Розділ 7.

КЛАСИЧНІ ТЕХНОЛОГІЇ ІНТЕЛЕКТУАЛЬНОГО АНАЛІЗУ ДАНИХ

7.1. Класичні технології класифікації в Data Mining

Класифікація є найбільш простою і одночасно найбільш часто вирішуваною задачею Data Mining [1, 9, 29, 58]. Зважаючи на поширеність задач класифікації необхідне чітке розуміння суті цього поняття. Наведемо декілька означень.

Означення 1. Класифікація - системний розподіл предметів, явищ, процесів, які вивчаються, за родами, видами, типами, за якими-небудь істотними ознаками для зручності їх дослідження; групування вихідних понять і розташування їх у певному порядку, що відображає міру цієї схожості.

Означення 2. Класифікація - впорядкована за деяким принципом множина об'єктів, які мають схожі класифікаційні ознаки (одну або декілька властивостей), вибраних для визначення схожості або відмінності між цими об'єктами.

Класифікація вимагає дотримання наступних правил:

- у кожному акті ділення необхідно застосовувати лише одну підставу;
- ділення має бути відповідним, тобто загальний обсяг видових понять повинен дорівнювати обсягу поділеного родового поняття;
- члени ділення повинні взаємно виключати одне одного, їх обсяги не повинні перехрещуватися;
- ділення має бути послідовним.

В класичній теорії розглядають:

- допоміжну (штучну) класифікацію, яка виконується за зовнішньою ознакою і служить для придання множині предметів (процесів, явищ) потрібного порядку;

- природну класифікацію, яка виконується за суттєвими ознаками, що характеризують внутрішню спільність предметів і явищ. Вона є результатом і важливим засобом наукового дослідження, оскільки передбачає і закріплює результати вивчення закономірностей об'єктів, що класифікуються.

Залежно від вибраних ознак, їх поєднання і процедури ділення понять класифікація може бути:

- простою - ділення родового поняття лише за ознакою і лише один раз до розкриття всіх видів. Прикладом такої класифікації є дихотомія, при якій членами ділення бувають лише два поняття, кожне з яких є таким, що суперечить іншому (тобто дотримується принцип: «A і не A»);
- складною - застосовується для ділення одного поняття за різними підставами і синтезу таких простих ділень в єдине ціле. Прикладом такої класифікації є періодична система хімічних елементів.

Ми під класифікацією будемо розуміти віднесення об'єктів (спостережень, подій) до одного із заздалегідь відомих класів. Класифікація - це закономірність, що дозволяє робити висновок відносно визначення характеристик конкретної групи. Таким чином, для проведення класифікації мають бути присутніми ознаки, що характеризують групу, до якої належить та або інша подія або об'єкт (зазвичай при цьому на підставі аналізу вже класифікованих подій формулюються деякі правила).

Класифікація відноситься до стратегії навчання з вчителем (supervised learning), яка також іменують контрольованим або керованим навчанням. Задачею класифікації часто називають передбачення категоріальної залежної змінної (тобто залежної змінної, що є категорією) на основі вибірки безперервних і категоріальних змінних. Наприклад, можна передбачити, хто з клієнтів фірми є потенційним покупцем певного товару, а хто - ні, хто скористається послугою фірми, а хто - ні, і так далі. Цей тип задач відноситься до задач *бінарної класифікації*, в них залежна змінна може набувати лише два значення (наприклад, «так чи ні», «0 або 1»). Інший варіант класифікації (*багатокласова класифікація*) виникає, якщо залежна змінна може приймати

значення з деякої множини зумовлених класів. Наприклад, коли необхідно передбачити, яку марку автомобіля захоче купити клієнт. У цих випадках розглядається множина класів для залежної змінної.

Класифікація може бути *одновимірною* (за однією ознакою) і *багатовимірною* (по двом і більше ознакам). Багатовимірна класифікація була розроблена біологами при вирішенні проблем дискримінації для класифікації організмів. Однією з перших робіт, присвячених цьому напряму, рахують роботу Р. Фішера, в якій організми розділялися на підвиди залежно від результатів вимірів їх фізичних параметрів. Біологія була і залишається найбільш жаданим і зручним середовищем, для розробки багатовимірних методів класифікації.

Розглянемо задачу класифікації на простому прикладі [15]. Допустимо, є база даних про клієнтів морського курорту з інформацією про вік і дохід за місяць. Є рекламний матеріал двох видів: дорожчий і комфортніший відпочинок і дешевший, традиційний відпочинок. Відповідно, визначено два класи клієнтів: клас 1 і клас 2. База даних приведена в таблиці 7.1.

Табл. 7.1.

База даних клієнтів туристичного агентства

Код клієнта	Вік	Дохід	Клас
1	25	25000	1
2	26	80000	1
3	30	70000	1
4	32	120000	1
5	22	15000	2
6	27	29000	1
7	21	20000	2
8	20	23000	2
9	22	75000	1
10	24	24000	2

Необхідно визначити, до якого класу належить новий клієнт і який з двох видів рекламних матеріалів йому варто посилати.

Для наочності представимо базу даних в двомірному вимірі (вік – вісь X і дохід – вісь Y), у вигляді множини об'єктів, що належать класам 1 і 2 (рис. 7.1). Рішення нашої задачі полягатиме в тому, аби визначити, до якого класу відноситься новий клієнт (біла мітка).

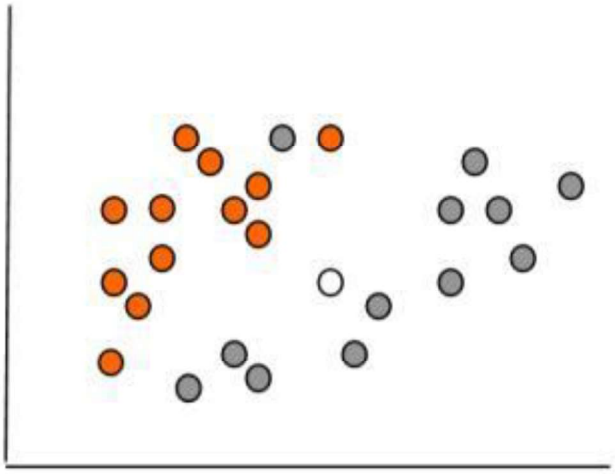


Рис. 7.1. Множина об'єктів бази даних в двомірному вимірі.

Мета процесу класифікації полягає в тому, аби побудувати модель, яка використовує прогнозуючі атрибути як вхідні параметри і отримує значення залежного атрибуту. Процес класифікації полягає в розбитті множини об'єктів на класи по певному критерію. Класифікатором називається деяка сутність, що визначає, якому із зумовлених класів належить об'єкт по вектору ознак.

Для проведення класифікації за допомогою математичних методів необхідно мати формальний опис об'єкту, яким можна оперувати, використовуючи математичний апарат класифікації. Таким описом в нашому випадку виступає база даних. Кожен об'єкт (запис бази даних) несе інформацію про деяку властивість об'єкту. Набір даних розбивають на дві множини: *навчальну* і *тестову*. Навчальна множина (training set) - множина, яка включає дані, що використовуються для навчання (конструювання) моделі. Така множина містить вхідні і вихідні (цільові) значення прикладів. Вихідні значення призначені для навчання моделі. Тестова (test set) множина також

містить вхідні і вихідні значення прикладів. Тут вихідні значення використовуються для перевірки працездатності моделі.

Процес класифікації складається з двох етапів: конструювання моделі і її використання.

Конструювання моделі передбачає опис множини зумовлених класів. Зокрема, на цьому етапі виконуються наступні дії:

- кожен приклад набору даних відноситься до одного зумовленого класу;
- використовується навчальна множина, на якій відбувається конструювання моделі;
- отримана модель представляється класифікаційними правилами, деревом рішень або математичною моделлю.

