

## ЗМІСТ

Вступ.....	2
1. Основні поняття теорії розпізнавання образів .....	3
1.1. Підходи до розпізнавання образів .....	5
1.2. Приклади задач розпізнавання образів .....	5
1.3. Процедура розпізнавання .....	6
1.4. Розробка системи розпізнавання .....	6
1.5. Перетворення зорових образів у цифровий код.....	7
1.6. Метод нормалізації плоских зображень .....	8
2. Методи розпізнавання.....	11
2.1. Метод січних площин .....	12
2.2. Метод потенціалів для розпізнавання образів .....	18
2.3. Класифікація образів із використанням функції відстані .....	22
3. Кластеризація.....	25
3.1. Ієрархічні алгоритми кластеризації.....	26
3.2. Неієрархічні алгоритми кластеризації .....	28
Література .....	33

## Вступ

Навчальна дисципліна "Теорія розпізнавання образів" вивчається студентами заочної форми навчання факультету інформаційних технологій.

Актуальність вивчення основних понять теорії розпізнавання образів фахівцями у галузі комп'ютерних наук зумовлена потребою розроблення інтелектуальних інформаційних систем, які у більшості випадків містять підсистеми розпізнавання. Тому важливою складовою процесу підготовки фахівців сфери ІТ є вироблення знань та навичок, які стосуються розуміння та використання сучасних методів розпізнавання образів.

Методичний посібник складається із трьох основних розділів. У першому розділі наведено основні означення теорії розпізнавання образів, розглянуто приклади задач розпізнавання, описано найбільш популярні підходи до розробки інформаційних розпізнаючих систем та методи перетворення зображень у цифровий код, а також їх нормалізацію. Другий розділ присвячений методам розпізнавання. У ньому наведено метод січних площин, метод потенціалів та найпростіші метричні методи класифікації, засновані на принципі "найближчого" сусідства. У третьому розділі розглянуто основні поняття кластерного аналізу, наведено найбільш розповсюджені міри близькості між об'єктами та класами, описано найпростіші ієрархічні та неієрархічні алгоритми класифікації.

## 1. Основні поняття теорії розпізнавання образів

*Теорія розпізнавання образів* — розділ кібернетики, що розвиває теоретичні основи й методи *класифікації* і *ідентифікації* предметів, явищ, процесів, сигналів, ситуацій і т. п. об'єктів, які характеризуються скінченним набором деяких властивостей і ознак. Такі задачі вирішуються досить часто, наприклад, при переході або проїзді вулиці за сигналами світлофора. Розпізнавання кольору лампи світлофора, що засвітилася, і знання правил дорожнього руху дозволяє прийняти правильне рішення про те, можна, чи не можна переходити вулицю в цей момент.

У більшості випадків, сприймаючи явища навколишнього світу, людина здійснює їх *класифікацію*, тобто розбиває ці явища (предмети, ситуації) на групи схожих явищ (саме схожих, а не тотожних). По тим чи іншим причинам необхідно віднести в одну групу у чомусь "подібні" явища чи предмети, які можуть при цьому значно відрізнятися один від одного. Наприклад, усі фігури, зображені на наступному малюнку, ми називаємо "літерою А", незважаючи на їх значну відмінність у їх написанні (див рис. 1).



Рис. 1.

Суттєвим є той факт, що виділивши такі групи (множини) об'єктів, ми отримуємо здатність "упізнавати", тобто встановлювати належність до вже відомої множини, нові об'єкти, які ще не зустрічалися нам раніше, наприклад впізнавати букви, написані новим для нас почерком.

Отримавши уявлення про те, що являє собою буква "А" на основі деякої, зазвичай невеликої кількості екземплярів цієї букви, ми спроможні упізнати як завгодно велику кількість інших її екземплярів.

Проте, далеко не всі множини об'єктів дають змогу на основі невеликої частини множини упізнати як завгодно багато інших невідомих нам її представників. Наприклад, фотографії студентів деякого вузу утворюють множину. Проте неможливо після ознайомлення зі скажімо десятьма фотографіями студентів визначити по новій фотографії є людина студентом цього вузу чи ні.

Таким чином існують множини деякого особливого типу. Ці множини мають *характерну властивість*, яка виявляється у тому, що *після ознайомлення із скінченною частиною об'єктів цих множин, можна упізнавати* як завгодно велику кількість інших їх представників. Множини такого типу будемо називати *образами*.

Прикладами образів можуть бути такі множини: чоловіки, дитячі портрети, ссавці, картини Пікассо, цифри 5, зображення винищувачів МІГ. Застосовувати до них термін "образ" ми можемо тому, що ознайомлення з образом не пов'язано із запам'ятовуванням окремих об'єктів, а упізнавання нового об'єкта відбувається без безпосереднього порівняння із кожним раніше відомим.

Характерна властивість образів об'єктивна у тому сенсі, що різні люди (живі істоти), які навчалися на різних групах об'єктів образу, у переважній більшості однаково і незалежно один від одного класифікують одні і ті самі нові об'єкти. Саме об'єктивність цієї властивості образів дає змогу людям, які вчилися у різних школах, успішно розпізнавати раніше невідомий їм почерк.

Проте, самі образи у той самий час є у певній мірі "розпливчастими" і питання про належність об'єкта до даного образу не завжди може бути вирішене однозначно. На рис. 2 показано, як за допомогою невеликих послідовних змін можна перетворити цифру 5 у цифру 3.

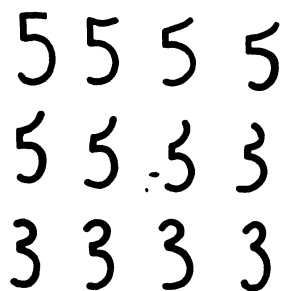


Рис. 2.

Рис. 2 ілюструє важливу особливість об'єктів кожного образу. Можна в деяких межах змінювати об'єкт, але він усе одно залишиться об'єктом того самого образу.

Створення штучних систем *розпізнавання образів* залишається складною теоретичною й технічною проблемою. Необхідність у такому розпізнаванні виникає в самих різних областях — від військової справи й систем безпеки до оцифрування різних аналогових сигналів.

Сприйняття явищ у формі образів відіграє надзвичайно важливу роль у процесах пізнання зовнішнього світу. У процесі біологічної еволюції багато тварин за допомогою зорового й слухового апарата вирішили задачу *розпізнавання образів* досить добре. Як впливає із самого означення образу "розпізнаванню" нових для нас об'єктів передуює процес навчання. Під час навчання істоти ознайомлюються із деякою кількістю об'єктів і, крім цього, із якогось джерела (наприклад від батьків, старших і т.п.) отримують інформацію про те, до якого образу відноситься кожний із цих об'єктів. Цей процес отримав назву "*навчання з учителем*".

Більш загальний характер має "*навчання без учителя*", у процесі якого система вчиться спонтанно виконувати поставлене завдання без втручання з

боку "вчителя". Навчання машин без вчителя формулюється як задача *кластерного аналізу*. Вибірка об'єктів розбивається на кластери (множини, що мають порожній перетин), таким чином, що кожний кластер складається із "схожих" об'єктів, а різні кластери "суттєво" відрізняються один від одного. *Кластеризація* часто використовується в якості допоміжного засобу розв'язування задач класифікації та регресійного аналізу. Деякі алгоритми розв'язування задач класифікації комбінують навчання з учителем та навчання без учителя (наприклад навчання мереж векторного квантування).

Традиційно задачі розпізнавання образів включають у коло задач *штучного інтелекту*. Можна виділити два основних напрямки:

- Вивчення здібностей до розпізнавання, якими володіють живі істоти, їхнє пояснення й моделювання;
- Розвиток теорії й методів побудови пристроїв, призначених для розв'язання окремих задач у прикладних цілях.

*Формальна постановка задачі розпізнавання:* розпізнавання образів — це віднесення вихідних даних до певного класу за допомогою виділення істотних ознак, що характеризують ці дані, із загальної маси несуттєвих даних. При постановці задач розпізнавання намагаються користуватися математичною мовою.

### **1.1. Підходи до розпізнавання образів**

Для оптичного розпізнавання образів можна застосувати метод перебору вигляду об'єкта під різними кутами, масштабами, зсувами й т. д. Для букв потрібно перебирати шрифт, властивості шрифту й т. д.

Другий підхід — знайти контур об'єкта й досліджувати його властивості (зв'язність, наявність кутів і т. д.)

Ще один підхід — використовувати штучні нейронні мережі (багатошарові перцептрони, мережі квантування, мапи Кохонена, рекурентні мережі). Цей метод вимагає або великої кількості прикладів задачі розпізнавання (із правильними відповідями), або спеціальної структури нейронної мережі, що враховує специфіку даної задачі.

### **1.2. Приклади задач розпізнавання образів**

Існує багато задач практично важливих задач розпізнавання. У наступному переліку наведено деякі з них:

- Розпізнавання літер.
- Розпізнавання штрих-кодів.
- Розпізнавання автомобільних номерів.
- Розпізнавання осіб.
- Розпізнавання мови.
- Розпізнавання зображень.
- Розпізнавання локальних ділянок земної кори, у яких знаходяться родовища корисних копалин.

*Розпізнавання образів застосовується в наступних областях:*

- Біоінформатика: пошук шаблонів в ДНК.
- Бази даних: пошук і класифікація .
- Обробка текстів: тематична класифікація .
- Аналіз зображень: розпізнавання символів, робота з картами, розпізнавання осіб, поділ об'єктів.
- Виробництво: контроль якості (візуальна перевірка коректності мікросхем).
- Пошук по мультимедіа: визначення жанрів.
- Біометрія: ідентифікація людини за відбитками пальців, по райдужній оболонці ока .
- Прогнозування: погода, сейсмологія, геологія.
- Обробка мови: переклад аудіо-сигналів в текст.

Повернемося до означення. *Розпізнавання образів* — процес віднесення об'єкта з фіксованою групою його властивостей до одного об'єкту з множини образів за заздалегідь обумовленим правилом. Наприклад, рибу у тенетах треба поділити на окунів і лососів. Припустимо, що це робиться по довжині риби. Тобто у нас є об'єкт "риба", і за значенням властивості "довжина" ми відносимо рибу або до образу "лосось", або до образу "окунь".

### **1.3. Процедура розпізнавання**

Виділимо найбільш важливі кроки у процедурі розпізнавання:

1. Сприйняття образу.  
На цьому етапі проводять отримання значень характеристичних властивостей об'єкта (вимірювання лінійних вимірів, фотографування, оцифровка звуку).
2. Попередня обробка.  
Видалення шумів, представлення зображення в чорно-білому варіанті, обрізання непотрібних частин зображення.
3. Виділення характеристик (індексація).  
На цьому етапі вимірюються характеристичні властивості об'єкта (вимірюємо довжину риби та її колір).
4. Класифікація (прийняття рішення).

