Міністерство освіти і науки України
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ "КИЄВО-МОГИЛЯНСЬКА АКАДЕМІЯ"
Кафедра математики факультету інформатики

Пошук сусідів в метричному просторі

Текстова частина до курсової роботи
за спеціальністю «Комп'ютерні науки та інформаційні технології» - 122

Керівник курсової роботи
доктор фізико-математичних наук, професор
Олійник Богдана Віталіївна

_____
(підпис)
“____” _____ 2020 року

Виконала студентка
Мазуркевич Віра Сергіївна
“____” _____ 2020 року

Київ 2020

Міністерство освіти і науки України

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «КИЄВО-МОГИЛЯНСЬКА АКАДЕМІЯ»

Кафедра математики факультету інформатики

ЗАТВЕРДЖУЮ

Зав. кафедри математики,

доктор фізико-математичних наук,

професор _____Б. В. Олійник

(підпис)

„____”_____2019 р.

ІНДИВІДУАЛЬНЕ ЗАВДАННЯ

на курсову роботу

студенту Мазуркевич Вірі Сергіївні факультету інформатики 4 курсу

ТЕМА Пошук сусідів в метричному просторі

Зміст ТЧ до курсової роботи:

Індивідуальне завдання

Календарний план

Вступ

1 Розділ 1. Властивості метричних просторів, конструювання та усунення

2 Розділ 2. Використання нерівностей трикутника

3 Розділ 3. Розмірності

4 Розділ 4. Розмірність та пошук найближчого сусіда

Висновки

Список літератури

Дата видачі „___” _____ 2019 р. Керівник _____

Завдання отримала _____

Тема: Пошук сусідів в метричному просторі

Календарний план виконання роботи:

№ п/п	Назва етапу курсового проекту (роботи)	Термін виконання етапу	Примітка
1.	Отримання завдання курсової роботи.	02.10.2019	
2.	Огляд літератури за темою роботи.	15.12.2019	
3.	Створення практичної частини роботи.	15.03.2020	
4.	Написання текстової частини роботи.	30.03.2020	
5.	Надання роботи керівнику для перевірки.	01.04.2020	
6.	Коригування роботи за результатами перевірки.	05.04.2020	
7.	Подання роботи на кафедру для перевірки на плагіат.	19.04.2020	
8.	Захист курсової роботи.	24.04.2020	

Студент Мазуркевич В. С.

Керівник Олійник Б. В.

“_____”_____

ЗМІСТ

ВСТУП

Однією з відомих проблем пошуку, що розглядаються для метричних просторів є проблема пошуку найближчого сусіда. Для заданого метричного простору (U, D) і заданої підмножини S множини точок U проблема пошуку найближчого сусіда полягає в тому, щоб побудувати структуру даних для S, щоб для точки q можна було швидко знайти точку $s \in S$, для якої відстань D(s, q) є мінімальною. В курсовій роботі розглядаються різні підходи до цієї проблеми. Підходи залежать як від властивостей метричного простору, зокрема від розмірності метричного простору, так і від потужності множини S. Крім того, є декілька варіацій задачі пошуку найближчого сусіда, які розглянуті нижче. Останніми роками було запропоновано декілька структур даних, які, очевидно, є зручними для просторів і їх підмножин з невеликої розмірності та (або) невеликою кількістю точок. Саме такі підходи розглядаються в курсовій роботі.

Припустимо, що U - це множина, а D – метрика, задана на U, тобто функція, яка приймає пари елементів з U та повертає невід'ємне дійсне число. Тоді, задавши підмножину $S \subset U$ з кількістю точок n, проблема пошуку найближчого сусіда полягає в тому, щоб побудувати структуру даних, яка для введеного запиту точки $q \in U$ знайде елемент $a \in S$ з $D(q, a) \leq D(q, x)$ для всіх $x \in S$. Назвемо елементи множини S точками, щоб відрізнити їх від інших елементів U, і говоритимемо, що відповідь a найближча до q в підмножині S. Іншими словами, якщо ми визначимо D (x, S) як min {D (x, s) | $s \in S$}, то шукаємо точку s таким чином, щоб виконувалась рівність D (q, s) = D (q, S).

Ця проблема пошуку вивчається давно і має багато назв у літературі. У одній з перших пропозиції щодо рішення, завдяки Дональду Кнуту була названа проблемою поштової доставки. В іншій ранній пропозиції [6] вона називалася найкращою відповідністю пошуку файла. У базі даних чи в пошуково-інформаційній літературі її можна назвати проблемою побудови індексу для пошуку подібності [5]. У літературі з теорії інформації ця проблема виникає як проблема побудови пристрою для кодування векторного квантування (обробки сигналів) [8, 9] . У літературі розпізнавання шаблонів (або статистики чи теорії навчання) це може бути названо проблемою побудови швидкого класифікатора найближчого сусіда [5, 10].

Як зазначалося раніше, проблема класифікації давно вирішується за допомогою пошуку найближчого сусіда. Точки в просторі відповідають наборам об'єктів, а координати точок кодують різні властивості об'єктів.

Кожен об'єкт також має "колір", наприклад, червоний або синій, що відповідає деякій додатковій властивості. Точки S - це навчальний набір, кожна точка якого із відомих кольорів. Класифікатор найближчого сусіда приймає на вхід точку невідомого кольору та повертає колір найближчої точки в навчальному наборі або колір більшості найближчих k точок. Це повернене значення - це передбачення справжнього кольору точки.

Таким чином, проблема пошуку найближчого сусіда виникає при визначенні найближчих точок у навчальному наборі. Шукати потрібно не лише найближчі точки, а лише їх кольори. Іноді можливо спростити проблему для отримання більш швидкого алгоритму.

Звернемось до іншої обчислювальної проблеми, тісно пов’язаної із пошуком найближчих сусідів, що виникають у застосуваннях. Одним із згаданих класифікацій є k-найближчих сусідів (k-NN): дається ціле число k та точка запиту q, і k-точок, найближчих до q. Тобто пошук найближчого сусіда є особливим випадком пошуку k-NN з пошуком k = 1. Ще одна пов'язана проблема - пошук діапазону відстаней: побудувати структуру даних таким чином, щоб задане значення відстані r та точка q, всі точки $p \in S$ з $D(q, p) \leq r$ можна швидко знайти. Якщо відстань до найближчого сусіда D (q, S) задана оракулом тоді відповідь діапазону на запит з параметром r = D (q, S) відповість на запит найближчого сусіда.

Апроксимуючі запити. Іноді може знадобитися не пошук найближчого сусіда, а лише сусіда (δ) - неподалік, тобто такого, відстань якого знаходиться в межах δ-коефіцієнта найближчої відстані, для деякої $\delta > 1$. Такі апроксимуючі запити найближчих сусідів цікавлять самі по собі, і можуть мати набагато швидші алгоритми, ніж алгоритми для запитів найближчого сусіда.

Зворотні запити. Іншою пов'язаною проблемою є побудова структури даних для запиту зворотного найближчого сусіда, де на вхід подається точка, але відповідь - це не точка, найближча до точки запиту q, а, скоріше, точки, які мають q як свого (другого) найближчого сусіда у $S \cup \{q\}$, тобто відповідь - це набір точок:

$$\{s \in S \mid D(s, q) \leq D(s, S \setminus \{s\})\}.$$

Як і у випадку пошуку найближчого сусіда, цю проблему також можна узагальнити за допомогою k та c: задана k та точка q, і всі точки, які мають q як k-ого найближчого сусіда, або дано $c > 0$, і всі точки, такі як q знаходяться в межах $1 + c$ найближчих. Ця проблема виникла як обчислювальне вузьке місце в керованих подіями астрофізичних моделюваннях, а також як поняття "вплив" у системах підтримки прийняття рішень та направлень. Це також

підіймається як підпроблема в побудові структур даних для запитів найближчих сусідів.