Використання моделі здійснює класифікацію нових або невідомих значень. Зокрема, на цьому етапі виконуються такі дії:

1. Оцінюється правильність та точність моделі, тобто:

- відомі значення з тестового прикладу порівнюються з результатами використання отриманої моделі;
- рівень точності визначається як відсоток правильно класифікованих прикладів в тестовій множині;
- тестова множина не повинна залежати від навчальної множини.

2. Якщо точність моделі допустима, можливе використання моделі для класифікації нових прикладів, клас яких невідомий.

Процес класифікації, а саме, конструювання моделі, представлений на рис. 7.2.

Оцінка точності класифікації може проводитися за допомогою *крос-перевірки*. Крос-перевірка (Cross-validation) - це процедура оцінки точності класифікації на даних з тестової множини, яку також називають крос-перевірочною множиною. Точність класифікації тестової множини порівнюється з точністю класифікації навчальної множини. Якщо процедура дає приблизно такі ж результати по точності, то вважається, що дана модель пройшла крос-перевірку.

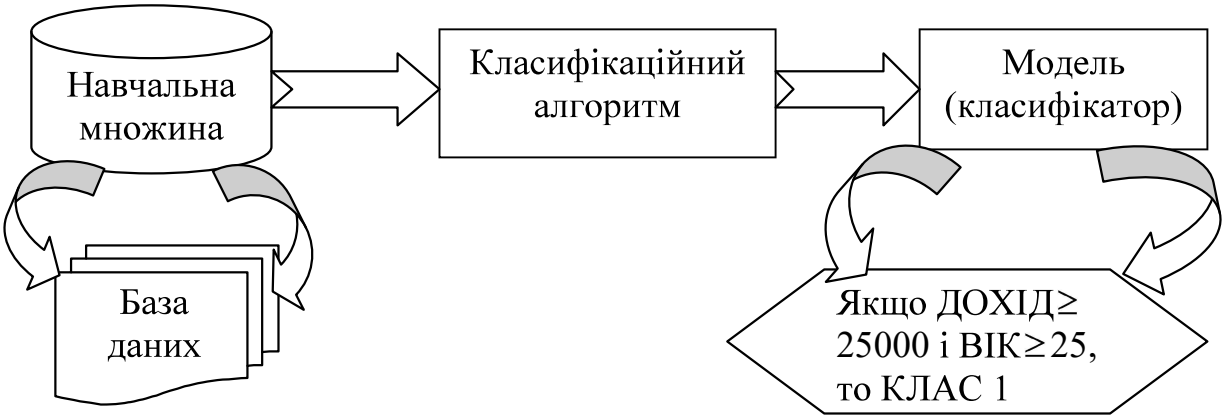


Рис. 7.2. Конструювання моделі класифікації.

При виборі методів класифікації слід проводити їх оцінювання, виходячи з таких характеристик: швидкість, робастність, інтерпретуємість, надійність. *Швидкість* характеризує час, який потрібний на створення моделі і її використання. *Робастність*, тобто стійкість до яких-небудь порушень деяких передумов, означає можливість роботи із зашумленими даними і пропущеними значеннями в даних. *Інтерпретуємість* забезпечує можливість розуміння моделі аналітиком. *Надійність* методів класифікації передбачає можливість роботи цих методів за наявності в наборі даних шумів і викидів.

Розглянемо деякі прикладні задачі, які ефективно вирішуються методами класифікації.

Задачі медичної діагностики. В ролі об'єктів виступають пацієнти. Ознаки характеризують результати обстежень, симптоми захворювання і методи лікування, що застосовувалися. Приклади бінарних ознак: стать, наявність головного болю, слабкості. Порядкова ознака - тяжкість стану (задовільний, середньої тяжкості, важкий, вкрай важкий). Кількісні ознаки - вік, пульс, артеріальний тиск, вміст гемоглобіну в крові, доза препарату. Ознаковий опис пацієнта є, по суті справи, формалізованою історією хвороби. Накопивши достатню кількість прецедентів в електронному вигляді, можна вирішувати різні задачі:

- класифікувати вигляд захворювання (диференціальна діагностика);

- визначати найбільш доцільний спосіб лікування;
- передбачати тривалість і результат захворювання;
- оцінювати ризик ускладнень;
- знаходити синдроми - найбільш характерні для даного захворювання.

Цінність такого роду систем в тому, що вони здатні миттєво аналізувати і узагальнювати величезну кількість прецедентів - можливість, недоступна фахівцеві-лікареві.

Передбачення родовищ корисних копалин. Ознаками є дані геологічної розвідки. Наявність або відсутність тих або інших порід на території району кодується бінарними ознаками. Фізико - хімічні властивості цих порід можуть описуватися як кількісними, так і якісними ознаками. Навчальна вибірка складається з прецедентів двох класів: районів відомих родовищ і схожих районів, в яких копалина, що цікавить, виявлена не була. При пошуку рідких корисних копалин кількість об'єктів може виявитися набагато менше, ніж кількість ознак. У цій ситуації погано працюють класичні статистичні методи. Задача вирішується шляхом пошуку закономірностей в наявному масиві даних. В процесі рішення виділяються короткі набори ознак, що володіють найбільшою інформативністю - здатністю щонайкраще розділяти класи. По аналогії з медичними задачами, можна сказати, що відшукуються «синдроми» родовищ. Це важливий результат дослідження, що представляє значний інтерес для геофізиків і геологів.

Оцінювання кредитоспроможності позичальників. Ця задача вирішується банками при видачі кредитів. Потреба в автоматизації процедури видачі кредитів вперше виникла в період буму кредитних карт 60-70-х років в США і інших європейських країнах. Об'єктами в даному випадку є фізичні або юридичні особи, що претендують на здобуття кредиту. В разі фізичних осіб ознаковий опис складається з анкети, яку заповнює сам позичальник, і, можливо, додаткової інформації, яку банк збирає про нього з власних джерел. Приклади бінарних ознак: стать, наявність телефону. Номінальні ознаки - місце мешкання, професія, працедавець. Порядкові ознаки - освіта, посада. Кількісні

ознаки - сума кредиту, вік, стаж роботи, дохід сім'ї, розмір заборгованостей в інших банках. Навчальна вибірка складається з позичальників з відомою кредитною історією. У простому випадку ухвалення рішень зводиться до класифікації позичальників на два класи: «хороших» і «поганих». Кредити видаються лише позичальникам першого класу. У складнішому випадку оцінюється сумарне число балів (score) позичальника, набраних по сукупності інформативних ознак. Чим вище оцінка, тим більше надійним вважається позичальник. На стадії навчання виконується синтез і відбір інформативних ознак і визначається, скільки балів призначати за кожну ознаку, аби ризик рішень, що приймаються, був мінімальний. Наступна задача - вирішити, на яких умовах видавати кредит: визначити процентну ставку, термін погашення, і інші параметри кредитного договору. Ця задача також може бути вирішення методами навчання по прецедентах.

Серед важливих задач, які також вирішуються методами класифікації, слід відзначити задачу передбачення відтоку клієнтів, оптичне розпізнавання символів, розпізнавання мови, виявлення спаму, класифікація документів та інше.

Існує велика різноманітність методів класифікації. Найбільш поширеними з них є:

- класифікація методом опорних векторів;
- байєсівська класифікація;
- статистичні методи, зокрема, лінійна регресія;
- класифікація за допомогою методу найближчого сусіда;
- класифікація CBR-методом;
- класифікація за допомогою штучних нейронних мереж;
- класифікація за допомогою дерев рішень;
- класифікація за допомогою генетичних алгоритмів.

Зупинимося докладніше на головних з них, які отримали найбільше практичне застосування. Зазначимо, що три останні методи вже були розглянуті відповідно в розділах 3, 4, 5.