### **1.4. Розробка системи розпізнавання**

При розробці систем розпізнавання виділяють наступні етапи:

1. Отримати навчальну вибірку (тренувальну колекцію) (навчальна вибірка — множина об'єктів, для яких відомі їхні образи.  
Наприклад колекція аудіо записів для кожного звуку, або колекція зображень кожної букви латинського алфавіту).
2. Вибрати модель представлення об'єктів.
3. Вибрати значущі характеристики.  
Це один з найважливіших етапів розробки системи розпізнавання. Наприклад, якщо у випадку ідентифікації риби окунь / лосось в якості

характеристики вибрати тільки довжину риби, то ніяке правило класифікації не зможе точно визначити тип риби, оскільки можна зустріти лосося і окуня однакової довжини.

4. Розробити правило класифікації.

Правило класифікації — правило, яке за значеннями характеристичних властивостей об'єкта відносить його до одного з образів.

5. Навчання розпізнавальної системи.

На цьому етапі *алгоритм навчання* "збирає досвід" на основі розпізнавання навчальної вибірки. Для того, щоб правильно виставити коефіцієнти (параметри) розпізнавальної системи, алгоритм навчання застосовують на навчальній вибірці, контролюючи результат роботи алгоритму .

6. Перевірка якості навчання. Повернення назад до кроків 2 (3, 4) ...

Якщо частота помилок алгоритму не влаштовує, то необхідно повернутися до п. 2, 3, 4. Інтуїтивно зрозуміло, що збільшення кількості характеристичних властивостей, збільшення тренувальної колекції покращують якість роботи алгоритму.

7. Оптимізація розпізнавальної системи.

Після того, як якість роботи системи розпізнавання підходить під умову розглянутої задачі, іноді доводиться провести його оптимізацію. Початковий алгоритм розпізнавання може бути занадто довгим або ресурсоємним. Прискорити процедуру розпізнавання можна зменшивши кількість характеристичних властивостей об'єкта, вибравши інші характеристичні властивості, використовуючи інше правило класифікації.

### 1.5. Перетворення зорових образів у цифровий код

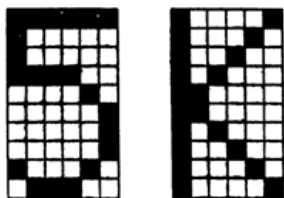


Рис. 3

Зображення розташовується у деякому прямокутному полі, розбитому на однакові квадрати (елементи поля або рецептори). Квадрати, на які потрапило зображення, цілком зафарбовуються у чорний колір і позначаються одиницею. Інші елементи позначаються нулем. Якщо пронумерувати елементи зліва направо і зверху вниз, то цифри 5 та літери "К" на рис. 3 будуть відповідати коди:

«5» — 11111110000010000011110000001000000100000100000  
 1100010011100  
 «К» — 1000011000101001001010001100001100001010001001  
 00100010100001.

У загальному випадку код фігури записується у вигляді  $x_1, x_2, \dots, x_n$ , де  $x_i \in \{0,1\}$ ,  $i = \overline{1, n}$ ,  $n$  — кількість елементів поля рецепторів.

Точка у просторі рецепторів називається *граничною точкою* множини, якщо у її кодї є хоча б одна цифра, зміна якої переводить точку у іншу множину. У протилежному випадку точка називається *внутрішньою*.

Якщо для множини точок виконуються умови

- а) число граничних точок невелике порівняно із їх загальним числом,

- б) довільні дві внутрішні точки множини можуть бути з'єднані достатньо плавною лінією, яка проходить лише через внутрішні точки,
- в) майже кожна внутрішня точка має достатньо великий окіл, який цілком міститься у множині,

то така множина точок називається *компактною множиною*.

Описаний метод кодування не враховує колір і тому придатний лише для чорно-білих зображень. У загальному випадку кожний розряд коду може приймати значення з деякого відрізка  $[a, b]$ .

**Гіпотеза компактності образів:** Простому зоровому образу відповідає компактна множина точок у просторі рецепторів. Поняття компактності проілюстровано на рис. 4.

Гіпотеза компактності дає змогу звести задачу навчання розпізнавання образів до задачі побудови відокремлюючих поверхонь по відомим точкам кожного образу. У результаті простір рецепторів розбивається на кілька областей, про кожну з яких відомо, що вона переважно містить точки, які відповідають певному образу. Якщо тепер потрібно розпізнати нову невідому фігуру, то потрібно перетворити цю фігуру у точку простору рецепторів і визначити, у якій частині простору цю точку розташована.

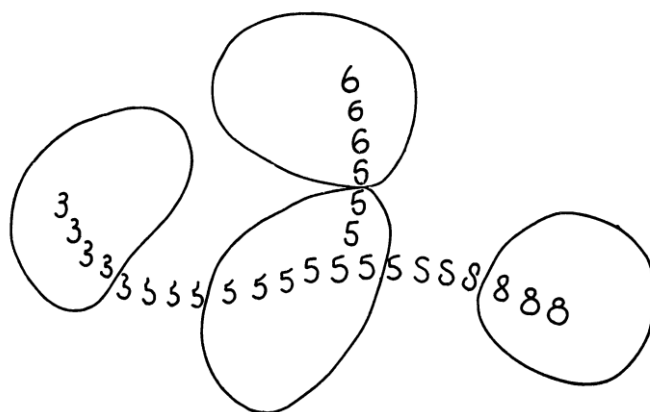


Рис. 4.

### 1.6. Метод нормалізації плоских зображень

Зображення об'єктів часто отримуються у вигляді проєкцій. Із множини усіх проєкцій можна виділити підмножини  $R_1, R_2, \dots$ , для яких виконується наступна закономірність: усі елементи цих множин можна отримати у результаті деяких квазіізоморфних перетворень однієї проєкції. Ця проєкція називається *стандартною*, а загадані вище перетворення — допустимими.

Одним із підходів, який дозволяє ототожнювати проєкції підмножин  $R_1, R_2, \dots$  полягає у *нормалізації* — зведені довільної проєкції  $Z \in R_i$  до стандартної проєкції  $Z^0 \in R_i$ ,  $i = 1, 2, \dots$ , тобто

$$H_{\Gamma}(X) = Z^0,$$



де  $H$  — оператор нормалізації,  $\Gamma$  — множина груп допустимих перетворень. Останні можуть бути поділені на групи  $\Gamma_1$  — трансляцій,  $\Gamma_2$  — обертань,  $\Gamma_3$  — гомотетій.

Довільне бінарне зображення на площині однозначно описується функцією яскравості  $B(\mathbf{z})$ , де  $\mathbf{z} = (x, y)^T$ . Тоді нормалізація, яка проводиться по групам перетворень  $\Gamma_1$ - $\Gamma_3$ , описується наступним чином:

$$\mathbf{z}_0 = M_k M_\psi (\mathbf{z} - \mathbf{z}_c),$$

де

$$\mathbf{z}_c = (x_c, y_c), M_\psi = \begin{pmatrix} \cos \psi & -\sin \psi \\ \sin \psi & \cos \psi \end{pmatrix}, M_k = \begin{pmatrix} k & 0 \\ 0 & k \end{pmatrix},$$

де  $\mathbf{z}_c$  — вектор перетворень на етапі 1,  $M_\psi, M_k$  — матриці перетворень на етапах 2-3 нормалізації.

Вектор  $\mathbf{z}_c$  характеризує паралельний зсув проекції відносно стандартної проекції (центрування). Вважають, що цент мас стандартної системи співпадає із початком координат. Тоді

$$x_c = \frac{m_{10}}{m_{00}}, y_c = \frac{m_{01}}{m_{00}}$$

де  $m_{01}, m_{10}$  — моменти зображення першого порядку.

Моменти порядку  $\alpha + \beta$  обчислюються за формулою

$$m_{\alpha\beta} = \sum_i \sum_j B(x_i, y_j) x_i^\alpha y_j^\beta.$$

З останньої формули випливає, що  $m_{00}$  — кількість забарвлених точок зображення.

У результаті центрування проекція буде описуватися зміщеною функцією яскравості  $B_1(x, y) = B(x - x_c, y - y_c)$ . Довільний центральний момент порядку  $\alpha + \beta$  позначимо  $\mu_{\alpha\beta}$ .

Для однозначності орієнтування доцільно фіксувати на центрованому зображенні деякий вектор  $\psi$  з початком у точці  $O$ . Будемо вимагати, щоб для стандартного зображення цей вектор співпадав із віссю  $OX$ . Алгоритм пошуку вектора  $\psi$  має бути інваріантним відносно елементів груп перетворень  $\Gamma_2$  та  $\Gamma_3$ . Для орієнтації використовуються центральні моменти другого порядку. При повороті центрованого зображення навколо початку координат на кут  $\psi$  проти годинникової стрілки центральні моменти другого порядку наступним чином виражаються через початкові моменти  $\mu'_{20}, \mu'_{02}, \mu'_{11}$ :

$$\begin{aligned} \mu_{20}(\psi) &= \mu'_{20} \cos^2 \psi + \mu'_{02} \sin^2 \psi - \mu'_{11} \sin 2\psi \\ \mu_{02}(\psi) &= \mu'_{20} \sin^2 \psi + \mu'_{02} \cos^2 \psi + \mu'_{11} \sin 2\psi \\ \mu_{11}(\psi) &= 0,5 (\mu'_{20} - \mu'_{02}) \sin 2\psi + \mu'_{11} \cos 2\psi \end{aligned}$$

Для останньої залежності можна вказати такий кут  $\psi_0$ , для якого  $\mu_{11} = 0$ . Очевидно, що

$$\psi_0 = 0,5 \operatorname{arctg} \frac{2\mu'_{11}}{\mu'_{02} - \mu'_{20}} + \nu \frac{\pi}{2} \quad (\nu = 0, 1, 2, \dots).$$

Але виконання умови  $\mu_{11} = 0$  ще не забезпечує однозначності орієнтування, оскільки при повороті зображення на кут  $2\pi$  є чотири положення, для яких  $\mu_{11} = 0$ . Можна показати, що якщо  $\mu_{11} = 0$ , то один моментів  $\mu_{20}(\psi)$ ,  $\mu_{02}(\psi)$  набуває своє найбільше значення, а інший — найменше. Тому, якщо доповнити умову  $\mu_{11} = 0$  умовою  $\mu_{20}(\psi) > \mu_{02}(\psi)$ , то множина усіх можливих орієнтацій може бути зведена до двох, які відрізняються на кут  $\pi$ .