Пакетні запити. Існує кілька загальних проблем, які можуть бути вирішені за допомогою структур даних для пошуку найближчого сусіда або пошуку k-NN. Наприклад, проблема знаходження найближчої пари полягає у визначенні двох ділянок s_1 і s_2 таким чином, щоб D (s_1, s_2) = min {D (p_1, p_2) | p_1, p_2 ∈ S, $p_1 \neq$, p_2}. Це можна вирішити, застосувавши структуру даних для двох-найближчих сусідів до кожної точки по черзі. Аналогічно, проблема знаходження усіх-k-найближчий-сусідів (all-k-NN) повинна знайти для кожної точки s, k точок, найближчих до s. Відповідь на проблему найближчої пари легко знайти, використовуючи відповідь на проблему all-k-NN. Аналогічно відстань max-min:

$$max_i\ min_j\ \mathrm{D}(s_i, s_j),$$

яку було запропоновано як міру розходження, можна знайти серед результатів загальної кількості k-NN. Інтегральна проблема кореляції є аналогом запиту діапазону усіх найближчих сусідів: задається значення r > 0, знайдуться всі пари точок на відстані r одна від одної.

Біхроматичні проблеми. Окрім згаданої вище «хроматичної» проблеми класифікації найближчих сусідів, є ще один клас проблем - біхроматичний. На вхід подається два набори S_0 і S_1, і найближча пара точок є по одному з кожного набору.

Розділ 1. Властивості метричних просторів, конструювання та усунення

1.1 Метричні простори

Функція відстані або метрика D метричного простору (U, D) задовольняє наступним умовам для всіх x, y, z ∈ U:

1. невід'ємність: D (x, y) ≥ 0;

2. мала дистанція об'єкта до самого себе: D (x, x) = 0;

3. ізоляція: x ≠ y означає D (x, y) > 0;

4. симетрія: D (x, y) = D (y, x);

5. нерівність трикутника: D (x, z) ≤ D (x, y) + D (y, z).

1.2 Усунення в метриках відстані

Дуже багато випадків пошуку найближчого сусіда, природно, задаються за допомогою функцій відстані, які можуть не бути метриками, але визначаються певними *пов'язаними* метриками. Якщо будь-яка з умов 3–5 не виконується, а інші виконуються, існує природна пов'язана функція, яка є метрикою, описаною далі.

Якщо умова 3, ізоляція, не виконується, тоді (U, D) називається псевдометрикою. Розділення U на класи еквівалентності на основі D, де x і y еквівалентні тоді і тільки тоді, коли D (x, y) = 0. З природною відстані D ([x], [y]) = D (x, y) на класи, результат - метричний простір.

Якщо умова 4, симетрія, не виконується, тоді (U, D) - квазіметрика. Пов'язана метрика Ď(x, y) := (D(x, y) + D(y, x)) /2 буде задовольняти симетрію і таким чином утворює метричний простір.

Якщо умова 5, нерівність трикутника, не виконується: це симетричний або позитивно-зважений неорієнтований граф. Пов'язану метрику можна знайти за допомогою найкоротших шляхів: нехай

$$\hat{D}(x, y) := \inf \sum_i D(z_i, z_{i+1}),$$

де точна верхня границя (infimum) приймається за всі послідовності в U

$$x = z_1, z_2, \ldots, z_N = y,$$

для всіх N > 1. Ď задовольняє умові нерівності трикутника і є метрикою. Це найкоротший шлях у графі, вершинами якого є точки, є метрикою графа.

Усунення нерівності трикутника часто використовують в іншому напрямку: з огляду на скінченний метричний простір (U, D), знайдеться граф з набором вершин U і з кількома ребрами, таким чином, що отримана метрика графа є хорошим наближенням до оригінальної метрики D. Такі графи називають остовими.

Інше можливе усунення нерівності трикутника є використання $\check{D}(x, y) := D(x, y)^1/w$; для досить великого w, Ď задовольняє нерівність трикутника. Якщо тільки w = ∞ буде задовільнятись, то рівномірна метрика (D (x, y) = 1, якщо x ≠ y), є результуючим Ď. В іншому випадку, при w < ∞ цей підхід може представляти інтерес, оскільки він зберігає нерівності між відстанями, так, що найближчий сусід у D такий самий як у Ď. Для скінченних просторів $max_{x,y,z \in U, x \neq y} log_2 (D(x, z)/D(x/y))$ буде досить великим, наприклад; ця кількість обмежена розкидом. Перетворення зберігає рангову позицію: якщо у далі від x, ніж z, це також буде усунено. Таким чином, міри відстані, які не виконують нерівність трикутника, можуть бути перетворені в метрики, зі збереженими відповідями на запити найближчих сусідів. З іншого боку, це перетворення згладжує відстань, що може зробити кластери менш чіткими та погіршити деякі алгоритми пошуку.

Виправлення квазіметрик, наведених вище, є обчислювально тривіально. Псевдометрика насправді не потребує «виправлення» для пошуку найближчого сусіда: потрібно лише пам’ятати, що відповідь є представником класу еквівалентності та можливістю того, що окремі точки мають нульову відстань. Виправлення нерівності трикутника може бути складно застосувати в контексті пошуку найближчого сусіда. Однак графові метрики є поняттям, що викликає значний інтерес та важливе поняття в оптимізації. З огляду на довільну функцію відстані, яка має властивість D (x, x) = 0, асоційовану метрику можна було б знайти, використовуючи найкоротші шляхи, щоб отримати функцію, яка задовольняє нерівність трикутника, а потім

усереднювати для забезпечення симетрії та остаточно групувати в класи еквівалентності для досягнення ізоляції.

1.3 Метричні побудови простору

Один базовий метричний простір для будь-якого заданого набору U - це рівномірна метрика, де для всіх x, y ∈ U, D (x, y) = 1, якщо x ≠ y, і D (x, x) = 0. Інший базовий простір - це множина дійсних чисел $\Re$, з відстанню | x - y | для x, y ∈ $\Re$. Більше того, метричні простори можна побудувати з інших просторів. Припустимо, (U, D) - це метричний простір, як і деякі (U_1, D_1). . . (U_d, D_d).

Підпростори. Очевидно, що будь-який $(U`, D`)$, де $U` \subset$ U і $D`$ є D обмеженим до $U` \times U`$ є метричним простором.

Добутки. Нехай Ũ - векторний добуток $U_1 \times U_2 \times \ldots U_d$, тобто d-кортежі над U_i. Для деякого значення p з $1 \leq p \leq \infty$ визначимо Ď таким чином: для x, y ∈ Ũ, нехай:

$$\check{D}(x, y) \coloneqq \left(\sum_i \check{D}(x_i, y_i)^p\right)^{1/p}$$

буде метрикою добутку. Коли всі $U_i = \Re$ і D_i (x, y) = | x - y |, то виходить $\Re^d$ з мірою відстані l_p, так Ď (x, y) = $||x - y||_p$. Коли p = d = 2, це просто евклідова площина. Коли p = 1 та всі D_i – це рівномірна метрика, результат - відстань Хеммінга.

Стрічки. Нехай U^* позначає рядки над U. Припустимо, Ď є мірою відстані U^*, яка визначається наступним чином: коли вилучення або додавання одного символу з x дає y, то Ď (x, y) = 1; при заміні символу a в x на символ b виходить y, тоді Ď (x, y) = D (a, b). Тоді (U^*, Ď) - це симетричний, і найкоротший шлях «відновлення» називається редагуванням рядка або відстанню Левенштейна. Іншими словами, відстань редагування рядка між x, y ∈ U^* є мінімальною послідовністю витрат операцій видалення, вставки або заміни символу для отримання y з x. Якщо видалення та вставка мають нескінченну вартість, то це відстань Хеммінга на рядках. Цей простір може бути використане для корекції орфографії та порівняння генетичних послідовностей.