Метод опорних векторів. У 60–70-і роки колективом математиків під керівництвом В. Н. Вапника був розроблений метод узагальненого портрета, заснований на побудові оптимальної розділяючої гіперплощини. Вимога оптимальності полягало в тому, що навчальні об'єкти мають бути віддалені від розділяючої поверхні настільки далеко, наскільки це можливо. У 90-і роки метод здобув широку світову популярність і після деякої переробки і серії узагальнень став називатися машиною опорних векторів (Support Vector Machine - SVM). В даний час він вважається одним з кращих методів класифікації.

Метод опорних векторів відноситься до групи граничних методів. Він визначає класи за допомогою границь областей. За допомогою даного методу вирішуються задачі бінарної класифікації.

Метод SVM володіє декількома чудовими властивостями. По-перше, навчання SVM зводиться до задачі квадратичного програмування, яка має єдине рішення, яке обчислюється досить ефективно навіть на вибірках в сотні тисяч об'єктів. По-друге, рішення володіє властивістю розрідженості: положення оптимальної розділяючої гіперплощини залежить лише від невеликої долі навчальних об'єктів. Вони і називаються *опорними векторами*; останні об'єкти фактично не задіюються. Нарешті, за допомогою введення *функції ядра* метод узагальнюється на випадок нелінійних розділяючих поверхонь.

В основі методу лежить поняття площин рішень. Площина (plane) рішення розділяє об'єкти з різною класовою приналежністю. Розглянемо задачу класифікації для двох класів, що не перетинаються, в якій об'єкти описуються n - мірними векторами: $X = R^n$, $Y = \{-1,+1\}$. Побудуємо лінійну функцію класифікації вигляду

$$a(x) = sign\left(\sum_{j=1}^{n} w_j x^j - w_0 \right) = sign(< w, x > - w_0),$$

де $x = (x^1,...,x^n)$ - ознаковий опис об’єкту x, вектор $w = (w_1,...,w_n) \in R^n$ та скалярний поріг w_0 є параметрами алгоритму (рис. 7.3).

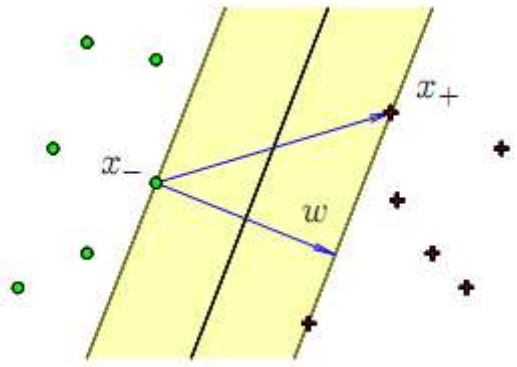


Рис. 7.3. Лінійно-подільна вибірка.

Об'єкти x_- і x_+ знаходяться на границі розділяючої смуги. Вектор нормалі w до розділяючої гіперплощини визначає ширину смуги. Рівняння $< w,x >= w_0$ описує гіперплощину, що розділяє класи в просторі R^n. Новий об'єкт, що потрапляє направо, класифікується як об'єкт класу x_+ або - як об'єкт класу x_-, якщо він розташувався ліворуч від розділяючої прямої.

Умова $-1 < wx^j - w_0 < 1$ задає смугу, що розділяє класи. Жодна з точок навчальної вибірки не може лежати усередині цієї смуги. Границями смуги є дві паралельні гіперплощини з направляючим вектором w. Точки, найближчі до розділяючої гіперплощини, лежать точно на границях смуги (рис. 7.4).

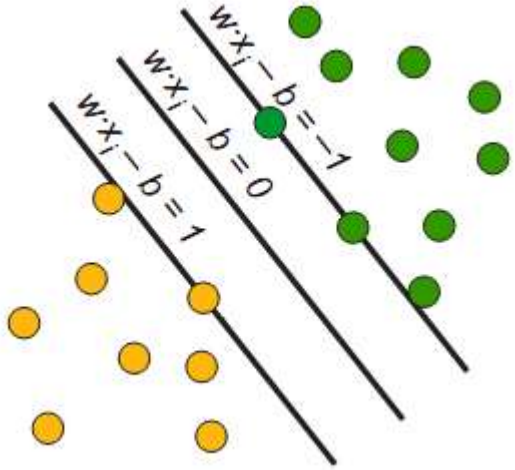


Рис. 7.4. Розділяюча смуга.

Аби розділяюча гіперплощина якнайдалі відстояла від точок вибірки, ширина смуги має бути максимальною і визначається як

$$\left\langle (x_+ - x_-), \frac{w}{\|w\|} \right\rangle = \frac{< wx_+ > - < wx_- >}{\|w\|} = \frac{(w_0 + 1) - (w_0 - 1)}{\|w\|} = \frac{2}{\|w\|}.$$

Ширина смуги максимальна, коли норма вектора w мінімальна. Таким чином, метод відшукує зразки, що знаходяться на границях між двома класами, тобто опорні вектора; що змальовані на рис. 7.5.

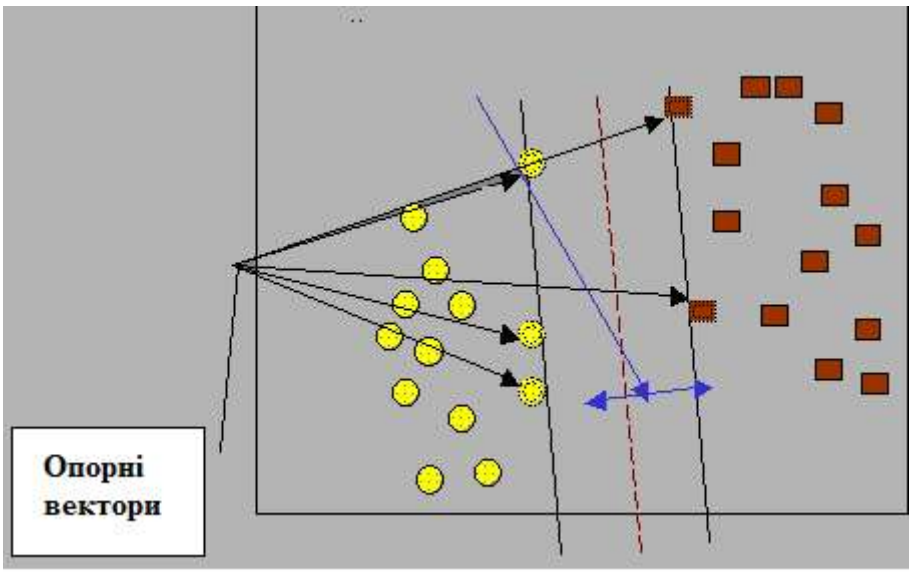


Рис. 7.5. Опорні вектори.

Опорними векторами називаються об'єкти множини, що лежать на границях областей. Класифікація вважається хорошою, якщо область між границями порожня.

Сформулюємо умови задачі пошуку оптимальної розділяючої смуги для випадку лінійної роздільності. Нехай існують обмеження $y_i(wx_i - b) \geq 1$. При цих обмеженнях y_i і x_i є константами, оскільки це елементи навчальної множини, w і b є змінними. Потрібно знайти такі w і b, аби виконувалися всі лінійні обмеження, і при цьому якомога менше була норма вектора w (отже, ширше розділяюча смуга), тобто необхідно мінімізувати: $\|w\|^2 = w * w$. Така задача називається задачею квадратичної оптимізації - при лінійних обмеженнях знайти мінімум квадратичної функції.