Для однозначного орієнтування необхідна така характеристика, яка різко змінюється при повороті на кут  $\pi$ . Таким є момент  $\mu_{30}(\psi)$ , який обчислюється наступним чином:

$$\mu_{30}(\psi) = \mu'_{30} \cos^3 \psi - 2\mu'_{21} \cos^2 \psi \sin \psi + 2\mu'_{12} \cos \psi \sin^2 \psi - \mu'_{03} \sin^3 \psi,$$

де початкові центральні моменти позначені штрихом. Із останнього співвідношення випливає, що  $\mu_{30}(\pi + \psi) = -\mu_{30}(\psi)$ . Тому, вважаючи, що стандартному зображенню відповідає значення  $\mu_{30} > 0$ , отримуємо наступні умови для однозначної орієнтації проєкції:

$$\mu_{11} = 0, \mu_{20}(\psi) > \mu_{02}(\psi), \mu_{30} > 0. \quad (1)$$

Після знаходження координат  $x_c$  та  $y_c$  для обчислення моментів можна скористатися формулою:

$$\mu_{\alpha\beta} = \sum_{i=1}^{M_c} \left\{ \left[ \sum_{\nu=0}^{\alpha} (-1)^\nu C_\alpha^\nu x_i^{\alpha-\nu} x_c^\nu \right] \left[ \sum_{\xi=0}^{\beta} (-1)^\xi C_\beta^\xi y_i^{\beta-\xi} y_c^\xi \right] \right\},$$

де  $x_i$  та  $y_i$  — координати  $i$ -го елемента зображення до центрування.

Для зображень, які мають центр симетрії вище, ніж другого порядку, умова (1) не є застосовною, оскільки  $\mu_{11}(\psi) \equiv 0$ ,  $\mu_{20}(\psi) = \mu_{02}(\psi)$ . Для таких зображень застосовують інші умови, наведені у [8].

Для визначення коефіцієнту  $k$  використовують масштабування по площі. Тоді якщо  $S_c^*$  та  $S_c$  — площі відповідно стандартного зображення та зображення, яке потребує нормалізацію, то

$$k = \sqrt{\frac{S_c^*}{S_c}}.$$

## 2. Методи розпізнавання

Виділяють 4 групи методів розпізнавання [1, 5]:

### 1. Порівняння із зразком.

Застосовуємо геометричну нормалізацію і вважаємо відстань до прототипу. Найбільш наочно застосування цього методу в розпізнаванні тексту.

Завдання. У нас є зображення відсканованого символу і колекція зображень зразків (усіх букв абетки), ми хочемо визначити, якій букві алфавіту відповідає відскановане зображення. Розв'язок. Масштабуємо зображення символу до розмірів зразків і виберемо той, відстань до якого мінімальна.

### 2. Нейронні мережі.

Вибираємо вид мережі і налаштовуємо коефіцієнти. На вхід нейронної мережі подається об'єкт для розпізнавання. Група рецепторів мережі відповідає за прийом своєї характеристичної властивості.

### 3. Статистичні методи.

### 4. Структурні та синтаксичні методи.

Розбираємо об'єкт на елементи. Будуємо правило, в залежності від входження окремих елементів та їх послідовностей.

**Приклад.** Припустимо, нам необхідно визначити вид риби, поданої на вхід. Спростимо задачу до випадку 2 видів (окунь і лосось). Перший розв'язок, який приходить в голову — зробити припущення про вигляді риби по її довжині, як це наведено на рис. 5.

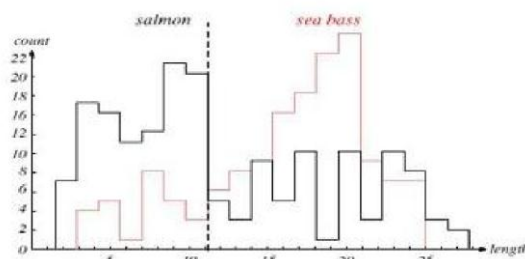


Рис. 5.

Інша характеристична властивість — забарвлення. Керуючись лише однією характеристикою не можливо точно визначити вид риби.

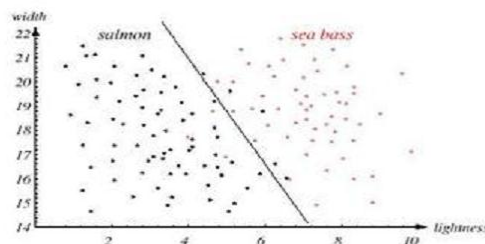


Рис. 6.

Але керуючись обома ознаками, можна визначати вид риби більш точно з використанням гіперплощини (рис. 6).

### 2.1. Метод січних площин

Алгоритм навчання, заснований на методі січних площин, полягає у відокремленні образів за допомогою частин гіперплощин. Алгоритм складається з наступних етапів [9].

1. Навчання (формування гіперплощин для відокремлення множин):
  - a. проведення січних площин;
  - b. вилучення зайвих гіперплощин;
  - c. вилучення зайвих частин площини.
2. Розпізнавання нових об'єктів.

Припустимо, що потрібно навчити комп'ютер розпізнавати 3 образи  $a$ ,  $b$ , та  $c$ .

#### Проведення січних площин

У комп'ютер вводять коди двох точок, які належать різним образам та проводять довільну пряму, яка їх відокремлює (рис. 7).

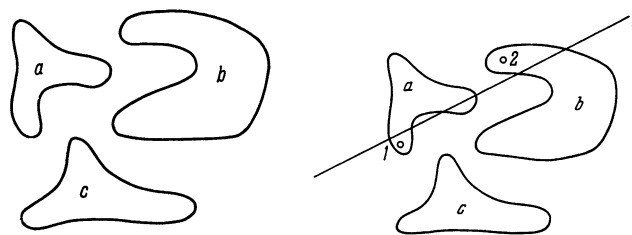


Рис. 7.

Далі береться третій об'єкт і перевіряємо, правильність класифікації відносно проведеної прямої. При перевірці нової точки можливі 3 випадки (рис. 8):

- 1) виникає протиріччя (точка 3) ;
- 2) протиріччя не виникає, точка потрапляє у свою частину простору;
- 3) протиріччя не виникає, точка потрапляє у вільну (непозначену) частину простору (точки 4 і 5);

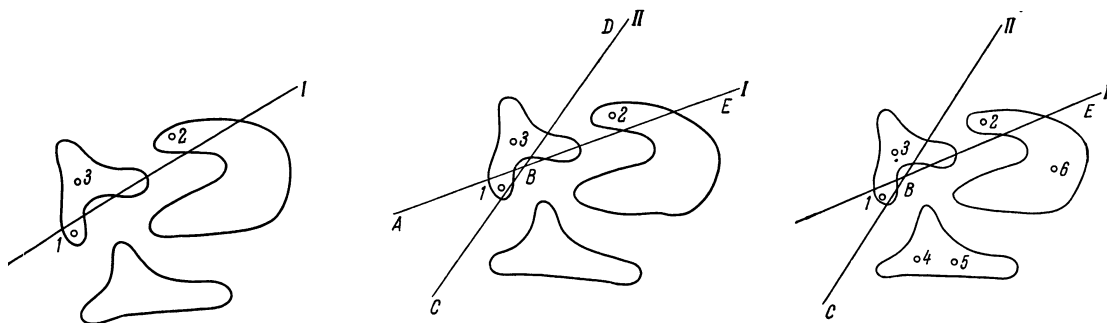


Рис. 8.

Для відокремлення точки 6 потрібно провести дві нові прямі (рис. 9):

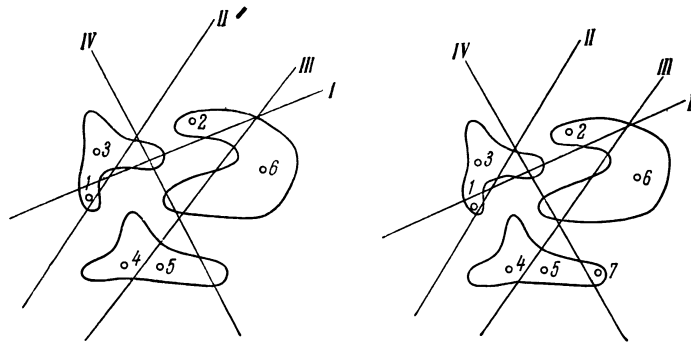


Рис. 9.

Класифікація точки 7 суперечить класифікації точки 6, як це видно з рис. 10. а. Для їх відокремлення потрібно провести площину  $V^1$ .

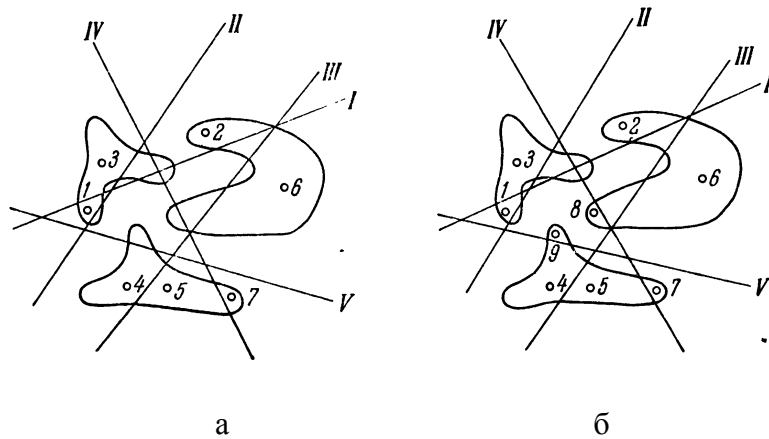


Рис. 10

Точка 8 потрапляє у незаповнену частину простору, але потім "конфліктує" з точкою 9. Протиріччя усуває пряма VI наведена на рис. 11 а. Поява десятої точки, яка "суперечить" точці 8, призводить до потреби використання прямої VII (рис 11. б).

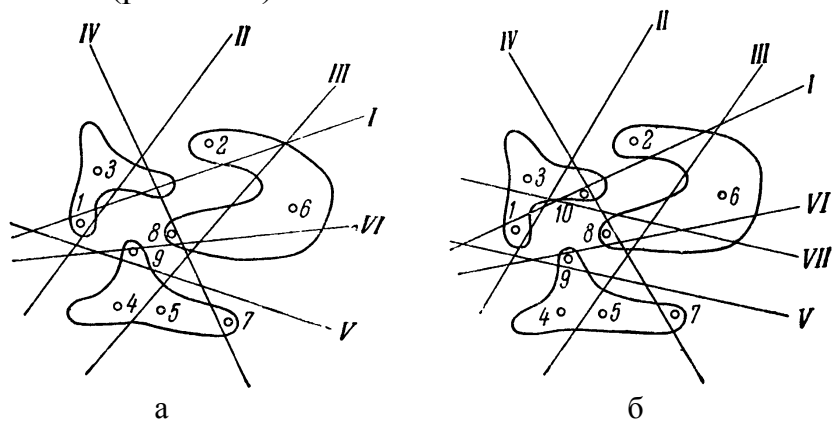


Рис. 11

<sup>1</sup> У нашого прикладі у якості гіперплощин у двовимірному просторі виступають прямі на площині

Кінцевий результат стадії а) процесу навчання наведений на рис. 12.

