всі кластери рівноправні і знаходяться на одному рівні кластеризації) та ієрархічною (коли кластери бувають вкладені один в одного і утворюють деревоподібну структуру).

У разі ієрархічної агломеративної кластеризації ми дійсно починатимемо з кластерів з одного об'єкта, поступово об'єднуючи їх, а вже послідовність цих об'єднань надасть структуру вкладеності кластерів. Навіть якщо в результаті ми використовуватимемо кластери з одного рівня, не заглиблюючись у жодну вкладеність, кластеризація все одно називається ієрархічною, оскільки ієрархія природним чином виникає в процесі роботи алгоритму.

Сам алгоритм виглядає дуже просто:

Створюємо стільки кластерів, скільки ми об'єктів у вибірці, кожен об'єкт – у своєму окремому кластері.

Повторюємо ітеративно злиття двох найближчих кластерів, доки виконається критерій зупинки.

Відстань до ієрархічної кластеризації

Як виміряти відстань між кластерами з одного об'єкта? Потрібно просто взяти відстань між цими об'єктами. Залишається питання, як узагальнити відстань між об'єктами до відстані між кластерами (якщо вони більше одного об'єкта). Традиційні рішення - брати середню відстань між об'єктами кластерів, мінімальну відстань чи максимальну. Якщо позначити кластери *U* та *V*, відстань між ними в цьому випадку можемо обчислювати за однією з формул:

$$d_{avg}(U,V) = \frac{1}{|U| \cdot |V|} \sum_{u \in U} \sum_{v \in V} \rho(u,v)$$

$$d_{min}(U,V) = \min_{(u,v) \in U \times V} \rho(u,v)$$

$$d_{max}(U,V) = \max_{(u,v) \in U \times V} \rho(u,v)$$

(4)

Формула відстані між кластерами, що використовується, – один з гіперпараметрів алгоритму. Крім наведених стандартних варіантів бувають і екзотичніші, наприклад відстань Уорда (Ward distance). У найбільш загальному вигляді способи завдання відстані між кластерами даються формулою Ланса – Вільямса (Lance – Williams).

Самі ж відстані між об'єктами можна задавати будь-якою метрикою, як евклідовою, так і манхеттенською відстанню або, наприклад, косинусною мірою (з тією лише поправкою, що це міра близькості, а не відстань).

Умови закінчення роботи алгоритму (критерії зупинки)

Як умови для завершення роботи алгоритму можемо вибрати або отримання потрібної кількості кластерів (кількість кластерів може бути гіперпараметром алгоритму), або виконання евристик на основі відстані між кластерами, що об'єднуються (наприклад, якщо відстань кластерів, що зливаються, значно зросла в порівнянні з минулою ітерацією). На практиці ж зазвичай кластеризацію проводять аж до одного кластера, що включає в себе всю вибірку, а потім аналізують ієрархічну структуру, що вийшла, за допомогою дендрограми.

DBSCAN.

Алгоритм DBSCAN розвиває ідею кластеризації за допомогою виділення зв'язкових компонентів.

Перш ніж перейти до побудови графа, введемо поняття густини об'єктів вибірки у просторі ознак. Щільність DBSCAN визначається в околиці кожного об'єкта вибірки

x_i як кількість інших точок вибірки у кулі $B(\varepsilon, x_i)$. Крім радіусу околиці як гіперпараметр алгоритму задається поріг N_0 за кількістю точок в околиці.

Далі всі об'єкти вибірки поділяються на три типи: внутрішні / основні точки (core points), граничні (border points) та шумові точки (noise points). До основних відносяться точки, в околиці яких більше N_0 об'єктів вибірки.

До граничних - точки, в околиці яких є основні, але загальна кількість точок в околиці менше N_0. Шумові називають точки, в околиці яких немає основних точок і в цілому міститься менше N_0 об'єктів вибірки. Алгоритм кластеризації виглядає так:

1.Шумові точки забираються з розгляду і не приписуються до жодного кластера.

2.Основні точки, які мають загальну околицю, з'єднуються ребром.

3.В отриманому графі виділяються компоненти зв'язності.

4.Кожна гранична точка відноситься до кластера, в який потрапила найближча до неї основна точка.