Найкращою функцією класифікації є функція, для якої очікуваний ризик мінімальний. Поняття очікуваного ризику в даному випадку означає очікуваний рівень помилки класифікації. Безпосередньо оцінити очікуваний рівень помилки побудованої моделі неможливо, це можна зробити за допомогою поняття емпіричного ризику. Проте слід враховувати, що мінімізація останнього не завжди приводить до мінімізації очікуваного ризику. Цю обставину слід пам'ятати при роботі з відносно невеликими наборами навчальних даних. Емпіричний ризик - рівень помилки класифікації на навчальному наборі. Таким чином, в результаті рішення задачі методом опорних векторів для даних, що лінійно розділяються, ми отримуємо функцію класифікації, яка мінімізує верхню оцінку очікуваного ризику.

Однією з проблем, пов'язаних з вирішенням задач класифікації даним методом, є та обставина, що не завжди можна легко знайти лінійну границю між двома класами. У таких випадках один з варіантів - збільшення розмірності, тобто перенесення даних з площини в тривимірний простір, де можливо побудувати таку площину, яка ідеально розділить множину зразків на два класи. Опорними векторами в цьому випадку служитимуть об'єкти з обох класів, які є екстремальними. Таким чином, за допомогою додавання так званого оператора ядра і додаткової розмірності, знаходяться границі між класами у вигляді гіперплощин. Проте слід пам'ятати: складність побудови SVM-моделей полягає в тому, що чим вище розмірність простору, тим складніше з ним працювати. Один з варіантів роботи з даними високої розмірності - це попереднє вживання якого-небудь методу пониження розмірності даних для виявлення найбільш істотних компонент, а потім використання методу опорних векторів.

Як і будь-який інший метод, метод SVM має свої сильні і слабкі сторони, які слід враховувати при виборі даного методу. Недолік методу полягає в тому, що для класифікації використовується не вся множина зразків, а лише їх невелика частина, яка знаходиться на границях. Достоїнство методу полягає в тому, що для класифікації методом опорних векторів, на відміну від

більшості інших методів, досить невеликого набору даних. При правильній роботі моделі, побудованої на тестовій множині, цілком можливо вживання даного методу на реальних даних.

Байєсівська класифікація. Байєсівські процедури класифікації розроблені на основі теореми Байєса і спеціально призначені для роботи з вхідними даними високої розмірності. Не дивлячись на простоту таких процедур, результати їх роботи по своїх характеристиках можуть перевершити результати роботи досить складних алгоритмів класифікації. Спочатку байєсівська класифікація використовувалася для формалізації знань експертів в експертних системах, зараз байєсівська класифікація широко застосовується як один з методів Data Mining [45, 61].

З метою продемонструвати основні принципи роботи байєсівських процедур класифікації, розглянемо приклад (рис. 7.6). Як видно, об'єкти можуть бути розділені на два класи: GREEN або RED. Наша мета - класифікувати нові спостереження в міру їх поступлення, тобто потрібно вирішити до якого класу вони належать, використовуючи інформацію про приналежність класам вже наявних в нашому розпорядженні об'єктів.

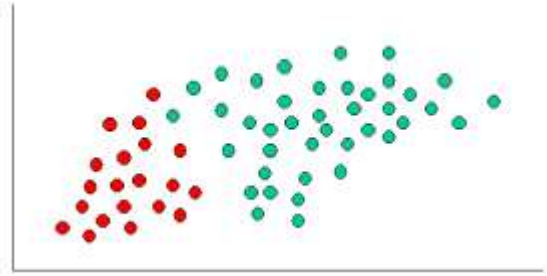


Рис. 7.6. Класи об’єктів.

Оскільки об'єктів типа GREEN в два рази більше об'єктів типа RED, розумно передбачити, що шанси приналежності нового спостереження класу GREEN, в два рази більше шансів належати класу RED. В термінах байєсівського аналізу це припущення називається *апріорною ймовірністю*. Апріорна ймовірність визначається накопиченим досвідом (у нашому випадку процентним

співвідношенням об'єктів типа GREEN і RED). Ця величина зазвичай використовується для передбачення результатів до їх реального настання. Таким чином, ми можемо записати:

Prior probability for GREEN = Number of GREEN objects/Total number of objects

Prior probability for RED = Number of RED objects/Total number of objects

Оскільки загальне число об'єктів - 60, 40 з них належать класу GREEN і 20 - класу RED, то апріорна ймовірність приналежності класу буде:

Prior probability for GREEN = 40/60

Prior probability for RED = 20/60

Визначивши апріорну ймовірність, ми готові класифікувати новий об'єкт (білий круг) (рис. 7.7). Через хороше угрупування об'єктів, розумно передбачити, що чим більше об'єктів типа GREEN (або RED) потрапляє в окресність точки X, тим вірогідніше, що нове спостереження належатиме цьому класу.

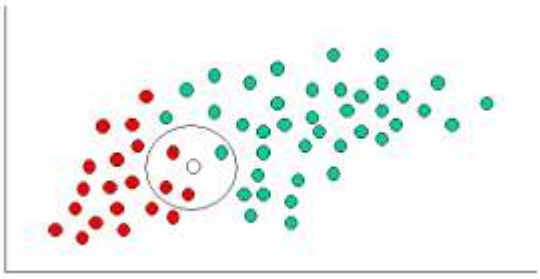


Рис. 7.7. Класифікація нового об’єкта.

Для обчислення міри правдоподібності, проведемо коло з центром в точці X, яка охопить апріорі вибране число точок безвідносно до їх класової приналежності. Потім підраховується число точок кожного типа. За цими даними обчислюємо міру правдоподібності

Likelihood of X given GREEN = Number of GREEN

in the vicinity of X / Total number of GREEN cases

Likelihood of X given RED = Number of RED

in the vicinity of X / Total number of RED cases

На наведеній вище ілюстрації видно, що міра правдоподібності приналежності X класу GREEN нижче за відповідне значення для класу RED, оскільки коло включає 1 об'єкт типа GREEN і 3 об'єкти типа RED. Отже

Probability of X given GREEN = 1/40

Probability of X given RED = 3/40

Хоча апріорна ймовірність вказує на можливу приналежність спостереження X класу GREEN (об'єктів типа GREEN в два рази більше об'єктів типа RED), величина міри правдоподібності приводить до протилежного висновку: X належить класу RED (у околиці точки X об'єктів типа RED більш ніж об'єктів типа GREEN). Кінцеве класифікуюче рішення в байєсівському аналізі приймається на основі двох джерел інформації: апріорній імовірності і мірі правдоподібності. Для визначення апостеріорної ймовірності застосовується правило Байєса (названо на честь Thomas Bayes).

*Posterior probability of X being GREEN = Prior probability of GREEN **

*Likelihood of X given GREEN = 4/6 * 1/40 = 1/60*

*Posterior probability of X being RED = Prior probability of RED **

*Likelihood of X given RED = 2/6 * 3/40 = 1/40*

В результаті ми класифікуємо X як об'єкт типа RED, оскільки апостеріорна ймовірність приналежності цьому класу має найбільшого значення.

«Наївний» (спрощений) алгоритм Байєса (naive-bayes approach) є одним з ефективних алгоритмів класифікації. Точність класифікації, здійснюваної «наївним» алгоритмом, порівнюється з точністю більшості інших алгоритмів. З точки зору швидкості навчання, стабільності на різних даних і простоти реалізації, «наївний» алгоритм Байєса перевершує практично всі відомі ефективні алгоритми класифікації.

Навчання алгоритму виконується шляхом визначення відносних частот значень всіх атрибутів вхідних даних при фіксованих значеннях атрибутів класу. Класифікація здійснюється шляхом застосування правила Байєса для

обчислення умовної ймовірності кожного класу для вектора вхідних атрибутів. Вхідний вектор приписується класу, умовна ймовірність якого при даному значенні вхідних атрибутів максимальна. «Наївність» алгоритму полягає в припущенні, що вхідні атрибути умовно (для кожного значення класу) незалежні один від одного, тобто $P(X_i = x_i, X_j = x_j \mid C = c_k) = P(X_i = x_i \mid C = c_k)P(X_j = x_j \mid C = c_k)$ для всіх атрибутів X_i, X_j і значень класу C. Це припущення є дуже сильним, і, у багатьох випадках неправомірним, що робить факт ефективності класифікації за допомогою «наївного» алгоритму Байєса досить несподіваним.