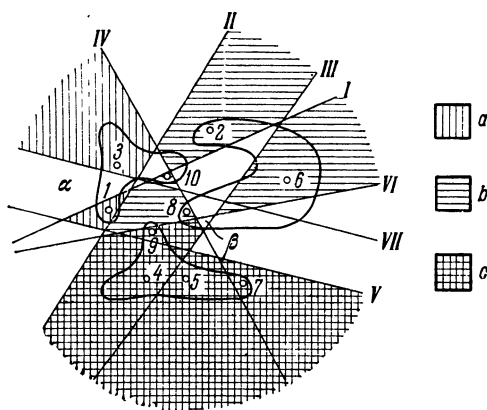


Рис. 12.

### Видалення зайвих площин.

З рис. 12 видно, що площини I, III та V можуть бути вилученні цілком, оскільки вони не містять частин, відкидання яких призводить до появи протиріч. Результат наведений на рис. 13.

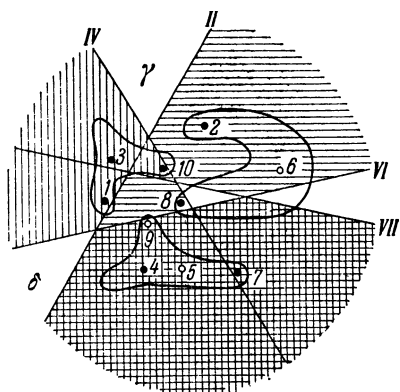


Рис. 13.

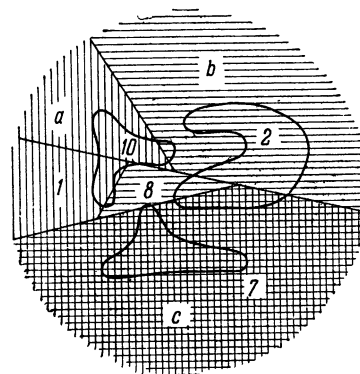


Рис. 14.

Видалення зайвих частин площин. Перевірка усіх частин площини II, потім усіх частин площини IV і т.д. Результат наведений на рис 14.

### Опис алгоритму

Випадковим чином вибираємо  $n$  чисел  $\lambda_1, \dots, \lambda_n$ . Далі для координат відповідних точок обчислюємо значення сум

$$\sigma^{(1)} = \sum \lambda_i x_i^{(1)}, \quad \sigma^{(2)} = \sum \lambda_i x_i^{(2)}.$$

Після цього вибирається число  $\lambda_{n+1} = (\sigma^{(1)} + \sigma^{(2)}) / 2$  (вважаємо, що значення сум різні). Тоді рівняння

$$\lambda_1 x_1 + \dots + \lambda_n x_n - \lambda_{n+1} = 0.$$

Будемо позначати положення точки відносно площини знаком відхилення

$$\lambda(\mathbf{x}) = \lambda_1 x_1 + \dots + \lambda_n x_n - \lambda_{n+1}.$$

Далі процес описується за допомогою таблиць знаків (1 відповідає додатному відхиленню, 0 — від'ємному): Початкові знаки (для перших двох точок) наведені у табл. 1.

Таблиця 1

Номер точки	Образ	Номер площини	
		I	
		Знак точки	
1	$a$	0	
2	$b$	1	

Далі обчислюється знак точки 3:

Таблиця 2

Номер точки	Образ	Номер площини	
		I	
		Знак точки	
1	$a$	0	
2	$b$	1	
3	$a$	1	

Поява третьої точки зумовлює необхідність використання прямої (гіпер-площини) II. Тому у табл. 3 з'являється новий стовпчик, для опису значень відхилень відносно II прямої. За рахунок цього протиріччя зникає.

Таблиця 3

Номер точки	Образ	Номер площини	
		I	II
		Знак точки	
1	$a$	0	1
2	$b$	1	0
3	$a$	1	1

Обчислення знаків відхилень точок 4, 5 та 6 наведено у табл. 4. Для правильної класифікації потрібно провести ще дві прямі, як це наведено у табл. 5.

Таблиця 4

Номер точки	Образ	Номер площини		
		I	II	
		Знак точки		
1	<i>a</i>	0	1	
2	<i>b</i>	1	0	
3	<i>a</i>	1	1	
4	<i>c</i>	0	0	
5	<i>c</i>	0	0	
6	<i>b</i>	0	0	

Таблиця 5

Номер точки	Образ	Номер площини			
		I	II		
		Знак точки			
1	<i>a</i>	0	1	1	1
2	<i>b</i>	1	0	1	0
3	<i>a</i>	1	1	1	1
4	<i>c</i>	0	0	1	1
5	<i>c</i>	0	0	0	1
6	<i>b</i>	0	0	0	0

Надходження нових точок зумовлює потребу у розширенні таблиці знаків. При цьому використовують ті самі принципи заповнення, що й при побудові таблиць 1-5. Кінцева таблиця після завершення першої стадії алгоритму для десяти вхідних об'єктів має наступний вигляд:

Далі переходимо на стадію видалення зайвих площин. Перевіряємо, чи є перша площина суттєвою. Для цього розглядаємо таблицю знаків, які містить усі стовпці крім стовпця, який відповідає площині I. Для отриманої таблиці протиріччя не виникають. Отже, перша площина є зайвою і може бути видаленою. Такі самі перевірки виконуємо для площин II, III і т. д (при перевірці на кожному кроці використовується таблиця, яка отримується після попереднього кроку). Для нашого прикладу зайвими виявляються площини I, III та V. Після завершення другого етапу отримуємо таблицю 7.



Таблиця 6

Номер точки	Образ	Номер площини						
		I	II	III	IV	V	VI	VII
		Знак точки						
1	<i>a</i>	0	1	1	1	0	1	1
2	<i>b</i>	1	0	1	0	0	1	0
3	<i>a</i>	1	1	1	1	0	1	0
4	<i>c</i>	0	0	1	1	1	0	1
5	<i>c</i>	0	0	0	1	1	0	1
6	<i>b</i>	0	0	0	0	0	1	0
7	<i>c</i>	0	0	0	0	1	0	1
8	<i>b</i>	0	0	1	1	0	1	1
9	<i>c</i>	0	0	1	1	0	0	1
10	<i>a</i>	0	0	1	1	0	1	0

Таблиця 7

Номер точки	Образ	Номер площини						
		II	IV	VI	VII			
		Знак точки						
1	<i>a</i>	1	1	1	1			
2	<i>b</i>	0	0	1	0			
3	<i>a</i>	1	1	1	0			
4	<i>c</i>	0	1	0	1			
5	<i>c</i>	0	1	0	1			
6	<i>b</i>	0	0	1	0			
7	<i>c</i>	0	0	0	1			
8	<i>b</i>	0	1	1	1			
9	<i>c</i>	0	1	0	1			
10	<i>a</i>	0	1	1	0			

У таблиці 7 є однакові рядки знаків. Після їх відкидання отримуємо табл. 8.

Після цього розпізнавання нових об'єктів можна виконувати наступним чином: обчислюються знаки об'єкту відносно усіх гіперплощин і отриманий код порівнюється з усіма рядками таблиці знаків. У випадку співпадання рядків об'єкт відноситься до відповідного образу.

Таблиця 8

Номер точки	Образ	Номер площини					
		II	IV	VI	VII		
		Знак точки					
1	<i>a</i>	1	1	1	1		
2	<i>b</i>	0	0	1	0		
3	<i>a</i>	1	1	1	0		
4	<i>c</i>	0	1	0	1		
7	<i>c</i>	0	0	0	1		
8	<i>b</i>	0	1	1	1		
10	<i>a</i>	0	1	1	0		

## 2.2. Метод потенціалів для розпізнавання образів

Точковий електричний заряд у однорідному середовищі створює електричне поле, зображене на рис. 15. Радіальні лінії — це силові лінії поля, концентричні кола — лінії однакового потенціалу.

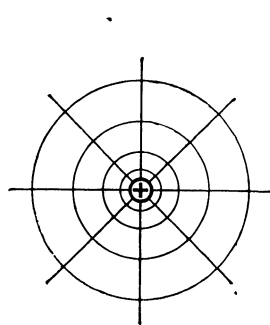


Рис. 15.

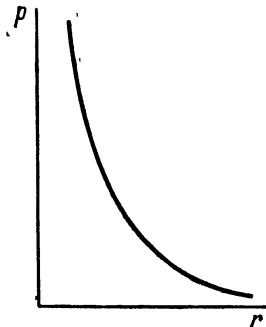


Рис. 16.

Потенціал  $p$  у кожній точці простору визначається співвідношенням

$$p = a \frac{q}{r^2},$$

де  $a$  — деяка стала,  $q$  — величина заряду,  $r$  — відстань від заданої точки до заряду.

Крива зміни потенціалу як функції відстані наведена на рис. 16. Потенціал зменшується по мірі віддалення від його джерела. Якщо поле утворене кількома зарядами, то потенціал в кожній точці рівний сумі потенціалів, які створюються кожним із зарядів.

Припустимо, що у просторі розташовані дві компактні групи зарядів. У одній групі — позитивні, у другій — негативні. На рис. 17 показаний розподіл потенціалів у околі цих зарядів, на рис. 18 ці потенціали алгебраїчно просумовані.

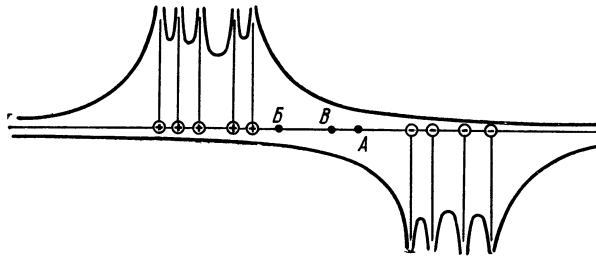


Рис. 17.

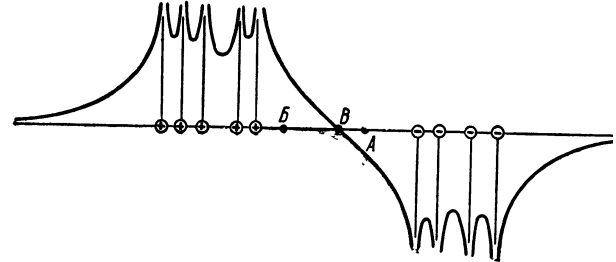


Рис. 18.