Підмножини. Відстань Хаусдорфа між підмножинами U дорівнює

$$\check{D}(S, T) := \min \{D`(S, T), D`(T, S)\},$$

де

$$D`(S, T) := \frac{sup}{s\in S} \frac{\inf}{t\in T} D(s,t).$$

Така відстань може бути використана для геометричних фігур. (Технічно це лише псевдометрія, але це показник для всіх закритих обмежених підмножин.) Ще одна поширена відстань між підмножинами – це

$$D(S,T) := \frac{\inf}{s\in S,\ t\in T} D(s,t).$$

Коли U має міру μ, вивчено відстань μ (AΔB), де AΔB - симетрична різниця A і B; ця метрика узагальнює відстань Хеммінга.

Невід'ємні комбінації. Припустимо, U_i всі рівні в множині U, але D_i є різними. Дано $\alpha_1 \dots \alpha_d$ при $\alpha_1 \geq 0$, визначимо Ď на Ď (x, y):=$\sum_i \alpha_i D_i$ (x, y). Тоді (U, Ď) - це метрика, невід'ємна комбінація оригіналів. Іншими словами, набір метрик на U є близьким під невід'ємне поєднання і утворює конус.

Метричні перетворення. Якщо f - реальна величина функції невід'ємних дійсних чисел, а f (0) = 0, а f (z) - монотонно зростаюча і угнута при z ≥ 0, то Ď (x, y): = f (D(x, y)) - є метрикою. Наприклад, якщо f вдвічі більше, $f'(z) \geq 0$, і $f''(z) \leq 0$ при z ≥ 0, то f є монотонно зростаючою і вигнутою. Однією з таких функцій є f (z): = z^{ϵ}, для будь-якої заданого є з 0 < є < 1. Нову метрику D(x, y) $^{\epsilon}$ іноді перетворення сніжинки або силою перетворення оригіналу. Функція з f (z) = z / (1 + z) також задовольняє заданим умовам і дає міру обмеженої відстані.

Перетворення Штейнгауза. Якщо (U, D) - метричний простір, та a ∈ U, тоді (U, Ď) також є метричним простором, де

$$\check{D}(x,y) := \frac{2D(x,y)}{D(x,a)+ D(y,a)+ D(x,y).}$$

що називається перетворенням Штейнгауза.

Коли це перетворення застосовується до відстані D (A, B) = μ (AΔB), та з a, яке буде нульовим набором Ф, результатом є

$$\check{D}(A,B) = \frac{2\mu(A\Delta B)}{\mu(A\Delta\Phi)+\mu(B\Delta\Phi)+\mu(A\Delta B)} = \frac{2\mu(A\Delta B)}{\mu(A)+ \mu(B)+ \mu(A\Delta B)} = \frac{2\mu(A\Delta B)}{2\mu(A\cup B)} = \frac{\mu(A\Delta B)}{\mu(A\cup B)}$$

що називається відстанню Штейнгауза.

Розділ 2 Використання нерівностей трикутника

2.1 Межі нерівності трикутника

Властивості метричних просторів дозволяють отримати деякі основні спостереження, які можуть дати значно швидші алгоритми для пошуку найближчого сусіда. Вони випливають із нерівності трикутника, яка дозволяє межі на відстані, яку ми, можливо, не обчислили, скажімо, D (q, s), отримати з двох відстаней, які ми, можливо, вже знаємо, скажімо, D (q, p) і D (p, s). Ось наступні властивості, які випливають з цього.

Лема 2.1 Для q, s, p ∈ U, будь-яке значення r та будь-яке P ⊂ U:

1. | D (p, q) – D (p, s) | ≤ D (q, s) ≤ D (q, p) + D (p, s);

2. D (q, s) ≥ D_p (q, s): = $\max_{p \in P}$ | D (p, q) − D (p, s) |;

3. якщо D (p, s) > D (p, q) + r або D (p, s) < D (p, q) − r, тоді D (q, s) > r;

4. якщо D (p, s) ≥ 2D (p, q), тоді D (q, s) ≥ D (q, p).

Доведення: Застосовуючи нерівність трикутника трьома можливими способами:

$$D(q, s) \leq D(q, p) + D(p, s)$$

$$D(p, s) \leq D(p, q) + D(q, s)$$

$$D(q, p) \leq D(q, s) + D(s, p)$$

Перша з них - верхня межа для D (q, s) в (1), а дві інших означають нижню межу (1). Твердження (2) випливає з (1), дві частини твердження (3) випливають із останніх двох нерівностей, а твердження (4) випливає з твердження (3) з r = D (p, q).

Якщо точки в U представлені вектором їх відстаней до P, тоді D_P(q, s) - відстань l_∞ між q і s. Оскільки D_P (q, s) ≤ D (q, s), відображення від початкового (U, D) до ($\Re^{|P|}$, D_∞) вважається стискаючим; такі стискаючі відображення можуть бути корисними при пошуку відстані: якщо проблема відображається у векторному поданні, то відповідь на запит відповідає супермножині відповіді у початковому просторі.

2.2 Алгоритм Орчарда, AESA, Метричні дерева

Вищеописані межі від нерівності трикутника дозволяють уникнути обчислення відстані від точки q до багатьох точок, накладаючи межі на їх відстані, які дозволяють виключити точки як найближчі.

2.2.1 Алгоритм Орчарда

Розглянемо таку просту схему. Для кожної точки p створимо список точок у порядку збільшення відстані до p.

Щоб визначити найближчу точку до точки q, виберемо деяку точку c в якості початкового кандидата для найближчої точки. Обчислимо D (c, q) та пройдемося по списку до c, обчислюючи відстані до точок у списку. Якщо деяка точка s ближче до q, ніж c, встановимо c := s. Тепер повторимо ту саму процедуру, використовуючи новий c та його список. Припустимо, деякий такий список переходить до точки s з D (c, s) > 2D (c, q). Тоді за лемою 3.1 (4), c - найближча точка: будь-яка залишився в списку точка для c повинна бути далі від q, ніж c.

Цей алгоритм простий і швидкий, однак йому потрібна попередня обробка $\Omega(n^2)$, що робить його непридатним для великих баз даних. Також, для зберігання треба теж $\Omega(n^2)$. Для багатьох застосувань це неможливо. Однак для цільового застосування векторного квантування ці витрати можуть бути прийнятними.

Алгоритм Орчарда є екземпляром методу "обходу", і тому його можна прискорити, використовуючи техніку пропуску списку.

Одне покращення алгоритму Орчарда полягає в тому, щоб переконатися, що відстань від q до будь-якого даної точки обчислюється лише один раз на запит; один із способів зробити це - зберегти біт позначки для кожної точки, який спочатку дорівнює нулю для всіх точок. Коли обчислюється відстань до точки, біт позначки встановлюється одиницею, і точка заноситься у зв'язний список. Якщо точка розглядається для обчислення відстані, якщо біт позначки встановлений одиницею, точку можна ігнорувати: вона не може бути ближча за поточну точку. Після запиту зв'язний список проходить, і біти позначок встановлюються на нуль для

точок у списку. Така схема дозволяє підтримувати розмітки бітів у часі, пропорційному кількості оцінок відстані.