Результатом роботи методу є так звані «прозорі» моделі. Властивостями наївної класифікації є:

1. Використання всіх змінних і визначення всіх залежностей між ними.
2. Наявність двох припущень відносно змінних:
    - всі змінні є однаково важливими;
    - всі змінні є статистично незалежними, тобто значення однієї змінної нічого не говорить про значення іншої.

В зв'язку з цим закономірний пошук таких модифікацій в алгоритмі, які ослабили б передумову про умовну незалежність атрибутів. Модифікацією алгоритму, що вирішують ці проблеми, можуть служити так звані байєсівські мережі. *Байєсівською мережею* називається направлений граф без циклів, що дозволяє представляти спільний розподіл випадкових змінних. Кожен вузол графа представляє випадкову змінну, а дуги – прямі залежності між ними. Точніше, мережа описує наступний вислів: кожна змінна залежить лише від безпосередніх предків. Таким чином, граф описує обмеження на залежність змінних одну від одної, що зменшує кількість параметрів спільного розподілу. Параметри спільного розподілу кодуються в наборі таблиць для кожної змінної у формі умовних розподілів за умови на значення змінних-предків. Структура графа і умовні розподіли вузлів при значеннях їх предків однозначно описують спільний розподіл всіх змінних, що дозволяє вирішувати задачу класифікації як

визначення значення змінної класу, що максимізувало її умовну ймовірність при заданих значеннях вхідних змінних (рис. 7.8).

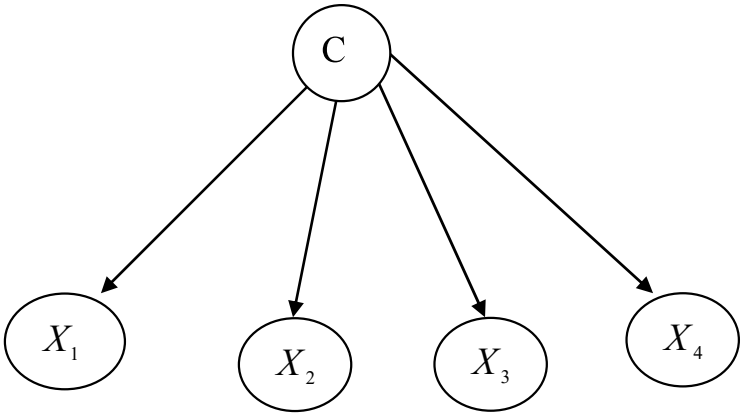


Рис. 7.8. Граф «наївної» байєсівської мережі.

Вочевидь, що «наївний» алгоритм Байєса є частинним випадком байєсівської мережі, де кожна вхідна змінна залежить лише від змінної класу, яка є єдиним коренем графа.

Навчання байєсівських мереж стало одним з актуальних напрямів обчислювальної математики і до цих пір є предметом активних досліджень. Проте, до цих пір визначення структури байєсівської мережі в загальному вигляді є складним завданням як з теоретичної, так і з обчислювальної точки зору. Підхід в загальному вигляді володіє наступними недоліками:

- обчислювальна складність;
- при спробі врахувати велику кількість залежностей між змінними, оцінки умовної ймовірності набувають великої дисперсії, оскільки їх спільна поява в даних є маловірогідною подією. Таким чином, оцінки параметрів можуть стати недостовірними, що у результаті може приводити до погіршення якості класифікації навіть в порівнянні з «наївним» алгоритмом Байєса;
- через велику кількість параметрів, модель виходить дуже орієнтованою на навчальні дані. Це приводить до дуже добрих результатів класифікації на навчальних даних і незадовільних результатів на тестових даних. Тобто модель

описує не загальні закономірності в структурі даних, а швидше набір окремих випадків в навчальній вибірці.

Відзначають також такі достоїнства байєсівських мереж як методу Data Mining:

- у моделі визначаються залежності між всіма змінними, це дозволяє легко обробляти ситуації, в яких значення деяких змінних невідомі;
- байєсівські мережі досить просто інтерпретуються і дозволяють на етапі прогностичного моделювання легко проводити аналіз сценарію «що, … якщо»;
- байєсівський метод дозволяє природним чином поєднувати закономірності, виведені з даних, і, наприклад, експертні знання, отримані в явному вигляді;
- використання байєсівських мереж дозволяє уникнути проблеми перенавчання (overfitting), тобто надлишкового ускладнення моделі, що є слабкою стороною багатьох методів (наприклад, дерев рішень і нейронних мереж).

Для вирішення позначених проблем використовуються обмеження на структуру графа і розглядаються такі розширення «наївної» моделі Байєса, де кожен вузол додатково може мати не більш за одного предка серед інших вхідних змінних. Модель з такими обмеженнями отримала назву TAN (Tree Augmented Naive Bayes). Оптимальна структура TAN-моделі (з точки зору функції правдоподібності) відповідає моделі з максимальною сумарною умовною по змінній класу взаємної інформації між вузлами і їх предками.

В іншому випадку застосовують байєсівські мережі, які моделюють послідовності змінних - динамічні байєсівські мережі та гібридні байєсівські мережі.

Нещодавно байєсівська класифікація була запропонована для персональної фільтрації спаму. Перший фільтр був розроблений Полем Грахемом (Paul Graham). Для роботи алгоритму потрібне виконання двох вимог. Перша вимога - необхідно, аби у об'єкта, що класифікується, була присутня достатня кількість ознак. Цьому ідеально задовольняють всі слова

листів користувача, за винятком зовсім коротких і таких, що дуже рідко зустрічаються. Друга вимога - постійне перенавчання і поповнення набору «спам - не спам». Такі умови дуже добре працюють в локальних поштових клієнтах, оскільки потік «не спаму» у кінцевого клієнта досить постійний, а якщо змінюється, то не швидко.

Проте для всіх клієнтів сервера точно визначити потік «не спаму» досить складно, оскільки один і той же лист, що є для одного клієнта спамом, для іншого спамом не є. Словник виходить дуже великим, не існує чіткого розділення на спам і «не спам», в результаті якість класифікації, в даному випадку рішення задачі фільтрації листів, значно знижується.

Метод «найближчого сусіда» або системи міркувань на основі аналогічних випадків.

У багатьох прикладних задачах вимірювати міру схожості об'єктів істотно простіше, ніж формувати ознакові описи. Наприклад, набагато легко порівняти дві фотографії і сказати, що вони належать одній людині, чим зрозуміти, на підставі яких ознак вони схожі. Такі ситуації часто виникають при розпізнаванні часових рядів або символьних послідовностей. Вони характеризуються тим, що «сирі» вихідні дані не годяться як ознакові описи, але в той же час, існують ефективні і змістовно обґрунтовані способи оцінити міру схожості будь-якої пари «сирих» описів.

Є ще одна характерна особливість цих задач. Якщо міра схожості введена досить вдало, то виявляється, що схожим об'єктам, як правило, відповідають схожі відповіді. У задачах класифікації це означає, що схожі об'єкти набагато частіше лежать в одному класі, чим в різних. Якщо задача в принципі піддається рішенню, то границя між класами не може «проходить всюди»; класи утворюють компактно локалізовані підмножини в просторі об'єктів. Це припущення прийнято називати *гіпотезою компактності*.

Для формалізації поняття «схожості» вводиться функція відстані або метрика $\rho(x,x')$ у просторі об'єктів X. Алгоритми, засновані на аналізі

схожості об'єктів, часто називають *метричними*, навіть в тих випадках, коли функція ρ не задовольняє всім аксіомам метрики.