Точку можна віднести до тієї чи іншої множини точок у залежності від того, який знак має сумарний потенціал поля у цій точці.

Вищенаведені міркування по аналогії можна перенести на точки простору рецепторів. Кожній точці, яка з'являється у процесі навчання, поставимо у відповідність деяку функцію, аналогічну по формі до електричного потенціалу. Такою функцією може бути, наприклад, функція

$$\varphi(R) = \frac{1}{1 + \alpha R^2},$$

де  $\alpha$  — деякий коефіцієнт,  $R$  — відстань (у деякій метриці) між точкою-джерелом потенціалу та точкою, у якій обчислюється потенціал. Наприклад, у якості  $R$  можна використовувати евклідову відстань, віддаль Хеммінга, манхеттенську метрику тощо.

Нехай джерелами потенціалів є група точок, які відповідають деякому образу  $a$ . Тоді можна вважати, що середній потенціал, які створюють у деякій точці простору джерела цього образу, характеризує віддаль від цієї точки до усього образу у цілому.

Спираючись на гіпотезу компактності, можна запропонувати наступне правило розпізнавання: точку відносимо до того образу, середній потенціал якого у цій точці є максимальним.

### Алгоритм розпізнавання на основі методу потенціалів

1. У процесі навчання запам'ятовуються координати усіх точок та належність їх до відповідних образів  $a_1, \dots, a_m$ .
2. Розпізнавання.
  - а. Для точки  $\mathbf{x}$ , яка підлягає розпізнаванню, обчислюється потенціали кожного образу, тобто суми

$$\Phi_i(\mathbf{x}) = \frac{1}{n_i} \sum_{\mathbf{a} \in a_i} \varphi_{\mathbf{a}}(\mathbf{x}), \quad i = 1, 2, \dots, m,$$

де  $m$  — кількість різних образів,  $n_i$  — кількість точок відповідного образу, використаних у процесі навчання,  $\varphi_a(\mathbf{x}) = \frac{1}{1 + \alpha R^2(\mathbf{a}, \mathbf{x})}$  — потенціал, який

створює точка  $\mathbf{a}$  у точці  $\mathbf{x}$ .

б. Порівнюються величини  $\Phi_1(\mathbf{x}), \Phi_2(\mathbf{x}), \dots, \Phi_m(\mathbf{x})$  і точка  $\mathbf{x}$  відноситься до того образу, потенціал якого є найбільшим.

У випадку двох образів  $a_1$  та  $a_2$  рішення можна приймати на основі значення знаку функції  $\Delta\Phi(\mathbf{x}) = \Phi_{a_1}(\mathbf{x}) - \Phi_{a_2}(\mathbf{x})$ .

Застосування вищенаведеного алгоритму для розпізнавання цифр, зображених за допомогою рецепторного поля  $6 \times 10$  у середньому дало змогу досягнути правильного розпізнавання у 85% випадків за умови, що  $n_1 = n_2 = \dots = n_{10} = 12$ , причому подальше збільшення обсягу навчальної вибірки практично не впливає на якість розпізнавання.

Недоліки найпростішого алгоритму виявляються у тому випадку, коли точки навчальної вибірки розподілені нерівномірно. Відповідний приклад наведений на рис. 19-21.

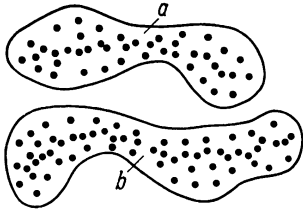


Рис. 19.

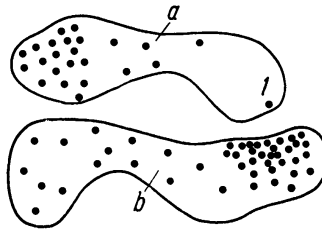


Рис. 20.

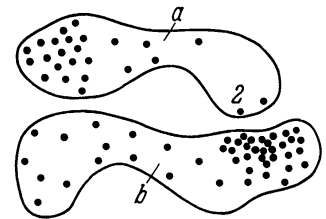


Рис. 21.

Навчальна вибірка для образів  $a$  та  $b$ , зображених на рис. 19, наведена на рис. 20. У цьому випадку у зв'язку із «невдалою» навчальною вибіркою, нова точка 2 (рис. 21) буде віднесена до образу  $b$ . Більш того, машина може не впізнати навіть відомий їй об'єкт, який зустрічався у процесі навчання, наприклад, точку 1, яка належить до образу  $a$ .

### Покращена модифікація алгоритму навчання

Алгоритм розпізнавання може бути покращений, шляхом введення поняття «ваги точки» та доповненням кроку навчання наступною операцією. Після запам'ятовування усіх об'єктів їм присвоюють початкову вагу 1. Далі до елементів навчальної вибірки застосовують крок 2 алгоритму навчання. Якщо відбувається помилка, то вагу відповідного об'єкту збільшують на деяку величину, наприклад на одиницю. Потім застосовується ще один такий самий цикл перевірки та корекції ваг і т.д. Цикли повторюються до тих пір, поки усі відомі фігури не будуть розпізнані правильно.

Застосування покращеного алгоритму до розпізнавання дало змогу підвищити відсоток правильного розпізнавання нових зображень цифр до 89%.

**Удосконалення кодування графічних об'єктів.** У випадку бінарних зображень традиційні функції відстані Хеммінга або Евкліда не завжди точно відображають відмінність між зображеннями. Відповідний приклад наведений на рис. 22. Шляхом зсуву вертикальної лінії із п'ятірки (рис. 22, а), можна отримати «трохи гіршу» п'ятірку (рис. 22, б) та трійку (рис. 22, в), кожна з яких відрізняється від початкової фігури у шести розрядах. Тобто, елемент «свого» образу та елемент «чужого» образу знаходяться на однаковій відстані від початкової фігури. Схожий приклад наведений на рис. 23 для випадку зображень одиниць та двійок, які відрізняються у десяти розрядах.

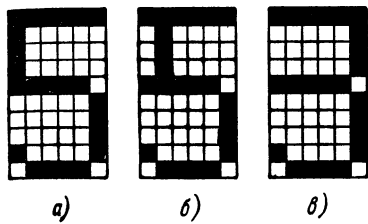


Рис. 22.

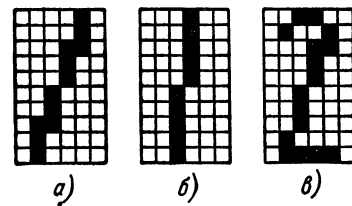


Рис. 23.

Ці приклади показують, що значні зміни елементів зображень, які значно спотворюють зображення, приводять тих самих змін у відстані, що і незначні. Тому застосований метод кодування недостатньо добре передає відмінність фігур.

Надійність розпізнавання може бути підвищена шляхом удосконалення методу кодування. Поставимо у відповідність кожному рецептору функцію, яка рівна одиниці на цьому рецепторі, спадаючу по усім напрямкам від нього, тобто функцію, аналогічну до потенціалу (рис. 24, а). Ця функція може бути апроксимована ступінчастими функціями (рис. 24, б-в).

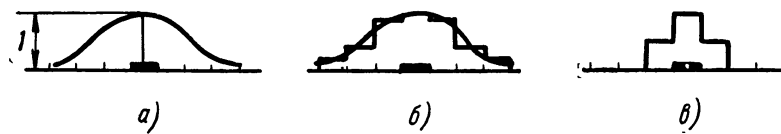


Рис. 24.

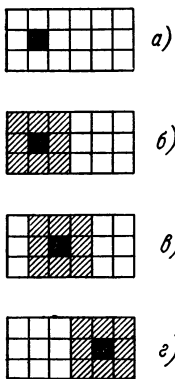


Рис. 25.

У випадку «грубої апроксимації» функція рівна одиниці на відповідному елементів рецепторного поля,  $\frac{1}{2}$  — на сусідніх елементах, нулю — на всіх інших. Для прикладу розглянемо одно кліткову фігуру, зображену на рис. 25, а. На рис. 24, б зображено значення ступінчастої функції відстані.

Код такої фігури має вигляд

$$\frac{1}{2} \frac{1}{2} \frac{1}{2} 000 \frac{1}{2} 1 \frac{1}{2} 000 \frac{1}{2} \frac{1}{2} \frac{1}{2} 000.$$

Посуємо фігуру на один елемент управо (рис. 25, в). Отримаємо фігуру із кодом

$$0 \frac{1}{2} \frac{1}{2} \frac{1}{2} 000 \frac{1}{2} 1 \frac{1}{2} 000 \frac{1}{2} \frac{1}{2} \frac{1}{2} 00.$$

Евклідова відстань між цими фігурами рівна  $\sqrt{2}$ . Тепер посуємо фігуру іще на дві клітинки вправо. Отримаємо фігуру (рис. 25, в) із кодом

$$000 \frac{1}{2} \frac{1}{2} \frac{1}{2} 000 \frac{1}{2} 1 \frac{1}{2} 000 \frac{1}{2} \frac{1}{2} \frac{1}{2}.$$

Відстань від цієї фігури до початкової рівна  $\sqrt{6}$ . Як видно із прикладів, при такому методі кодування "більш віддаленим" фігурам відповідає більша відстань у просторі рецепторів.

Описаний вище прийом можна застосувати для більш складних фігур з використанням сумарних потенціалів рецепторів. На рис. 26 зображено кодування цифр із прикладу, наведеного на рис. 22.

2	2,5	2	2	2	1,5
2,5	2,5	1,5	1,5	1,5	1
2	1,5	0	0	0	0
2,5	2,5	1,5	1,5	1	0,5
2	2,5	2	2	2	1
1	1,5	1,5	1,5	2	2
0	0	0	0	1,5	2
0,5	0,5	0	0	1,5	2
1,5	1,5	1,5	1,5	2	2
1	2	2	2	2	1

2	2,5	2,5	2	2	1,5
2	3	2,5	1,5	1,5	1
1,5	2	1,5	0	0	0
2	3	2,5	1,5	1	0,5
2	2,5	2,5	2	2	1
1	1,5	1,5	1,5	2	2
0	0	0	0	1,5	2
0,5	0,5	0	0	1,5	2
1,5	1,5	1,5	1,5	2	2
1	2	2	2	2	1

1,5	2	2	2	2,5	2
1	1,5	1,5	1,5	2,5	2,5
0	0	0	0	1,5	2
1	1,5	1,5	1,5	2	2
1,5	2	2	2	2	1,5
1	1,5	1,5	1,5	2	2
0	0	0	0	1,5	2
0,5	0,5	0	0	1,5	2
1,5	1,5	1,5	1,5	2	2
1	2	2	2	2	1

Рис. 26.