Зручність DBSCAN полягає в тому, що він сам визначає кількість кластерів (за модулем завдання інших гіперпараметрів ε і N_0), а також у тому, що метод успішно справляється навіть із досить складними формами кластерів. Кластери можуть мати вигляд протяжних стрічок або бути вкладеними один в одного як концентричні гіперсфери. На зображенні нижче показано приклад виділення кластерів досить складної форми за допомогою DBSCAN:

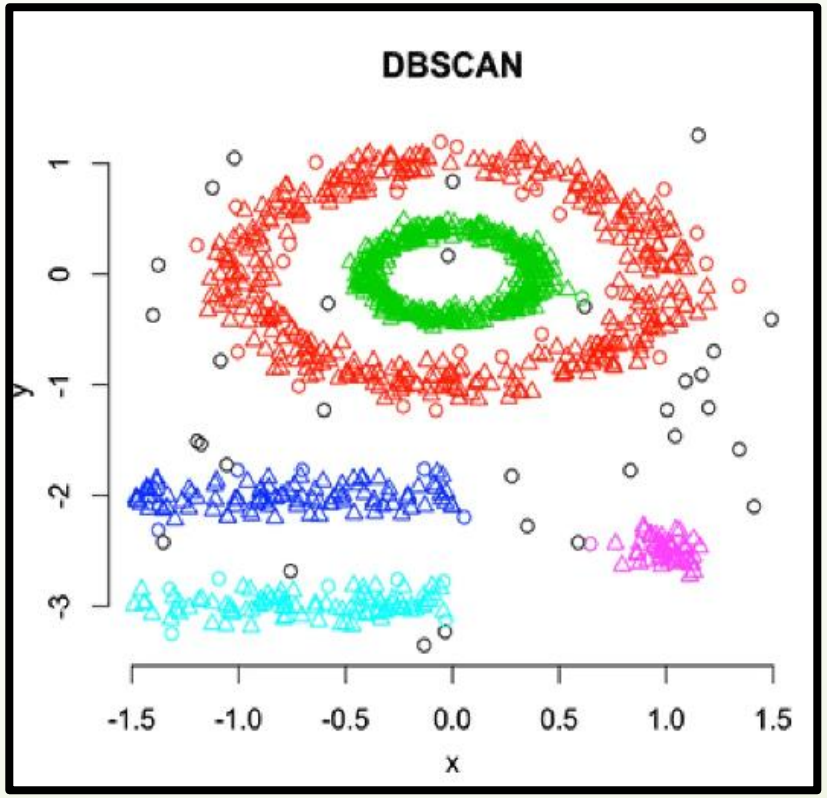


Рис.7

DBSCAN — один із найсильніших алгоритмів кластеризації, але працює він, як правило, помітно довше, ніж mini-batch K-means, до того ж дуже чутливий до розмірності простору ознак, тому використовується на практиці DBSCAN тільки тоді, коли встигає відпрацьовувати за прийнятне час.

Який метод кластеризації вибирати?

Якщо порівнювати частоту використання *K-means*, ієрархічної кластеризації і DBSCAN, то першому місці, безперечно, буде *K-means*, а друге місце ділитимуть ієрархічний підхід і DBSCAN. Ієрархічна кластеризація - більш відомий і простий у розумінні метод, ніж DBSCAN, але досить рідко відпрацьовує якісно. Часта проблема ієрархічної кластеризації - раннє утворення одного гігантського кластера та низки дуже невеликих, що призводить до сильної незбалансованості кількості об'єктів у підсумкових кластерах. У той же час DBSCAN - менш широко відомий підхід, але коли його можна застосувати, якість, як правило, виходить вищою, ніж у *K - means* або ієрархічної кластеризації.

Оцінка якості кластеризації.

Середня внутрішньокластерна відстань

$$F_0 = \frac{\sum_{i=1}^{n}\sum_{j=i}^{n}\rho(x_i, x_j)\mathbb{I}[a(x_i) = a(x_j)]}{\sum_{i=1}^{n}\sum_{j=i}^{n}\mathbb{I}[a(x_i) = a(x_j)]} \quad (5)$$

Сума відстаней між точками з одного й того кластера ділиться на кількість пар точок, що належать до одного кластера. У наведеній вище формулі пари виду (x_i, x_i) включені до розгляду, щоб уникнути невизначеності 0/0, у разі, коли у кожному кластері рівно по одному об'єкту. Однак іноді записують суми по $i < j$, просто визначаючи F_0 в даному випадку нулем.

Вирішуючи задачу кластеризації, ми хочемо по можливості отримувати якомога кучніші кластери, тобто мінімізувати F_0.

Якщо кластери мають центри μ_k, часто розглядається аналогічна метрика - середній квадрат внутрішньокластерної відстані:

$$\Phi_0 = \frac{1}{nK} \sum_{k=1}^{K} \sum_{i=1}^{n} \rho(\mu_k, x_i)^2 \mathbb{I}[a(x_i) = k] \qquad (6)$$