Для довільного об'єкту $u \in X$ розташуємо елементи навчальної вибірки $x_1,...,x_l$ в порядку зростання відстаней до u:

$$\rho(u,x_u^{(1)}) \le \rho(u,x_u^{(2)}),...,\rho(u,x_u^{(l)}),$$

де через $i:u$ позначається номер i-го сусіда об'єкту u. Відповідно, відповідь на i-му сусідові об'єкту u є $y_u^{(i)} = y^*(x_u^{(i)})$. Фактично, будь-який об'єкт $u \in X$ породжує свою перенумерацію вибірки $X^l = \{(x_u^{(1)}, y_u^{(1)}),...,(x_u^{(l)}, y_u^{(l)})\}$.

Означення 3. Метричний алгоритм класифікації з навчальною вибіркою X^l відносить об'єкт u до того класу $y \in Y$, для якого сумарна вага найближчих навчальних об'єктів $\Gamma_y(u, X^l)$ максимальна:

$$a(u, X^l) = \arg\max \Gamma_y(u, X^l),\ \Gamma_y(u, X^l) = \sum_{i=1}^{l} [y_u^{(i)} = y] w(i,u),$$

де вагова функція $w(i,u)$ оцінює міру важливості i-го сусіда для класифікації об'єкту u. Функція $\Gamma_y(u, X^l)$ називається *оцінкою близькості об'єкту u до класу y*.

Навчальна вибірка X^l відіграє роль параметра алгоритму a. Налаштування зводиться до запам'ятовування вибірки, і, можливо, оптимізації якихось параметрів вагової функції, проте самі об'єкти не піддаються обробці і зберігаються «як є». З цієї причини метричні алгоритми відносяться до *методів міркування по прецедентах* (case-based reasoning, CBR). Тут дійсно можна говорити про «міркування», оскільки на питання «чому об'єкт u був віднесений до класу y?» алгоритм може дати сповна зрозуміле пояснення: «тому, що є схожі з ним прецеденти класу y», і пред'явити список цих прецедентів.

Прецедент - це опис ситуації у поєднанні з детальною вказівкою дій, що робляться в даній ситуації. Підхід, заснований на прецедентах, умовно можна поділити на наступні етапи:

- збір детальної інформації про поставлену задачу;

- зіставлення цієї інформації з деталями прецедентів, що зберігаються в базі, для виявлення аналогічних випадків;
- вибір прецеденту, найбільш близького до поточної проблеми, з бази прецедентів;
- адаптація вибраного рішення до поточної проблеми, якщо це необхідно;
- перевірка коректності кожного знов отриманого рішення;
- занесення детальної інформації про новий прецедент в базу прецедентів.

Таким чином, вивід, заснований на прецедентах, є такий метод аналізу даних, який робить висновки відносно даної ситуації за результатами пошуку аналогій, що зберігаються в базі прецедентів.

Даний метод за своєю суттю відноситься до категорії «навчання без вчителя», тобто є «самонавчальною» технологією, завдяки чому робочі характеристики кожної бази прецедентів з часом і накопиченням прикладів покращуються. Розробка баз прецедентів по конкретній предметній області відбувається на природній для людини мові, отже, може бути виконана найбільш досвідченими співробітниками компанії - експертами або аналітиками, що працюють в даній предметній області. Проте це не означає, що CBR-системи самостійно можуть приймати рішення. Останнє завжди залишається за людиною, даний метод лише пропонує можливі варіанти рішення і вказує на «найрозумніший» із його точки зору.

Вибираючи вагову функцію $w(i,u)$, можна отримувати різні метричні класифікатори:

$w(i,u) = [i = 1]$ - метод найближчого сусіда (1NN);

$w(i,u) = [i \le k]$ - метод k найближчих сусідів (k NN);

$w(i,u) = [i \le k]q^i$ - метод k зважених найближчих сусідів;

$w(i,u) = K\left(\frac{\rho(u,x_u^{(i)})}{h}\right)$ - метод парзеновського вікна ширини h;

$w(i,u) = K\left(\frac{\rho(u,x_u^{(i)})}{\rho(u,x_u^{(k+1)})}\right)$ - метод парзеновського вікна змінної ширини;

$w(i,u) = \gamma_u^{(i)} K\left(\frac{\rho(u, x_u^{(i)})}{h_u^{(i)}}\right)$ - метод потенційних функцій.

Алгоритм *найближчого сусіда* (nearest neighbor, NN) є найпростішим алгоритмом класифікації. Він відносить класифікуємий об'єкт $u \in X^l$ до того класу, якому належить найближчий навчальний об'єкт $a(u; X^l) = y_u^{(1)}$. Навчання NN зводиться до запам'ятовування вибірки X^l. Єдине достоїнство цього алгоритму - простота реалізації. Недоліків значно більше:

- нестійкість до похибок. Якщо серед навчальних об'єктів є *викид* - об'єкт, що знаходиться в оточенні об'єктів чужого класу, то не лише він сам буде класифікований невірно, але і ті об'єкти, що оточують його, і для яких він виявиться найближчим, також будуть класифіковані невірно;
- відсутність параметрів, які можна було б налаштовувати по вибірці. Алгоритм повністю залежить від того, наскільки вдало вибрана метрика ρ.
- в результаті - низька якість класифікації.

З метою усунення недоліків попереднього методу був розроблений алгоритм k *найближчих сусідів* (k nearest neighbors, kNN). Аби згладити шумовий вплив викидів, класифікуватимемо об'єкти шляхом *голосування* по k найближчих сусідів. Кожен з сусідів $x_u^{(i)}$, $i = 1,...,k$ голосує за віднесення об'єкту u до свого класу $y_u^{(i)}$. Алгоритм відносить об'єкт u до того класу, який набере більше число голосів

$$a(u; X^l, k) = \arg\max \sum_{i=1}^{k} [y_u^{(i)} = y].$$

При $k = 1$ цей алгоритм збігається з попереднім, отже, нестійкий до шуму. При $k = l$, навпаки, він надмірно стійкий і вироджується в константу. Таким чином, крайні значення k небажані. На практиці оптимальне значення параметра k визначають по критерію *ковзаючого контролю з виключенням об'єктів поодинці* (leave-one-out, LOO). Для кожного об'єкта $x_i \in X^l$ перевіряється, чи правильно він класифікується по своїх k найближчих сусідах.

$$LOO(k, X^l) = \sum_{i=1}^{l} [a(x_i; X^l \setminus \{x_i\}, k) \neq y_i] \to \min .$$

Якщо класифікуємий об'єкт x_i не виключати з навчальної вибірки, то найближчим сусідом x_i завжди буде сам x_i, і мінімальне (нульове) значення функціонала LOO(k) досягатиметься при $k = 1$.

Ще один спосіб задати ваги сусідам - визначити w_i як функцію від відстані $\rho(u, x_u^{(i)})$, а не від рангу сусіда i реалізується в методі парзеновського вікна. Введемо *функцію ядра* $K(z)$, що не зростає на $[0, \infty)$, і розглянемо алгоритм

$$a(u; X^l, h, k) = \arg\max \sum_{i=1}^{l} [y_u^{(i)} = y] K\left(\frac{\rho(u, x_u^{(i)})}{h}\right).$$

Параметр h називається *шириною вікна* і грає приблизно ту ж роль, що і число сусідів k. «Вікно» - це сферична околиця об'єкту u радіусу h, при попаданні в яку навчального об'єкту x_i об'єкт u «притягується» до класу y_i.

Розглянемо докладніше принципи роботи методів для вирішення задач класифікації і регресії (прогнозування).

Спочатку дослідимо рішення задачі класифікації нових об'єктів. Ця задача схематично змальована на рис. 7.9. Приклади (відомі екземпляри) відмічені знаком «+» або «-», що визначає приналежність до відповідного класу («+» або «-»), а новий об'єкт, який потрібно класифікувати, позначений кружечком. Нові об'єкти також називають точками запиту.