Тоді відстань між «п'ятірками» рівна 2,25, а відстань між першою п'ятіркою та трійкою — 5,22.

Застосування удосконаленого методу кодування дозволило збільшити середню надійність розпізнавання букв до 94%.

### 2.3. Класифікація образів із використанням функції відстані

У якості критерію класифікації використовується підхід заснований на класифікації образів за критерієм мінімуму відстані.

#### Класифікації образів за критерієм мінімуму відстані

##### 1. Випадок одного еталону. Метод найближчого сусіда.

В деяких задачах об'єкти деяких класів (образів) мають тенденцію до групування навколо деякого об'єкту, який є типовим або репрезентативним для

відповідного образу. Типовим прикладом є задача зчитування банківських чеків, для якої вектори, які відповідають образам кожного класу, будуть майже ідентичними.

Розглянемо  $M$  класів, які допускають зображення за допомогою еталонних представників  $\mathbf{z}_1, \dots, \mathbf{z}_M$ . Евклідова відстань між довільним вектором  $\mathbf{x}$  та цими образами обчислюється за формулою

$$D_i = \|\mathbf{x} - \mathbf{z}_i\| = \sqrt{(\mathbf{x} - \mathbf{z}_i)'(\mathbf{x} - \mathbf{z}_i)}.$$

Вектор  $\mathbf{x}$  відноситься до класу  $\omega_i$ , якщо умова  $D_i < D_j$  виконується для усіх  $j \neq i$ . Перетворимо формулу обчислення відстані

$$\begin{aligned} D_i^2 &= \|\mathbf{x} - \mathbf{z}_i\|^2 = (\mathbf{x} - \mathbf{z}_i)'(\mathbf{x} - \mathbf{z}_i) = \\ &= \mathbf{x}'\mathbf{x} - 2\mathbf{x}'\mathbf{z}_i + \mathbf{z}_i'\mathbf{z}_i = \mathbf{x}'\mathbf{x} - 2\left(\mathbf{x}'\mathbf{z}_i - \frac{1}{2}\mathbf{z}_i'\mathbf{z}_i\right). \end{aligned}$$

Остання формула показує, що вибір мінімальної відстані до класу еквівалентний максимізації величини  $\mathbf{x}'\mathbf{x} - 2\left(\mathbf{x}'\mathbf{z}_i - \frac{1}{2}\mathbf{z}_i'\mathbf{z}_i\right)$ . Тому функції рішень можна визначити як

$$d_i(\mathbf{x}) = \mathbf{x}'\mathbf{z}_i - \left(\frac{1}{2}\right)\mathbf{z}_i'\mathbf{z}_i, \quad i = 1, 2, \dots, M,$$

де  $d_i(\mathbf{x})$  — лінійні функції рішень. Якщо покласти

$$\begin{aligned} w_{ij} &= z_{ij}, \quad j = 1, 2, \dots, n, \\ w_{i, n+1} &= -\left(\frac{1}{2}\right)\mathbf{z}_i'\mathbf{z}_i \end{aligned}$$

і  $\mathbf{x} = (x_1, \dots, x_n, 1)$ , то функції рішень можна записати у вигляді

$$d_i(\mathbf{x}) = \mathbf{w}_i'\mathbf{x}, \quad i = 1, 2, \dots, M,$$

де  $\mathbf{w}_i = (w_{i1}, \dots, w_{in}, w_{i, n+1})$ . На рис. 27 зображена відокремлююча межа для випадку двох класів.

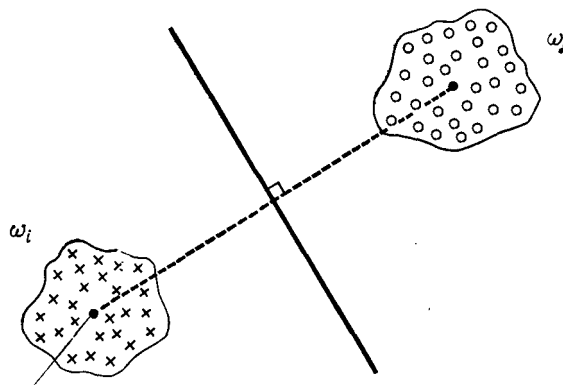


Рис. 27.

Границя між класами є гіперплощиною, яка рівновіддалена від еталонних представників класів. Описаний підхід ще називають кореляцією або співставленням із кластером.

## 2. Випадок кількох еталонів.

У цьому випадку кожний клас  $\omega_i$  можна охарактеризувати еталонами  $z_i^1, \dots, z_i^{N_i}$ ,  $N_i$  — кількість еталонів  $i$ -го класу. Одним із найпростіших підходів є метод "найближчого сусіда" (НС-правило). Відстань між  $i$ -им класом та об'єктом  $x$  можна обчислити за формулою

$$D_i = \min_l \|x - z_i^l\|, \quad l = 1, 2, \dots, N_i;$$

тобто з використанням відстані до найближчого до  $x$  еталона класу. Процедура віднесення нового об'єкта до класу така сама, як і раніше. На рис. 28 показані границі класів у випадку двох класів, які містять по два еталони кожний.

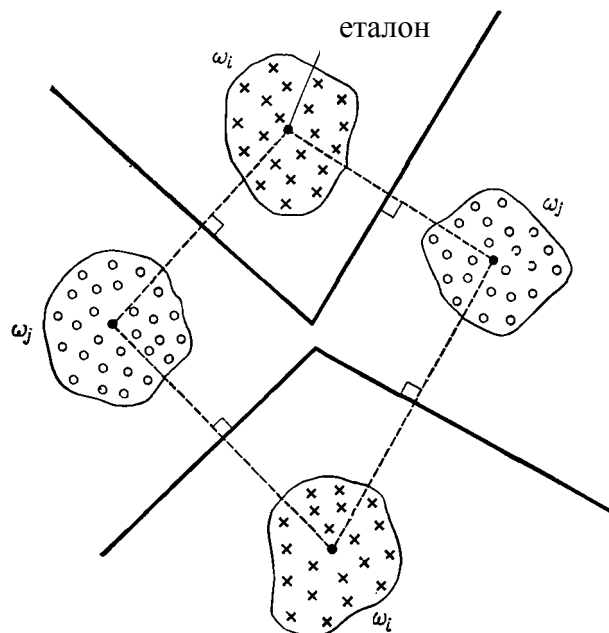


Рис. 28.

Отриманий класифікатор є частинним випадком *кусково-лінійного класифікатора*.

## 3. Узагальнення принципів класифікації по мінімуму відстані. Метод $k$ -найближчих сусідів.

НС-правило може бути узагальнено на  $k$ -НС правило, згідно якого обчислюються  $k$ -найближчих сусідів і об'єкт відноситься до того класу, до якого відноситься найбільша кількість з  $k$ -найближчих еталонів.



### 3. Кластеризація

Кластерний аналіз полягає у поділі заданого набору об'єктів на кластери — групи об'єктів, схожі між собою за тим чи іншим критерієм подібності. Близькими по змісту до терміну "кластер" є поняття "клас" та "таксон". Тому кластеризацію часто називають таксономією або класифікацією без учителя, оскільки, на відміну від задачі "звичайної" класифікації, класи апріорно не задані.

Слід зазначити, що у кластеризації присутні як об'єктивні, так і суб'єктивні моменти. Це добре ілюструється прикладом Богнарда. Фігури, зображені на рис. 29 можна розбити на різні класи у залежності від критерію кластеризації: а) по кольору — на два кластери, по кількості кутів — на три, по розміру — на два або три. Тому результат кластеризації визначається багато у чому завданнями та засобами, які використовують у її процесі

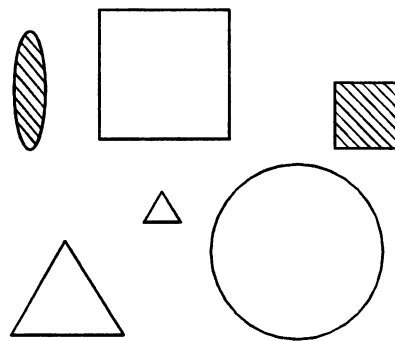


Рис. 29.

**Міри близькості.** Для виділення кластерів потрібно у першу чергу ввести функцію подібності об'єктів. Розглянемо міри близькості у випадку кластеризації об'єктів, заданих у векторній формі. Найбільш часто використовується евклідова відстань. Для булевих векторів використовують відстань Хеммінга. Також часто використовують відстань Махаланобіса

$$d(\mathbf{x}, \mathbf{y}) = \sqrt{(\mathbf{x} - \mathbf{y})^T S^{-1} (\mathbf{x} - \mathbf{y})}$$

де  $S$  — коваріаційна матриця,  $\mathbf{x}, \mathbf{y}$  — об'єкти-змінні.

Можна також використовувати неметричні функції близькості

$$s(\mathbf{x}, \mathbf{z}) = \frac{\mathbf{x}'\mathbf{z}}{\|\mathbf{x}\|\|\mathbf{z}\|}, \quad (2)$$

яка рівна косинусу кута між векторами  $\mathbf{x}$  та  $\mathbf{z}$  (у останній формулі через  $\mathbf{x}'$  позначається вектор-рядок транспонований, до вектора-стовпчика  $\mathbf{x}$ ).

У випадку двійкових векторів величина  $\mathbf{x}'\mathbf{z}$  рівна кількості співпадаючих ознак для об'єктів  $\mathbf{x}$  та  $\mathbf{z}$ . Іншим варіантом формули (2) є міра Танімото

$$s(\mathbf{x}, \mathbf{z}) = \frac{\mathbf{x}'\mathbf{z}}{\|\mathbf{x}\|^2 + \|\mathbf{z}\|^2 - \mathbf{x}'\mathbf{z}}.$$

Слід зазначити, що у зв'язку із неоднорідністю шкал, у яких вимірюються ознаки об'єктів, при розрахунках відстаней може виявитися, що деякі ознаки набагато сильніше впливають на близькість об'єктів, ніж інші. Тому бажано попередньо провести нормалізацію змінних.

**Основні підходи до побудови методів кластеризації.** Методи кластеризації поділяють на ієрархічні та неієрархічні. У ієрархічних методах будується дерево вкладених кластерів (дендограма). Приклад зображення дендограм наведено на рис. 30.