Метод відшарування. Для полегшення навантаження на зберігання іншою схемою є збереження лише одного з відсортованих списків алгоритму Орчарда, діючи наступним чином. Для деяких точок p^* складається відсортований список інших точок за збільшенням відстані до p^*. Як і в алгоритмі Орчарда, зберігати кандидата найближчого до точки c. Щоб знайти точки, які є ближчі до q, ніж до c, проходимо по списку для p^* з позиції c, по черзі в кожному напрямку та обчислюємо відстані. Як і в алгоритмі Орчарда, якщо знайдено точка s, яка ближче до q, ніж c, встановлюємо c := s та продовжимо. Якщо точка s на нижній стороні має D (p^*, s) < D (p^*, q) − D (c, q), тоді ніякі більше точки на нижній стороні не повинні розглядатися за лемою 2.1 (пункт 3). Аналогічно, якщо точка на вищій стороні має D (p^*, s) > D (p^*, q) + D (c, q), тоді не потрібно розглядати подальші точки з вищої сторони. Якщо обидві умови дотримані, то поточний кандидат c є найближчим.

2.2.2 AESA (Алгоритм пошуку шляхом наближення та усунення)

Алгоритм пошуку шляхом наближення та усунення або AESA застосовує лему 2.1 і подібно до методу Орхарда, використовує попередню обробку та зберігання $\Omega(n^2)$. Алгоритм AESA попередньо обчислює та зберігає відстані D (x, y) для всіх x, y ∈ S та використовує функцію D_P нижньої межі, визначену в лемі 3.1. Коли AESA відповідає на запит для точки q, кожна точка x ∈ S знаходиться в одному з трьох станів:

• Відома, так що D (x, q) було обчислено; Відомі точки утворюють множину P;

• Невідома, так що доступна лише нижня межа D_P (x, q);

• Відхилена, так що D_P (x, q) більша за відстань найближчого відомої точки.

Алгоритм починається з усіх точок x невідомих, з D_P (x, q) = ∞, і повторює наступні кроки, поки всі точки не будуть відхилені або відомі:

1. вибрати невідому точку x з найменшим D_P (x, q);

2. обчислити D (x, q), щоб x стало відомим;

3. оновити найменшу відстань r відому q;

4. встановити P := P ∪ {x}, а для всіх невідомих x' оновити D_P (x', q); зробити x' відхиленим, якщо D_P (x', q) > r.

Базуючись на означенні:

$$D_{P \cup \{x\}}(x', q) = \max \{D_P(x', q), | D(x, q) - D(x, x') |\},$$

легко підтримувати це значення, так як додаються до P.

Необхідно буде розірвати зв'язки на першому кроці вибору, коли всі точки на початку мають D_P (x, q) = ∞. Це може бути зроблено випадково.

Хоча ця схема проста і відповідає на запити швидко, квадратична попередня обробка та зберігання обмежують її застосування. Лінійний алгоритм пошуку наближення та усунення, або LAESA, зменшує ці потреби, попередньо обчислюючи та зберігаючи відстані від усіх точок лише до підмножини V точок, названими півотами. Алгоритм діє як у AESA, але застосовує лише крок 4 оновлення, коли x ∈ V. Тому алгоритм вибирає повороти переважно на етапі 1.

Хоча AESA дуже ретельно використовує межі, що випливають через нерівність трикутника, можливо, остаточним у цьому напрямку є робота Шашаанда Ванга, алгоритм якого розглядає матрицю верхніх і нижніх меж на відстанях між точками в S ∪ {q} , і визначає, що замикання меж випливає з оцінки відстані. Набір оцінених відстаней дає симетричний або невід’ємний зважений неорієнтований граф. Нерівність трикутника дає верхню межу відстані між двома точками шляхом найкоротшого шляху у графі, а нижню границю - через верхні границю та оціненими відстанями.

2.2.3 Метричні дерева

Метричне дерево T (S) можна побудувати так: якщо |S| = 1, дерево має один вузол; інакше,

1. вибрати кульку B, з точкою як центр;

2. рекурсивно побудувати T (S ∩ B) і T (S \ B);

3. зробити ці два дерева дітьми кореня;

4. зберегти опис B у корені, включаючи його центральну точку.

Кожен вузол метричного дерева, таким чином, відповідає перетину кульок і кульових доповнень, що зберігаються у його предків. Відповідаючи на запит для точку q, дерево обходиться і обчислюється відстані до центрів кульових вузлів. По мірі обходу мінімум обчислених відстаней дає верхню межу на відстань найближчого сусіда, а отже, радіус кулі B_{q}, центрований у q. При обході вузла на дереві розглядаються регіони двох дітей вузла; якщо на основі даних кулі на шляху до кореня можна довести, що B_{q} не відповідає області дитини, дитину не потрібно відвідувати. В іншому випадку дитина відвідується. Вартість відповіді на запит пропорційна кількості досліджених вузлів.

Розділ 3. Розмірності

Хоча легко побудувати або натрапити на метричні простори, для яких вичерпний пошук є найшвидшим, все ж корисно розглянути ситуації, в яких це швидше можна зробити. Більше того, можливо, властивості простору, які роблять бажаним здійснювати пошук найближчого сусіда, також дають можливість швидко здійснити пошук.

Однією з таких властивостей є обмежена розмірність метричного простору для широкого визначення терміну розмірності. Таке визначення надає спосіб присвоїти дійсне число метричному простору; всі визначення, які ми вважаємо, збігаються (присвоюється однакове число) для "простих" множин. Отже, розмірність $\Re^d$ або його відкрита підмножина - це d для будь-якого з цих визначень, і розмірність d-багатозв'язна в $\Re^{d`}$ буде завжди d, незалежно від того, наскільки велике $d`$. Тобто, розмірності, як правило, "внутрішні", і покладаються на властивості даного метричного простору, а не в будь-якому просторі, в якому даний простір розташований.

3.1 Розмірності метричних просторів

Для обговорення багатьох понять розмірності ключове значення мають поняття накриття та упакування.

Накриття та упакування. Розглянемо обмежені метричні простори Z = (U, D), так що є деякий r з D (x, y) < r для всіх x, y ∈ U. Враховуючи є > 0, є-накриття Z, яка є множиною Y ⊂ U з властивістю, що для кожного x ∈ U існує деякий y ∈ Y з D (x, y) < є. Іншим словами, нехай

$$B(y, \epsilon) := \{x \in U \mid D(x,y) < \epsilon\}.$$

Тоді Y є-накриття, тоді і тільки тоді, якщо $U = \cup_{y\in Y} B(y, \epsilon)$. Іншим словами Y є-накриттям U, тоді і тільки тоді, коли відстань Гаусдорфа від U до Y менше, ніж є.

Номер накриття C (U, є) - це розмір найменшого є-накриття U. (Залежність числа накриття від функції відстані D неявна.)

Наприклад, якщо U - одиничний квадрат у площині, число накриття дорівнює $\Theta(1/\epsilon^2)$ як $\epsilon \to 0$, оскільки диск радіусу є може охоплювати лише

площу, пропорційну ϵ^2. Взагалі, число накриття одиничного гіперкуба в $\Re^d$ становить $\Theta\,(1/\epsilon^d)$ з подібних причин.

Кількість $\log_2 C\,(U, \epsilon)$ називається є-ентропією або метричною ентропією, функцією є. Вона вимірює величину бітів, необхідних для виявлення елементу простору, аж до спотворення є. Елементи накриття можуть становити кодову книгу для n-квантизатора з n = C (U, є). Такому квантувальнику знадобиться $\log_2 n$ біт, щоб передати наближення до члена x ∈ U, таким чином, щоб найгірше (не очікуване) спотворення D (x, f (x)) було не більше є.

Підмножина Y ⊂ U є є-упаковкою тоді і тільки тоді, коли D (x, y) > 2є для кожного x, y ∈ Y. Тобто набір кульок {B (y, є) | y ∈ Y} - неперервні.

Номер упакування P (U, є) - це розмір найбільшої є-упаковки. Номер є-упаковки тісно пов'язаний з номером накриття, як показано в наступній лемі.