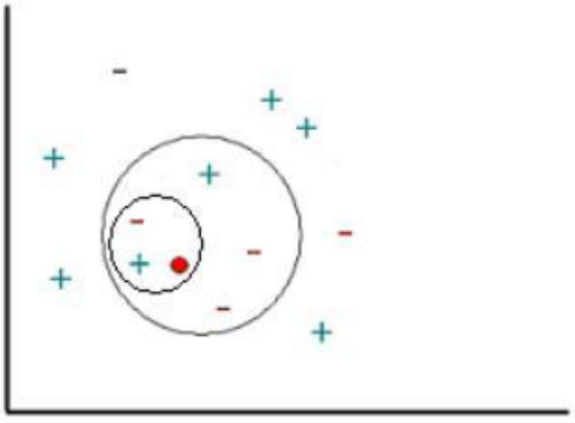


Рис. 7.9. Класифікація об'єктів множини при різному значенні параметра k.

Наша мета полягає в оцінці (класифікації) відгуку точок запиту з використанням спеціальний вибраного числа їх найближчих сусідів. Іншими словами, ми хочемо взнати, до якого класу слід віднести точку запиту: як знак «+» або як знак «-».

Спершу розглянемо результат роботи методу k найближчих сусідів з використанням одного найближчого сусіда. В цьому випадку відгук точки запиту буде класифікований як знак плюс, оскільки найближча сусідня точка має знак плюс. Тепер збільшимо число використовуваних найближчих сусідів до двох. Цього разу метод k найближчих сусідів не зможе класифікувати відгук точки запиту, оскільки друга найближча точка має знак мінус і обоє знаки рівноцінні (тобто перемога з однаковою кількістю голосів). Далі збільшимо число використовуваних найближчих сусідів до 5. Таким чином, буде визначена ціла околиця точки запиту (на графіці її границя відмічена колом). Оскільки в області міститься 2 точки із знаком «+» і 3 точки із знаком «-», алгоритм k найближчих сусідів привласнить знак «-» відгуку точки запиту.

Далі розглянемо принцип роботи методу k найближчих сусідів для вирішення задачі регресії. Регресійні задачі пов'язані з прогнозуванням значення залежної змінної по значеннях незалежних змінних набору даних. Розглянемо графік, показаний на рис. 7.10.

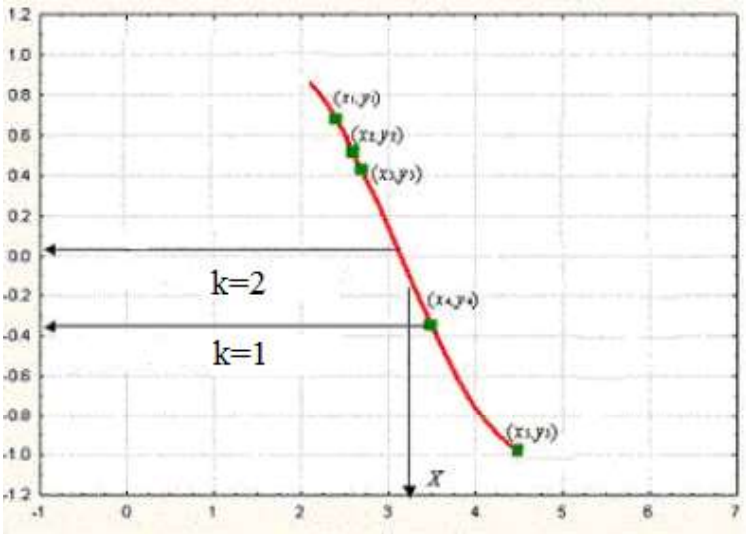


Рис. 7.10. Рішення задачі прогнозування при різних значеннях параметра k.

Змальований на ньому набір точок отриманий як зв'язок між незалежною змінною x і залежною змінною y (крива). Заданий набір об'єктів (тобто набір прикладів); ми використовуємо метод k найближчих сусідів для передбачення виходу точки запиту X по даному набору прикладів.

Спочатку розглянемо як приклад метод k найближчих сусідів з використанням одного найближчого сусіда, тобто при $k=1$. Шукатимемо набір прикладів і виділяємо з їх числа найближчий до точки запиту X. Для нашого випадку найближчий приклад - точка (x_4, y_4). Вихід x_4 (тобто y_4), таким чином, приймається як результат передбачення виходу X (тобто Y). Отже, для одного найближчого сусіда можемо записати: вихід $Y = y_4$.

Далі розглянемо ситуацію, коли $k=2$, тобто розглянемо двох найближчих сусідів. В цьому випадку ми виділяємо вже дві найближчі до X точки. На нашому графіку це точки y_3 і y_4 відповідно. Обчисливши середнє їх виходів, записуємо рішення для Y у вигляді $Y=(y_3 + y_4)/2$. Рішення задачі прогнозування здійснюється шляхом перенесення описаних вище дій на використання довільного числа найближчих сусідів таким чином, що вихід Y точки запиту X обчислюється як середньоарифметичне значення виходів k найближчих сусідів точки запиту.

Незалежні і залежні змінні набору даних можуть бути як неперервними, так і категоріальними. Для неперервних залежних змінних задача розглядається як задача прогнозування, для дискретних змінних - як задача класифікації. Критичним моментом у використанні методу k найближчих сусідів є вибір параметра k. Він один з найбільш важливих чинників, що визначають якість прогнозної або класифікаційної моделі. Якщо вибрано дуже маленьке значення параметра k, виникає ймовірність великого розкиду значень прогнозу. Якщо вибране значення дуже велике, це може привести до сильної зміщеності моделі. Таким чином, ми бачимо, що має бути вибране оптимальне значення параметра k. Тобто це значення має бути настільки великим, аби звести до мінімуму ймовірність невірної класифікації, і одночасно, досить малим, аби k сусідів

були розташовані досить близько до точки запиту. Таким чином, k розглядається як згладжуючий параметр, для якого має бути знайдений компроміс між силою розмаху (розкиду) моделі і її зміщеностью.

Один з варіантів оцінки параметра k - проведення крос-перевірки. Така процедура реалізована, наприклад, в пакеті STATISTICA. Крос-перевірка - відомий метод здобуття оцінок невідомих параметрів моделі. Основна ідея методу - розділення вибірки даних на v «складки», тобто, випадковим чином виділені ізольовані підвибірки. По фіксованому значенню k будується модель k найближчих сусідів для здобуття передбачень на v-му сегменті (останні сегменти при цьому використовуються як приклади) і оцінюється помилка класифікації. Для регресійних задач найчастіше як оцінка помилки виступає сума квадратів, а для класифікаційних задач зручніше розглядати точність (відсоток коректно класифікованих спостережень). Далі процес послідовно повторюється для всіх можливих варіантів вибору v. Після вичерпання v «складок», обчислені помилки усереднюються і використовуються як міра стійкості моделі. Вищеописані дії повторюються для різних k, і значення, відповідне найменшій помилці (або найбільшій класифікаційній точності), приймається як оптимальне. Слід враховувати, що крос-перевірка - обчислювально ємка процедура, і необхідно надати час для роботи алгоритму, особливо якщо об'єм вибірки досить великий.

Другий варіант вибору значення параметра k - самостійно задати його значення. Проте цей спосіб слід використовувати, якщо є обґрунтовані припущення відносно можливого значення параметра, наприклад, попередні дослідження схожих наборів даних.

Прикладом реального використання описаного вище методу є програмне забезпечення центру технічної підтримки компанії Dell, розроблене компанією Inference. Ця система допомагає співробітникам центру відповідати на більше число запитів, відразу пропонуючи відповіді на поширені питання і дозволяючи звертатися до бази під час розмови по телефону з користувачем. Співробітники центру технічної підтримки, завдяки реалізації цього методу,

можуть відповідати одночасно на значне число дзвінків. Програмне забезпечення CBR зараз розгорнуте в мережі Intranet компанії Dell.