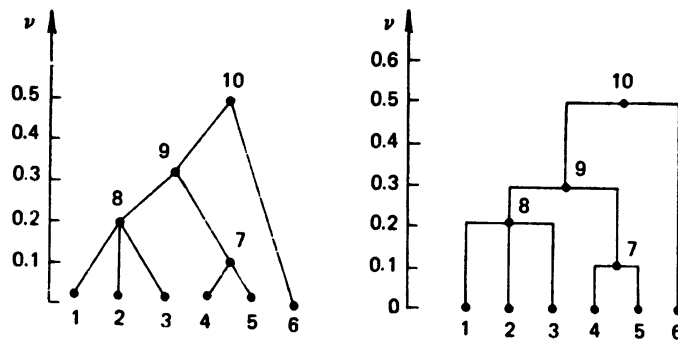


Рис. 30. Різні способи зображення дендограм

Ієрархічні методи поділяються на агломеративні методи та методи поділу.

Агломеративні методи полягають у послідовному об'єднанні початкових кластерів (побудова кластерів знизу вгору). Методи поділу — у розщепленні існуючих кластерів (побудова кластерів зверху вниз).

### 3.1. Ієрархічні алгоритми кластеризації

Нехай  $X$  — скінченна множина, яку потрібно кластеризувати. На першому кроці кожному елементу множини відповідає одноелементний кластер. На кожному наступному кроці вибираються два найближчі між собою кластери  $C_i$  та  $C_j$ , які об'єднуються у новий кластер  $C_i \cup C_j$ . Після кожного такого кроку кількість кластерів зменшується. У кінці процедури отримується кластер, який складається із одного об'єкта.

Для оцінки відстані  $d(C_i, C_j)$  між двома множинами використовують наступні функції:

- а) Відстань між найближчими об'єктами множин

$$d(C_i, C_j) = \min \{d(\mathbf{x}, \mathbf{y}) \mid \mathbf{x} \in C_i, \mathbf{y} \in C_j\}.$$

- б) Відстань між найбільш віддаленими об'єктами.

$$d(C_i, C_j) = \max \{d(\mathbf{x}, \mathbf{y}) \mid \mathbf{x} \in C_i, \mathbf{y} \in C_j\}.$$

в) Середня відстань між множинами

$$d(C_i, C_j) = \frac{1}{|C_i||C_j|} \sum_{\mathbf{x} \in C_i} \sum_{\mathbf{y} \in C_j} d(\mathbf{x}, \mathbf{y}).$$

г) Відстань між центрами мас множин

$$d(C_i, C_j) = d(\mathbf{z}_i, \mathbf{z}_j).$$

д) Статистична відстань

$$d(C_i, C_j) = \frac{|C_i| \cdot |C_j|}{|C_i| + |C_j|} \|\mathbf{z}_i - \mathbf{z}_j\|^2,$$

де  $\mathbf{z}_i, \mathbf{z}_j$  — центри мас відповідних множин, які обчислюються за

$$\text{формулою } \mathbf{z}(C) = \frac{1}{|C|} \sum_{\mathbf{x} \in C} \mathbf{x}.$$

Відстань  $d(\mathbf{x}, C)$  між та об'єктом та множиною покладають рівною  $d(\{\mathbf{x}\}, C)$ .

Агломеративні алгоритми кластеризації із функцією г) запропонував **Сокал**, алгоритм із функцією д) — **Уорд**.

### **Алгоритм Боннера**

Випадковим чином вибирається одноелементний кластер, який об'єднується з тими одноелементними кластерами, відстань до яких не перевищує заданий поріг  $T$ . Далі випадково вибирається наступний одноелементний кластер і так само проходить злиття. Процес продовжується до тих пір, поки ще є одноелементні кластери.

### **Алгоритм поділу Макнаутона-Сміта**

У алгоритмі багато разів застосовується наступна процедура поділу множини  $C$  на два кластера  $C_1$  та  $C_2$ :

На початку усі об'єкти множини заносяться у кластер  $C_1$ , а кластер  $C_2$  — порожній.

На кожному кроці знаходиться об'єкт  $\mathbf{x} \in C_1$ , для якого величина  $d_x = d(\mathbf{x}, C_1 \setminus \{\mathbf{x}\}) - d(C_2, \mathbf{x})$  приймає найбільше значення, де  $d(\mathbf{x}, C_1 \setminus \{\mathbf{x}\}), d(\mathbf{x}, C_2)$  — відстані між об'єктом  $\mathbf{x}$  та відповідними множинами.

Якщо  $d_x > 0$ , то об'єкт  $\mathbf{x}$  виключається із кластера  $C_1$  і заноситься у кластер  $C_2$  і відбувається перехід на наступний крок процедури поділу, інакше робота процедури завершується.

У алгоритмі Макнаутона-Сміта спочатку усі об'єкти заносяться у один кластер  $C$ . Потім цей кластер за допомогою вищенаведеної процедури розбивається на два кластери  $C_1$  та  $C_2$ . Далі процес розбиття застосовується до одного з кластерів  $C_1$  та  $C_2$ . У результаті отримують три кластери і т. д. Рекомендується на кожному кроці вибирати для розбиття кластер найбільшого діаметру.

### 3.2. Неієрархічні алгоритми кластеризації

Для неієрархічних методів притаманним є використання поняття центра кластера. Об'єкти відносяться до того кластера, відстань до центра якого є найменшою.

#### 1. Пороговий алгоритм кластеризації

Нехай  $S = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_r\}$  — множина об'єктів, яку треба розбити на кластери і нехай також задано деякий поріг  $T$  і визначено вектор  $\mathbf{z}_1$  — центр першого кластера (у найпростішій модифікації  $\mathbf{z}_1 = \mathbf{x}_1$ ).

Припустимо, що об'єкти  $\{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_i\}$ , ( $i < r$ ) вже віднесені до одного із  $m$  кластерів  $C_1, \dots, C_m$  із центрами у  $\mathbf{z}_1, \dots, \mathbf{z}_m$ , ( $m \leq i$ ). Для вектора  $\mathbf{x}_{i+1}$  потрібно обчислити величини  $d(\mathbf{x}_{i+1}, \mathbf{z}_1), \dots, d(\mathbf{x}_{i+1}, \mathbf{z}_m)$  — відстані від  $\mathbf{x}_{i+1}$  до центрів кластерів  $C_1, \dots, C_m$ . Якщо усі ці відстані більші за поріг  $T$ , то створюється новий кластер  $C_{m+1}$  із центром  $\mathbf{z}_{m+1} = \mathbf{x}_{i+1}$ . У протилежному випадку об'єкт  $\mathbf{x}_{i+1}$  включається у кластер із найближчим до нього центром.

Результат роботи порогового алгоритму залежить від вибору першого центру та порогу  $T$ . Ця залежність проілюстрована на рис. 31.

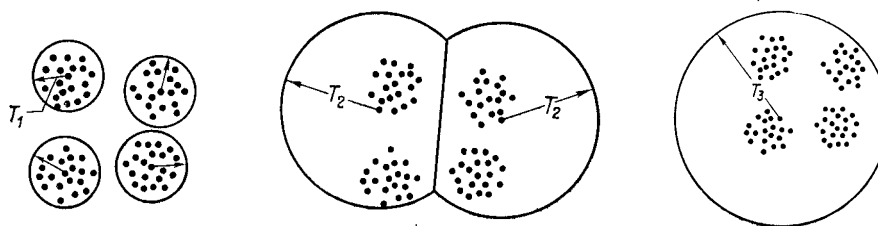


Рис. 31.

Існує велика кількість модифікацій порогового алгоритму. Наприклад, можна задавати бажану кількість кластерів та корегувати величину порогу  $T$ . Інша модифікація полягає у тому, що звичайний пороговий алгоритм застосовують  $r$  разів, змінюючи центр першого кластера (на  $i$ -ій ітерації покладають  $\mathbf{z}_1 = \mathbf{x}_i$ , ( $i = 1, \dots, r$ )). Для порівняння якості кластеризації (для фіксованого порогу  $T$ ) обчислюють величини

$$F_i = \sum_{j=1}^{m_i} \sum_{\mathbf{x} \in C_j^{(i)}} d(\mathbf{x}, \mathbf{z}_j), \quad i=1, \dots, r,$$

де  $C_j^{(i)}$  —  $j$ -ий кластер,  $m_i$  — кількість кластерів, які було отримано у результаті  $i$ -ої кластеризації. Очевидно, що величини  $F_i$  зменшуються із зменшенням порогу  $T$ .

Вибирають той варіант кластеризації  $\{C_1^{(i)}, \dots, C_{m_i}^{(i)}\}$ , для якого величина  $F_i$  є найменшою.

## 2. Алгоритм максимінної віддалі

Вхід. Множина об'єктів  $S = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_r\}$ ,  $\mathbf{x}_i \in \mathbb{R}^n$  ( $i = \overline{1, r}$ ,  $r > 2$ ).

Крок 1.  $\mathbf{z}_1 \leftarrow \mathbf{x}_1$ ,  $S' \leftarrow S \setminus \{\mathbf{x}_1\}$ .

Крок 2. У множині  $S'$  знаходиться об'єкт  $\mathbf{x}_2$ , який знаходиться на найбільшій відстані  $d_1$  від центра першого кластера  $\mathbf{z}_1$ .

$\mathbf{z}_2 \leftarrow \mathbf{x}_2$ ,  $S' \leftarrow S' \setminus \{\mathbf{x}_2\}$ ,  $m \leftarrow 2$ .

Крок 3. Якщо множина  $S'$  не порожня, то

1) Для усіх елементів  $\mathbf{x} \in S'$  обчислити відстані  $d(\mathbf{x}, \mathbf{z}_i)$ ,  $i = \overline{1, m}$  до центрів кластерів  $\{\mathbf{z}_1, \mathbf{z}_2, \dots, \mathbf{z}_m\}$

2) Знайти  $d(\mathbf{x}) = \min_{1 \leq i \leq m} \{d(\mathbf{x}, \mathbf{z}_i)\}$ .

3) Знайти  $d' = \max_{\mathbf{x}} d(\mathbf{x}) = d(\mathbf{x}')$  — максимум із знайдених мінімальних відстаней.

4) Вибрати коефіцієнт  $k_i$  ( $0 < k_i < 1$ ). Якщо

$$d' \leq k_i \cdot (d_1 + \dots + d_{m-1}) / (m-1),$$

то перейти на крок 4.

інакше

$$d_m \leftarrow d', \quad m \leftarrow m+1, \quad \mathbf{z}_m \leftarrow \mathbf{x}', \quad S' \leftarrow S' \setminus \{\mathbf{x}'\}$$

(кількість кластерів збільшується на одиницю, відповідний об'єкт  $\mathbf{x}'$  вилучається із навчальної вибірки та стає центром нового кластера).

5) перейти на початок кроку 3.

інакше перейти на крок 4.

Крок 4. На основі правила найближчого сусіда провести розбиття навчальної вибірки  $S$  на кластери із використанням заданої множини центрів  $\{\mathbf{z}_1, \mathbf{z}_2, \dots, \mathbf{z}_m\}$ .