Лема 3.1 Для заданого є > 0 та метричного простору (U, D), якщо P (U, є) і C (U, є) є скінченні, то

$$P(U, \epsilon) \leq C(U, \epsilon) \leq P(U, \epsilon/2).$$

Доведення: Максимальне (є/2)-пакування P має властивість, що жодна точка s ∈ U немає D (s, P) > є; інакше така точка може бути додана до P. Тобто максимальне (є/2)-пакування P є (є/2)-накриття, і тому найменше (є) - покриття не може бути більшим.

З іншого боку, для заданого (є/2)-накриття Y і (є/2)-пакування P кожна точка P повинна бути в B (y, є) для деякого y ∈ Y. Однак жодна з двох точок p, $p`$ ∈ P не може бути в такій самій кулі: тоді D (p, $p`$) < 2є нерівністю трикутника, що суперечить припущенню, що P - упаковка. Отже, кожне є-пакування не більше за будь-яке є-накриття.

Мережі та жадібний алгоритм. Тісний зв’язок упакування та накриття висвітлюється фундаментальною концепцією є-мереж. Множина Y ⊂ U – це є-мережа з (U, D), якщо вона є і є-накриттям, і (є/2)-пакуванням.

Є-мережа може бути побудована за наступним жадібним алгоритмом, на вхід якого подається є ≥ 0 та максимально допустимий розмір k, а також метричний простір (U, D). Алгоритм: вибрати s ∈ U довільно та встановити Y:= {s}. Повторювати наступне: вибрати s ∈ U, який максимізує D (s, Y) = min {D (s, y) | y ∈ Y}. Якщо D (s, Y) < є або |Y| ≥ k, зупинитись. В іншому випадку встановити Y: = Y ∪ {s} та продовжувати.

Повернений Y є ϵ'-накриттям для деяких ϵ', при цьому $\epsilon' < \epsilon$, якщо k досить велике. Нехай i-точка, додана до Y, позначається s_i, а Y_i позначає

множину Y перед додаванням s_i. Оскільки послідовність D (s_i, Y_i) при i = 2 ..., |Y| не збільшується, кожен член Y є щонайменше ϵ' від кожного іншого члена, і тому Y є (ϵ'/2)-упаковкою, а значить, ϵ'-мережею. Оскільки Y - (ϵ'/2)-упаковка, то за лемою вище, будь-яка (ϵ'/2)-накриття повинно мати щонайменше |Y| членів. Якщо цей жадібний алгоритм, який запускається на вході з ϵ = 0, то на виході Y матиме розмір k, а будь-яке (ϵ'/2)-накриття повинно мати принаймні k членів; тобто алгоритм дає відстань на відстані ϵ' не більше ніж удвічі найкраще для k точок: це алгоритм наближення для задачі k-центру, що визначає k точок, максимальна відстань яких до якої-небудь точки U зведена до мінімуму. Гонсалес, а також незалежно Хохбаум і Шмойс показали, що це найкращий можливий коефіцієнт наближення алгоритму поліноміального часу на загальному метричному просторі, поки рівність P = NP не виконується.

Як було зазначено, цей алгоритм застосовувався при побудові структур даних найближчих сусідів. Він також використовувався в обчислювальній хімії.

Розмірність Мінковського. Розмірність Мінковського $\dim_B$ (Z), де Z = (U, D) можна визначити наступним чином: це таке d, що задовольняє накривне число

$$C(U, \epsilon) = 1/\epsilon^{d+o(1)} \quad (2)$$

$\epsilon \to 0$, якщо таке d існує. Тобто накривні (і пакувальні) числа залежать приблизно від поліноміальної шкали вимірювання ϵ, і $\dim_B$ (Z) є граничним ступенем цього многочлена. Зазначена умова на d часто виражається як

$$d = \frac{\lim}{\epsilon \to 0} \frac{\log C(U,\epsilon)}{\log(1/\epsilon)}.$$

Розмірність Мінковського не повинна бути цілим числом; множини з не цілою розмірністю часто називають фракталами. Множина також може мати нульову розмірність, але бути повністю розмірною; наприклад, криві Гілберта в площині мають розмірність Мінковського два, але площину нуль. Раціональні числа мають розмірність Мінковського один, але довжину нуль. (Цю останню властивість, як правило, розглядають математично як "погану", оскільки для інших розмірностец зліченної множини $\cup_i U_i$ не більше, ніж $\sup_i$ і $\dim U_i$, тому раціональні числа "повинні" мати нульову розмірність.)

Інший погляд на розмірність Мінковського - це те, що вона є критичним значенням для вмісту коробки C(U, ϵ) = ϵ^t. Тобто, припустимо, кожна куля у покритті має об'єм, пропорційний щонайменше ϵ^t, як це було б і в $\Re^t$. Тоді вікно t-змісту є приблизним завищенням об'єму U, оскільки це сума об'ємів невеликої колекції множин, об'єднання яких містить U.

Припустимо, число покриття дорівнює $1/\epsilon^{d+o(1)}$; тоді t-вміст $1/\epsilon^{t-d+o(1)}$, де $\rightarrow 0$, що йде до 0 для $t > d$, та ∞ для $t < d$. Тобто, d є супремумом t, для якої t-вміст є нескінченністю, або числом t, для якого вміст t дорівнює нулю.

Розділ 4. Розмірність та пошук найближчого сусіда

Оцінка розмірності

Одиниці розмірності та пошук найближчого сусіда пов'язані в обох напрямках: обчислення деяких мір розмірності можна здійснити за допомогою пошуку найближчого сусіда, а простори з обмеженою розмірністю можуть мати швидші структури даних для пошуку найближчих сусідів, як теоретично, так і емпірично.

Пошук найближчого сусіда для оцінки розмірності.

Для даної множини точок оцінку дерева квандрантів простіше обчислити, ніж інтеграл кореляції, і тому Белуссі і Фалуцос [10] використовують квадратурний оцінювач дерева квадрантів у контексті просторових приєднань бази даних. Одним з видів просторового приєднань є сукупність пар точок, розташованих на відстані є одна від одної, для деяких заданих є. Тобто, його розмір є точною оцінкою на основі відстані кореляційного інтеграла. Белуссі і Фалутсос пропонують оцінювач дерева квадрантів, щоб допомогти оцінити розмір відповіді для просторових приєднань.

Точкова розмірність. Поки що були розглянуті оцінки, засновані на дереві квадрантів і з фіксованим радіусом у запитах; клас оцінювачів, ще більш безпосередньо пов'язаний із пошуком найближчого сусіда, - це ті, які базуються на k-NN пошуку. Наприклад, Катлер і Доусон [4] показали, що точкова розмірність, пов'язана з інформаційною розмірністю, має k-у відстань найближчого сусіда як оцінювач

$$\alpha_\mu(x) = \lim_{n\to\infty} \frac{\log(k/n)}{\log \delta_{k:n}(x)}, \quad (5)$$

з ймовірністю 1, де n - розмір вибірки і $\delta_{k:n}(x)$ - відстань x до його k-го найближчого сусіда у вибірці. Іншими словами,

$$\delta_{1:n}(x) = n^{-1/\alpha_\mu(x)+o(1)}$$

як $n \to \infty$. Аналогічні спостереження зробили Петтіс [11], Вервер та Дюін [VD95], Ван де Вотер та Шрам [5]. Виведення аналогічного оцінювача з максимальною вірогідністю дали Левіна та Бікель [12].

Евристично (5) можна зрозуміти, вважаючи ϵ_k таким, що куля B (x, ϵ_k) має масу ймовірності μ (B (x, ϵ_k)) = k / n. Очікувана кількість балів у вибірці, що падає в B (x, k), становить k, і так $\delta_{k:n}$ (x) ≈ k, і для тому

$$k/n = \mu(B(x, \epsilon_k)) \approx \epsilon_k^{\alpha_\mu(x)} \approx \delta_{k:n}(x)^{\alpha_\mu(x)},$$

з використанням позначення точкової розмірності, а (5) слідує після прийняття логарифмів та ділення. Це відношення до точкової розмірності говорить про те, що найближчі відстані можуть бути корисними при оцінці інших пов'язаних оцінок розмірності, таких як інформаційна, енергетична і навіть розмірність Хаусдорфа.