Робота алгоритму проілюстрована на рис. 32-33.

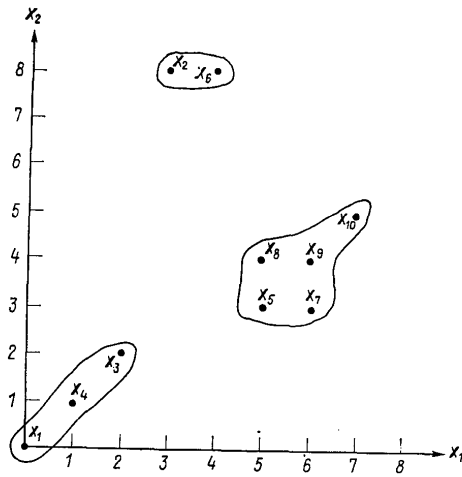


Рис. 32.

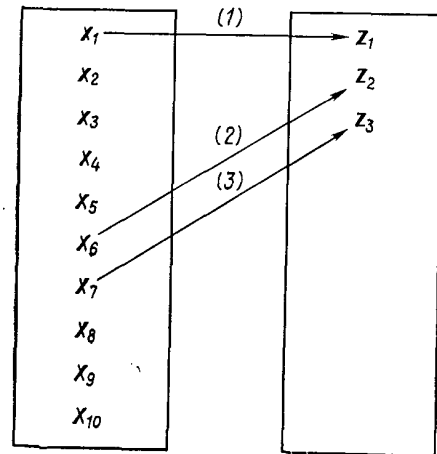


Рис. 33.

### 3. Алгоритм $k$ -means (групових середніх)

У цьому алгоритмі кількість кластерів скінченної множини  $X \subset \mathbb{R}^n$  задається наперед та мінімізується показник якості кластеризації, який вважається рівним сумі квадратів відстаней усіх елементів кластера до його центра.

Крок 1.  $r \leftarrow 0$  та вибрати вектори  $\mathbf{z}_1^{(1)}, \dots, \mathbf{z}_k^{(1)}$  — початкові центри кластерів.

Крок 2.  $r \leftarrow r + 1$  та розподілити усі об'єкти  $\mathbf{x}$  множини  $X$  по кластерам за наступним правилом:

$$\mathbf{x} \in C_j^{(r)} \Leftrightarrow \|\mathbf{x} - \mathbf{z}_j^{(r)}\| < \|\mathbf{x} - \mathbf{z}_i^{(r)}\|, i = 1, \dots, k, i \neq j.$$

Крок 3. Обчислити нові центри кластерів  $\mathbf{z}_1^{(r+1)}, \dots, \mathbf{z}_k^{(r+1)}$  за формулою (3).

$$\mathbf{z}_i^{(r+1)} = \frac{1}{|C_i^{(r)}|} \sum_{\mathbf{x} \in C_i^{(r)}} \mathbf{x}, i = 1, \dots, k, \quad (3)$$

де  $|C_i^{(r)}|$  — кількість елементів кластера  $C_i^{(r)}$ .

Крок 4. Якщо для усіх центрів виконується умова  $\mathbf{z}_i^{(r+1)} = \mathbf{z}_i^{(r)}, i = 1, \dots, k$ , то робота алгоритму завершена, інакше перейти на крок 2.

Вибір центрів кластерів на кроці 1 може проводитися довільним чином. Можна показати, що центри кластерів, координати яких обчислюються за формулою (3), є оптимальними згідно до показника якості. Кавер та Шлезінгер довели, що алгоритм групових середніх збігається за скінченну кількість кроків.

**Приклад.** Проілюструємо алгоритм  $k$ -means на прикладі розбиття 20-елементної множини  $X$ , елементи якої наведені на рис. 34, на два кластери.

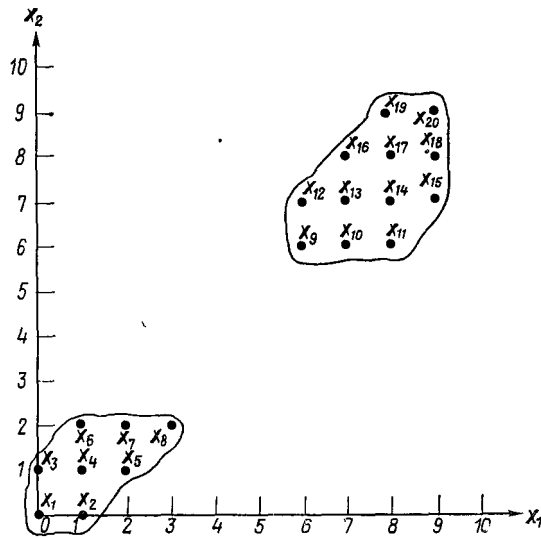


Рис. 34. Ілюстрація роботи алгоритму групових середніх

Крок 1. Покладаємо  $\mathbf{z}_1^{(1)} = \mathbf{x}_1 = (0,0)$ ,  $\mathbf{z}_2^{(1)} = \mathbf{x}_2 = (1,0)$ .

Крок 2.  $C_1^{(1)} = \{\mathbf{x}_1, \mathbf{x}_3\}$ ,  $C_2^{(1)} = \{\mathbf{x}_2, \mathbf{x}_4, \mathbf{x}_5, \dots, \mathbf{x}_{20}\}$ .

Крок 3. Перерахунок центрів кластерів

$$\mathbf{z}_1^{(2)} = \frac{1}{2}(\mathbf{x}_1 + \mathbf{x}_3) = (0; 0,5), \quad \mathbf{z}_2^{(2)} = \frac{1}{18}(\mathbf{x}_2 + \mathbf{x}_4 + \dots + \mathbf{x}_{20}) = (5,67; 5,33).$$

Крок 4. Так як  $\mathbf{z}_i^{(2)} \neq \mathbf{z}_i^{(1)}$ ,  $i = 1, 2$ , то виконується повернення на крок 2.

Крок 2. Використовуючи нові центри, отримуємо

$$C_1^{(2)} = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_8\}, \quad C_2^{(2)} = \{\mathbf{x}_9, \mathbf{x}_{10}, \dots, \mathbf{x}_{20}\}.$$

Крок 3. Перерахунок центрів кластерів

$$\mathbf{z}_1^{(3)} = \frac{1}{8}(\mathbf{x}_1 + \mathbf{x}_2 + \dots + \mathbf{x}_8) = (1,25; 1,13), \quad \mathbf{z}_2^{(3)} = \frac{1}{12}(\mathbf{x}_9 + \mathbf{x}_{10} + \dots + \mathbf{x}_{20}) = (7,67; 7,33).$$

Крок 4. Так як  $\mathbf{z}_i^{(3)} \neq \mathbf{z}_i^{(2)}$ ,  $i = 1, 2$ , то виконується повернення на крок 2.

Крок 2. Використовуючи нові центри, отримуємо

$$C_1^{(3)} = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_8\}, \quad C_2^{(3)} = \{\mathbf{x}_9, \mathbf{x}_{10}, \dots, \mathbf{x}_{20}\}.$$

Крок 3. Перерахунок центрів кластерів

$$\mathbf{z}_1^{(4)} = \frac{1}{8}(\mathbf{x}_1 + \mathbf{x}_2 + \dots + \mathbf{x}_8) = (1,25; 1,13), \quad \mathbf{z}_2^{(4)} = \frac{1}{12}(\mathbf{x}_9 + \mathbf{x}_{10} + \dots + \mathbf{x}_{20}) = (7,67; 7,33).$$

Крок 4. Так як  $\mathbf{z}_i^{(4)} = \mathbf{z}_i^{(3)}$ ,  $i = 1, 2$ , то робота алгоритму завершена.

Результатом алгоритму роботи буде розбиття  $\{C_1, C_2\}$  множини  $X$  на кластери  $C_1 = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_8\}$  та  $C_2 = \{\mathbf{x}_9, \mathbf{x}_{10}, \dots, \mathbf{x}_{20}\}$  із центрами  $\mathbf{z}_1 = (1,25; 1,13)$ ,  $\mathbf{z}_2 = (7,67; 7,33)$

Алгоритм  $k$ -середніх має багато модифікацій.

Одна з найпростіших модифікацій — алгоритм  $k$ -медіан (partitioning around medoids). На кроці 3 цього алгоритмі у якості центру  $i$ -го кластера використовують вектор  $\mathbf{z}_i^{(r+1)}$ , кожна координата якого є медіаною відповідних координат векторів кластеру  $C_i^{(r)}$ , ( $i=1, \dots, k$ ).

#### **4. Алгоритм ISODATA (алгоритм Болла-Холла)**

Алгоритм ІЗОДАТА є модифікацією алгоритму групових середніх. Спочатку випадковим чином із множини  $X$ , яка підлягає кластеризації, вибираються центри  $k$  кластерів. Інші об'єкти множини  $X$  відносяться до кластера із найближчим центром. Далі знаходяться центри кластерів так само, як у методі групових середніх. Два кластери  $C_i$  та  $C_j$  зливаються, якщо статистична відстань  $d(C_i, C_j)$  менша за задане порогове значення  $T_1$ . Якщо середньо групова дисперсія деякого кластера  $C_i$  по деякій координаті  $r$  ( $1 \leq r \leq n$ ) більша за порогове значення  $T_2$ , то відповідний кластер розбивається. Далі процедура застосовується до отриманих кластерів.



## Література

1. Duda R. O., Hart P. E., Stork D. G. Pattern classification, 2nd ed., Wiley & Sons, NY. — 654 с.
2. Форсайт Д., Понс Ж. — Компьютерное зрение. Современный подход. — М.: Вильямс, 2004. — 928 с.
3. Гонсалес Р., Вудс Р. Цифровая обработка изображений. — М.: Техносфера, 2005. — 1072 с.
4. Шапиро Л., Стокман Дж. — Компьютерное зрение. — М.: Вильямс, 2006. — 752 с.
5. Ту Дж., Гонсалес Р. Принципы распознавания образов. — М.: Мир, 1978. — 402 с.
6. Фисенко В.Т., Фисенко Т.Ю. — Компьютерная обработка и распознавание изображений. — СПб.: СПбГУ, 2008. — 192 с.
7. Васильев В.И. — Распознающие системы: Справочник. — К.: Наукова думка, 1983. — 432 с.
8. Анисимов Б.В., Курганов В.Д., Злобин В.К. — Распознавание и цифровая обработка изображений. — М.: Высш. шк., 1983. — 295 с.
9. Аркадьев А.Г., Браверман Э.М. Обучение машины распознаванию образов. — М.: Наука, 1964. — 110 с.