Доповідь Тао, пов'язана з оцінкою Белуссі та Фалуцосо, використовує оцінки точкової розмірності для оцінки вартості запиту найближчого сусіда та оцінки розміру; задане (S, D), точкову розмірність для кожної точки у вибірці P ⊂ S, а потім для даної точки запиту використовується оцінка точкової розмірності для прилеглої точки вибірки. Оцінка точкової розмірності проводиться за допомогою міри підрахунку і називається степеневим розподілом.

Найгірший випадок, пов'язаний з точковою розмірністю графової метрики використовується Гао і Чжаном в контексті маршрутизації. З огляду на співвідношення Гаусдорфа та точкової розмірності, можливо, їхній зв’язок є своєрідним розміром графовою розмірністю Хаусдорфа.

Екстремальні графи як оцінки розмірності. У налаштуваннях евклідових многовидів Коста та Герой пропонують використовувати в якості оцінок розмірності дерева мінімального розміру, відповідний граф k-NN або інші екстремальні графи. Припустимо, G - такий граф для множини n точок незалежно, однаково розподілених на d-многовиді. Для v з 0 < v < d, нехай

$$L(G, v) := \sum_{e \text{ an edge of } G} \ell(e)^v,$$

сума потужності довжини ребра G. Коста і Герон використовують цей факт, повертаючись до відомих результатів Бертвурда та ін., що

$$L(G, v)/n = n^{-v/d+o(1)}$$

де n → ∞, для щойно згаданих екстремальних графів та ін. Це дозволяє топологічну розмірність d многовида оцінити як функцію L (G, v) і n, так, наприклад,

$$d = \lim_{n \to \infty} \frac{\log(1/n)}{\log(L(G,1)/n)}$$

з ймовірністю один.

Вираз (5) відповідає для випадку 1-ближнього сусіда графа в d-многовиді, оскільки L (G, 1) / n - середня відстань найближчого сусіда в графі, і всі точки в многовиді мають точкову розмірність d. Крім того, алгоритми для виявлення екстремальних графів включають запити найближчих сусідів. Ці оцінки також забезпечують власне масштабування: немає є, яке прямує до нуля, як у визначенні розмірності, а скоріше масштаб вимірювання 1 / n є наслідком залучених відносин з найближчим сусідом.

4.2 Постійний розмірність та пошук найближчих сусідів

4.2.1 Основні властивості, розповсюдження

Деякі основні властивості метричних просторів Z = (S, D) з обмеженою розмірністю Ассоада, тобто постійною подвоєною розмірністю, корисна при пошуку найближчого сусіда. Нагадаємо, що для Z з постійно подвоєною розмірністю існує значення d = $\mathrm{doub}_A(Z)$, так що будь-яка кулька радіусу r має (rє) -покриття розміром не більше O(1/ϵ^d), де є → 0 Як показано нижче, це означає зворотну умову найближчого сусіда: кожна точка s ∈ S є найближчим сусідом до точок$^{O(2^{O(d)} \log \Delta(S))}$, де Δ (S) - відношення відстані між найдальшої пари точок на відстань до найближчої пари точок.

Перед тим, як буде показано умову найближчого сусіда, короткий відступ на відношення Δ (S): воно по-різному відоме як співвідношення відстані, співвідношення сторін і поширення, де останнє, здається, є найбільш поширеним. Алгоритми вирішення проблеми пошуку найближчого сусіда, які залежать від поширення, відомі вже давно: наприклад, алгоритми для знаходження всіх найближчих сусідів або all-k-NN за $\Re^d$, що займають час O(n logΔ (S)). Рідше описані також комбінаторні властивості точкових множин у $\Re^d$ з дуже низьким поширенням, а для класичних алгоритмів кластеризації з точки зору поширення були задані межі.

Хоча включати залежність від поширення у межі не так добре, часто ця залежність є лише від логарифму поширення. Зробити алгоритм складнішим для усунення такої залежності навряд чи складе проблеми на практиці.

Тут згадується зворотня умова найближчого сусіда. Це стосується не лише найближчих сусідів, але й у більш загальному плані "k-их (γ) - найближчих" сусідів. Точка a є k-им (γ) – найближчим сусідом до точки b, відносно S, якщо в S є максимум k − 1 точок, відстань до b яких знаходиться в межах коефіцієнта γ відстані, найближчого до b в S\{b}.

Лема 4.1. Для метричного простору Z = (S, D) з подвоєною розмірністю d = $\dim_A$ (Z) та для будь-якої точки s ∈ S, кількістю точок s` ∈ S, для яких s є k-им (γ) -найближчим у S до s` є $O((2\gamma)^d k \log \Delta(S))$, де 1 / γ → 0.

Подвоєння постійних просторів з Евклідовим простором. Для евклідових просторів існує гостріша форма вищевказаної межі, що застосовується до k-го найближчого сусіда: для S ⊂ $\Re^d$, точка k-а є найближчою до $k2^{O(d)}$ інших точок.

Ще одна умова, яка задовольняє підмножини евклідових просторів, - це те, що для будь-якої точки s і запиту q, якщо s є найближчим сусідом до q, тоді це можна довести за допомогою сусідів Делоне точки s. (Точки a і b - сусіди Делоне, якщо на його граничній сфері є куля з a і b, а в її внутрішній частині немає ділянок.) Якщо s ближче до q, ніж будь-який сусід Делоне з s до q, тоді s є найближчою до q в S. Точка може мати багато сусідів Делоне, навіть у площині, але для випадкових точок при багатьох розподілах ймовірностей точка може мати O (1) очікуваних сусідів Делоне. Якщо найближчий сусід до запитом можна «вгадати», то в таких випадках його статус можна довести в постійний очікуваний час. Будь-яка схожа умова для метричних просторів включає залежність від розкиду.

4.2.2 Розділяй та володарюй

Далі ми розглянемо застосування підходу Розділяй та Володарюй для побудови структури даних та відповіді на запит. За деяких умов можна розділити пошукову задачу на підпроблеми: набір точок S виражається як об'єднання множин S_1, S_2,..., так що для будь-якої точки запиту q, найближчою в S до q, є в одному з наборів S_i, і є ефективний тест для перевірки цієї умови. У цьому налаштуванні можна знайти природну структуру даних T (S) шляхом рекурсивного пошуку T (S_i) для всіх S_i та зробити кожного з них дочірніми від кореня дерева. Алгоритм пошуку є: застосовувати цей «ефективний тест», щоб вибрати S_i, а потім рекурсивно шукати T(S_i).

Така структура даних є хорошою, оскільки не потребує "зворотного повернення", тобто це структура дерева, за якою пошук йде від кореня, по одному шляху, до листа.

Ключові властивості такого підходу - це обмеження як на $\max_i|S_i|$, так і на загальний $\sum_i |S_i|$. Перший визначає кількість рівнів у структурі даних, а останній необхідний для визначення розміру структури даних.

Одним із прикладів такої схеми є алгоритм Кларксона для випадку Евкліда.

У наведених нижче прикладах схема ділення та володарювання базується на визначенні найближчого сусіда q у підмножині P ⊂ S. Для мотивації таких підходів ми повернемося до деяких основних міркувань щодо пошуку найближчого сусіда.

Межі за допомогою найближчих сусідів у підмножині. Завдання пошуку найближчого сусіда можна розглядати як таке, що складається з двох частин: встановлення найближчого сусіда та доведення, що всі інші точки не є найближчим сусідом. Більше того, будь-який найближчий сусідній алгоритм в будь-який момент часу під час обробки запиту обчислює відстань від точки запиту q до деякої підмножини P точок. Отже, алгоритму потрібно використовувати оцінку відстані від q до P, щоб довести, що деякі точки не можуть бути відповіддю на запит. Який найефективніший спосіб це зробити?

Дивлячись на лему 2.1 (1), щоб показати, що s ∈ S \ P далеко від q, враховуючи, що відстань деякого p ∈ P до q відома, нижню межу

$$| D (p, q) – D (p, s) |$$

для D (q, s) можна використати. Важко сказати, враховуючи різних учасників p з P, які p максимізували би цей вираз. Однак, щоб збільшити різницю двох відстаней у виразі, можна спробувати зробити одну чи іншу відстань якомога меншою. p ∈ P, що мінімізує D (p, q), звичайно, найближчий у P до q, тоді як p ∈ P, що мінімізує D (p, s), є найближчим у P до s. Отже, якщо деяке p ∈ P близьке до q і далеке від s, або близьке до s і далеке від q, це дає доказ того, що s не може бути близьким до q.

Ці факти говорять про те, що одна основна інформація для точки запиту q є найближчою точкою в P. Далі розглянемо, як таку інформацію можна використовувати разом із подвоєними постійними константами та подвоєними умовами вимірювання, щоб запропонувати деякі структури даних для пошуку найближчого сусіда, які мають доведені властивості. Ці структури даних будуть неефективними у своїх ресурсних межах, але будуть ілюструвати взаємозв'язки.

У кожному з трьох наведених нижче прикладів буде знайдено підмножину P з S розміром m, разом з кулькою B_p для кожного p ∈ P. Це буде мати властивість, що для точки запиту q, якщо p найближче до q в P, то для деяких умов найближчий сусід q в S міститься в B_p. Крім того, деякий прогрес буде досягнутий цим, або тому, що B_p є маленьким, або в ньому не так вже й багато точок. Три розглянуті випадки:

1) У просторі є подвоєна константа, P є-множина, і або q досить далеко від p, що p саме по собі приблизно найближче, або B_p містить найближчу до q в S. Кожна куля B_p менша за постійний коефіцієнт, ніж куля, що містить S.
2) Простір - це (емпірична) подвійна міра, P - випадкова підмножина, а B_p містить найближчу точку до q в S з дуже високою ймовірністю. Більше того, B_p містить O (n ($\log_n$) / m) точок.
3) У просторі є подвоєна константа, а запити взаємозамінні з точками. Тут P - випадкова підмножина, а B_p містить найближче до q у S з контрольовано високою ймовірністю 1−1 / K, для даного K. Більше того, очікується, що B_p містить O ((K_n/m) $\log^2\Delta$ (S)) точок.

Прямий підхід до використання цих конструкцій - це знову ж таки, застосовувати їх рекурсивно: щоб будувати T (S), треба знайти підмножину P і кулі B_p для кожного p ∈ P, а потім рекурсивно будувати T (S ∩ B_p) для кожного p ∈ P. Для пошуку найближчого сусіда в S до запиту точки q, треба знайти p ∈ P, найближче до q, а потім рекурсивно шукати T (S ∩ B_p)).

Розділяй та володарюй: Подвоєння постійних просторів. Розглянемо метричний простір Z = (U, D) з обмеженою подвоєною розмірністю d = $doub_A$ (Z), вхідними точками S ⊂ U та проблемою побудови структури даних для наближеного пошуку найближчого сусіда. Припустимо, ми масштабуємо вимірювання відстані так, що точки лежать у кулі радіуса одиниця, і припустимо, що підмножина P є δ^2-нетто, для параметра δ > 0. Це означає, зокрема, що будь-яка точка у S знаходиться в межах δ^2 точки в P. Більш того, умова подвоєного розмірності означає, що існує обмеження кількості точок у δ^2-мережі, а саме O (1 / δ^{2d}). Тепер припустимо, що точка запиту q має p як найближчого сусіда у P, та a як найближчого сусіда у S, а також найближчого сусіда a в P є p_a ∈ P. Тоді D (a, p_a) ≤ δ^2, і так

$$D(q, p) \le D(q, p_a) \le D(q, a) + D(a, p_a) \le D(q, a) + \delta^2.$$

Тобто, якщо D (q, a) > δ, тоді p є (1 + δ)-найближче до q в S. В іншому випадку, при D (q, a) ≤ δ маємо

$$D(p, a) \le D(p, q) + D(q, a) \le 2\delta + \delta 2 \le 3\delta,$$

для δ < 1. Ціною пошуку точок P ми підтверджуємо відповідь на запит до кулі B_p := B (p, 3δ), якщо тільки p не є прийнятною відповіддю. Припустимо, ми рекурсивно будуємо структуру даних для кожного p ∈ P для S ∩ B (p, 3δ), при δ <1/6. Тоді на глибині t у такій структурі даних точки знаходяться в кулі радіусом 1 / 2^t.

Побудова такої структури даних має припинятися, коли в поточному множині S існує лише одна точка. Таким чином, глибина цієї структури даних та вартість пошуку її найближчого сусіда пропорційна logΔ (S). Структура даних, накреслена вище, може відповісти на запит наближеним найближчим сусідом в часі O (2^t (d) logΔ (S)), якщо δ і так m = | P | = O (1 / δ2d) - константи.

Розділяй та володарюй: подвійна розмірність простору. Розглянемо тепер метричний простір (U, D), для якого емпірична міра μ_C подвоюється для S ⊂ U та q ∈ U. Такий простір має властивість, що | S ∩ B (x, r) | ≥ | S ∩ B (x, 2r) | / 2^C для значення C, для всіх x ∈ S ∪ {q} та r > 0.

Надалі скоротимо | S ∩ B (x, r) | до | B (x, r) |. Зафіксуймо точку запиту q, і нехай P - випадкова підмножина S, отримана шляхом вибору кожної точки S незалежно з ймовірністю m / n, для параметру m. Очікуваний розмір P є m. Для p ∈ P розглянемо ϵ_p, обране так, що |B (p, ϵ_p) | = Kn (log n) / m, де n:= |S| і значення K і m мають бути визначені. Для p ∈ P припустимо, що D (q, p) ≤ ϵ_p / 2, а p є найближчий до q в P. Тоді найближча ділянка до q в S міститься в B (p, ϵ_p) за лемою 3.1 (4). З іншого боку, якщо β := D (q, ϵ_p) > ϵ_p / 2, то

$$|B(q,\beta)| \geq |B(q,3\beta)|/4^C \geq |B(p,\epsilon_p)|/4^C \geq Kn(\log n)/m4^C.$$

де друга нерівність випливає з B (p, ϵ_p) ⊂ B (q, 3β), з якої випливає, що для x ∈ B (p, ϵ_p), з

$$D(q,x) \leq D(q,p) + D(p,x) \leq \beta + \epsilon_p \leq 3\beta.$$

Ймовірність того, що p найближче до q в P, не більше, ніж ймовірність того, що B (q, β) не має точок P, що не більше

$$(1-m/n)^{Kn(\log n)/m4^C} \leq e^{-K(\log n)/4^C} = 1/n^{K/4^C}.$$

Якщо, наприклад, K /4^C > 10, то q матиме p найближче в P з вірогідністю не більше 1 / n^{10}.

Лема 4.2 Припустимо, (U, D) - метричний простір, S ⊂ U, і q ∈ U, таке, що існує деяка константа C, для якої

$$|S \cap B(x, r)| \geq |S \cap B(x, 2^r)|/ 2^C$$

для всіх x ∈ S ∪ {q} і r. Припустимо, P - випадкова підмножина S, де p ∈ S вибирається незалежно для P з вірогідністю m / n. Тоді з ймовірністю щонайменше $1 - 1 / n^{K/4^C}$, найближчий сусід q до S міститиметься у підмножині S розміром Kn (log n) / m.

Якщо m := 10K logn, цей розмір дорівнює n / 10, тому якщо структура даних будується для кожної підмножини рекурсивно, глибина буде log n. Вибір K: = 10 (log n) 4^C означає, що ймовірність того, що будь-який крок у пошуку заданого q вийде з ладу, не більше ніж 1 / n^9.

Розділяй та володарюй: Взаємозамінні запити. Для того, щоб створити метричні простори з подвійною константою, можна побудувати структуру даних для наближеного пошуку найближчого сусіда, а для подвоєння метричних просторів можна побудувати структуру даних для точного пошуку. Хоча схеми, наведені вище, є грубими, найкращі структури даних, відомі для метричних просторів в цих умовах, мають схожу поведінку: наближене для подвоєної константи, точне для подвійного виміру. Це не задовольняє, оскільки умова подвоєного виміру здається дуже крихкою. Умова подвоєної константи є більш надійною, але алгоритми наближення мають таку складність, що для деяких метричних просторів та деяких застосувань вони можуть мати низьку точність: для точок, рівномірно розподілених у високій розмірності, кожна точка не набагато віддалена ві найближчої точки. Алгоритм наближення може взагалі повертати будь-яку точку.

Кращою метою, таким чином, була б структура даних для точних запитів, що добре для подвоєння константних просторів. На жаль, така структура даних не відома, тому варто знати про додаткові умови, за яких можна побудувати очевидно хороші структури даних.

Відома одна така умова: коли запити мають той самий розподіл, що й точки, то вони є взаємозамінними. Припущення тут є деяким випадковим генератором точок та запитів, таким чином, що справедливо наступне: для представленої точки запиту q множин P ∪ {q} і $P`$ мають однаковий розподіл, коли P і $P`$ є випадковими підмножинами S, та P має одну меншу точку, ніж $P`$. Це може мати місце, наприклад, коли точки та запити є незалежними, однаково розподіленими випадковими змінними або якщо точки та запити були обрані навмання з якогось великого дискретного набору. Такі умови

приблизно дотримуються, наприклад, для квантування векторів, де точки спеціально вибираються репрезентативними для розподілу запитів.

Цей стан разом із подвоєною константою розмірності передбачає деякі корисні межі. Зокрема, конструкція розділяй та володарює, аналогічна раніше заданій, полягає в наступному: вибрати випадкову підмножину P ⊂ S розміром m, а потім вибрати випадкову підмножину Ṕ ⊂ S розміром Km, де буде визначено K і m . Передбачаючи p ∈ P, розглянемо точку $q_p \in P`$, яка має p найближчу в P, але найвіддаленішу серед усіх таких точок у $P`$. Покажемо, що куля B_p := B(q_p, 3D (p, q_p)), ймовірно, буде містити точку відповідь, для взаємозмінних запитів q з p, найближчою до P. Також покажемо, що в B_p очікується не надто багато сайтів.

Лема 4.3 За умов трохи вище, для s ∈ Ṕ з p, найближчим до s в P, і найближчим до s в S, справедливо, що D (a, q_p) ≤ 3D (p, q_p).

Доведення: Оскільки q_p далі від p, ніж s, D (s, a) ≤ D (s, p) ≤ D (q_p, p), тоді D (q_p, a) ≤ D (q_p, p) + D (p , s) + D (s, a) ≤ 3D (p, q_p), використовуючи нерівність трикутника та припущення. #

Лема 4.4 За умов попередньої леми, якщо q є точкою запиту, що взаємозмінюється з p, найближчою до P, то з ймовірністю 1−1 / K найближчий сусід до q у S міститься у B (q_p, 3D (p, q_p)).

Лема 5.5 Для P ⊂ S випадкове підмножина розміром m, Ṕ ⊂ S випадкова підмножина розміру Km, та q - взаємозмінний запит, є в очікуванні

$$2^{O(d)}O(Kn/m)\log^2\Delta(S)$$

точки x такі, що: є деякий q` в Ṕ з x a (3) -поблизу точкою по відношенню до P, і деякий p ∈ P, найближчий до P до q і q`.

Нам дійсно потрібно лише обмежити очікувану кількість точок x у такій конфігурації q` = $q_p$. Однак, здається, простіше зв'язати число зі слабшою умовою на q`. Сукупність таких точок x для заданого p ∈ P містить B (q_p, 3D (q_p, p)) ∩ S.

Висновок

У курсовій роботі було розглянуто декілька підходів до відомої проблеми пошуку, що називається пошук найближчого сусіда. Задача полягає у пошуку такої структури даних, що для заданого метричного простору (U, D) і заданої підмножини S множини точок U знайти таку структури даних для S, такої, щоб для точки q можна було швидко знайти точку $s \in S$, для якої відстань D(s, q) є мінімальною. В курсовій роботі розглядається декілька різних підходів до цієї проблеми. Обговорюється залежність їх ефективності від розмірності метричного простору, властивостей і кількості елементів метричного простору і множини S. Також було розглянуто використання нерівностей трикутника при побудові структур даних, ще питання розмірності та пошуку найближчого сусіда.

Особливо детально розглянуто два алгоритми: Розділяй та володарюй алгоритм та алгоритм AESA. Порівняно складність та особливості застосувань цих алгоритмів.

Обидва розглянуті алгоритми реалізовано на мові Python для евклідової метрики. Усі вищеописані завдання було виконано.

Список використаної літератури

Характеристика джерела	Назва джерела
Статті:	1. Nearest-Neighbor Searching and Metric Space Dimensions, Kenneth L. Clarkson 2. An optimized version of the Approximating and Eliminating Search Algorithm (AESA) for Nearest Neighbour classification, A. Juan and E. Vidal 3. E. Fix and J. L. Hodges Jr. Discriminatory analysis, non-parametric discrimination. Technical Report 4, USAF School of Aviation Medicine, 1951. Project 21-49-004. 4. C. D. Cutler and D. A. Dawson. Estimation of dimension for spatiallydistributed data and related theorems. Journal of Multivariate Analysis, 28:115–148, 1989. 5. W. van de Water and P. Schram. Generalized dimensions from nearneighbor information. Phys. Rev. A, 37:3118–3125, 1988.
Книги:	6. [KTAD79. W. A. Burkhard and R. M. Keller. Some approaches to best-match file searching. Commun. ACM, 16:230–236, 1973. 7. G. R. Hjaltason and H. Samet. Index-driven similarity search in metric spaces. ACM Trans. Database Syst., 28(4):517–580, 2003. 8. Y. Linde, A. Buzo, and R. M. Gray. An algorithm for vector quantizer design. IEEE Transactions on Communications, 28:84–95, 1980. 9. R.M. Gray and D. L. Neuhoff. Quantization. IEEE Trans. Inform. Theory, 44:2325–2383, 1993. 9. L. Devroye, L. Gy¨orfi, and G. Lugosi. A Probabilistic Theory of Pattern Recognition. Springer-Verlag, New York, 1996. 10. A. Belussi and C. Faloutsos. Self-spatial join selectivity

estimation using fractal concepts. ACM Trans. Inf. Syst., 16(2):161–201, 1998.

11. K.W.Pettis, T.A.Bailey, A.K.Jain, and R.C. Dubes. An intrinsic dimensionality estimator from near-neighbor information. IEEE Transactions on Pattern Analysis and Machine Intelligence, 1:25–37, 1979.

12. E. Levina and P. J. Bickel. Maximum likelihood estimation of intrinsic dimension. In L. K. Saul, Y. Weiss, and L. Bottou, editors, Advances in Neural Information Processing Systems 17, pages 777–784. MIT Press, Cambridge, MA, 2005.