Інструментів Data Mining, що реалізовують розглянуті методи не надто багато. Серед найбільш відомих: CBR Express і Case Point (Inference Corp.), Apriori (Answer Systems), DP Umbrella (VYCOR Corp.), KATE tools (Acknosoft, Франція), Pattern Recognition Workbench (Unica, США), а також деякі статистичні пакети, наприклад, Statistica.

Алгоритми обмеженого перебору. Алгоритми обмеженого перебору були запропоновані в середині 60-х років М. М. Бонгардом для пошуку логічних закономірностей в даних. З тих пір вони продемонстрували свою ефективність при вирішенні множини задач з самих різних областей. Ці алгоритми обчислюють частоти комбінацій простих логічних подій в підгрупах даних. Приклади простих логічних подій: $X = a$, $X > a$, $a \le X < b$ та ін., де X - деякий параметр, a і b - константи. Обмеженням служить довжина комбінації простих логічних подій. На підставі аналізу обчислених частот робиться висновок про корисність тієї або іншої комбінації для встановлення асоціації в даних, для класифікації, прогнозування та інше.

Найбільш яскравими сучасними представниками цього підходу є системи WizWhy та WizRule компанії WizSoft. Хоча автор системи Абрам Мейдан не розкриває специфіку алгоритму, покладеного в основу роботи програмних комплексів, за результатами ретельного тестування системи були зроблені виводи про наявність тут обмеженого перебору (вивчалися результати, залежності часу їх здобуття від числа аналізованих параметрів і ін.).

Як WizWhy так і WizRule роблять, по суті, одне і те ж: переглядають задану базу даних і, зібравши статистику, відшукують правила і закономірності, яким підкоряються відомості, зібрані в базі. Потім програми поводяться по-різному. WizRule застосовує виведені закономірності до тільки що проаналізованої бази і відшукує записи, де ці закономірності порушуються, тобто велика вірогідність того, що у зміст записів вкрались помилки. WizRule можна назвати засобом підтримки цілісності баз даних. WizWhy,

проаналізувавши базу даних, дає можливість користувачеві зайнятися передбаченнями і прогнозами. Людина вводить значення відомих йому параметрів, а WizWhy, ґрунтуючись на виявлених нею в базі закономірностях, видає найбільш вірогідні значення невідомих параметрів (рис. 7.11).

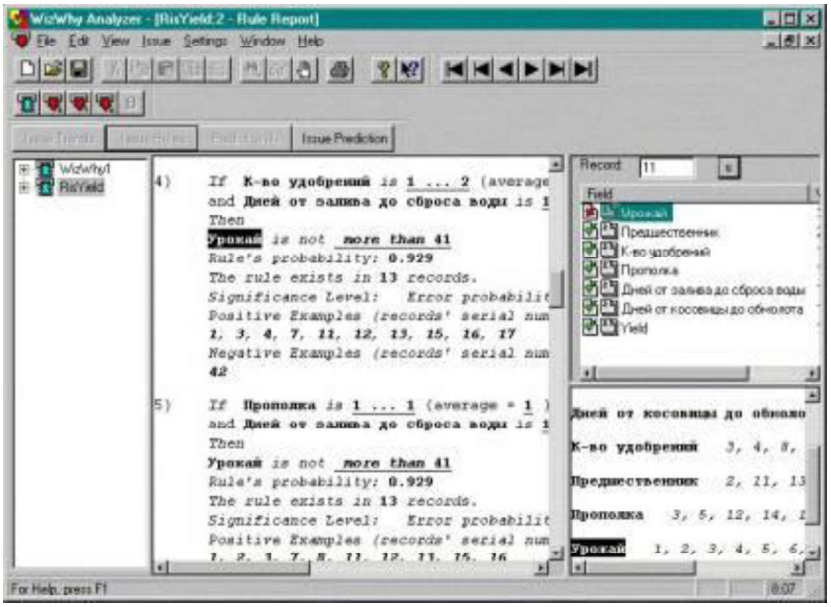


Рис 7.11. Система WizWhy виявила правила, що пояснюють низьку врожайність деяких сільськогосподарських ділянок.

Автор WizWhy стверджує, що його система виявляє всі логічні правила в даних. Насправді це, звичайно, не так. По-перше, максимальна довжина комбінації в «if - then» правилі в системі WizWhy дорівнює 6, і, по-друге, з самого початку роботи алгоритму виконується евристичний пошук простих логічних подій, на яких потім будується весь подальший аналіз. Зрозумівши ці особливості WizWhy, неважко було запропонувати просте тестове завдання, яке система не змогла взагалі вирішити. Інший момент - система видає рішення за прийнятний час лише для порівняно невеликої розмірності даних. Проте, система WizWhy є на сьогоднішній день одним з лідерів на ринку продуктів Data Mining. Це не позбавлено підстав. Система постійно демонструє вищі показники при вирішенні практичних задач, чим всі останні алгоритми.

7.2. Програмне забезпечення задач класифікації

Розробка програмних систем інтелектуального аналізу даних і, зокрема, комп'ютерною підтримки рішення задач класифікації активно ведуться в провідних зарубіжних країнах. Перш за все, це статистичні пакети обробки даних і візуалізації, в основі яких лежать методи різних розділів математичної статистики - перевірка статистичних гіпотез, регресійний аналіз, дисперсійний аналіз, аналіз часових рядів, і ін. Використання статистичних програмних продуктів стало стандартним і ефективним інструментом рішення задач класифікації, і, перш за все, початкового етапу досліджень, коли знаходяться значення різних усереднених показників, перевіряється статистична достовірність різних гіпотез, знаходяться регресійні залежності. В той же час статистичні підходи мають і істотні недоліки. Вони дозволяють оцінити статистичну достовірність значення параметра, гіпотези або залежності, проте самі методи обчислення величин, висунення гіпотез або знаходження залежностей мають очевидні обмеження. Перш за все, знаходяться усереднені по вибірці величини, що може бути досить грубим уявленням про аналізуємі або класифікуємі параметри. Будь-яка статистична модель використовує поняття «випадкових подій», «функцій розподілу випадкових величин» і тому подібне, тоді як взаємозв'язок між різними параметрами досліджуваних об'єктів, ситуацій або явищ є детермінованим. Саме використання статистичних методів передбачає наявність певного числа спостережень для обґрунтованості кінцевого результату, тоді як дане число може бути істотно більше можливого. Таким чином, при аналізі в принципі непредставимих даних, або на етапах початку накопичення даних, статистичні підходи стають неефективними як засіб аналізу і класифікації [6, 17, 25, 47].

Останніми роками з'явилися спеціалізовані пакети інтелектуального аналізу даних. Для даних пакетів характерна орієнтація на широкий круг практичних задач, а їх алгоритмічною основою є сукупність альтернативних моделей. Таким чином, на сьогоднішньому рівні розвитку методів рішення

задач інтелектуального аналізу даних і класифікації, переважною представляється дорога застосування програмних засобів, що включають основні існуючі підходи. В даному випадку підвищуються шанси підбору з наявних алгоритмів такого алгоритму, який забезпечить найбільш точне вирішення задач користувача на нових даних. Іншим важливим атрибутом систем аналізу і класифікації має бути наявність засобів автоматичного вирішення задач класифікації колективами алгоритмів. Дійсно, стандартною ситуацією є наявність декількох альтернативних алгоритмів або рішень, рівнозначних для користувача. Для вибору з них одного найбільш ефективного не вистачає інформації. Тоді природною альтернативою вибору є створення на базі наявних алгоритмів або рішень нових, більш перспективніших.

Розглянемо можливості та практику застосування найбільш відомих програмних пакетів при вирішенні задач класифікації.

Система PolyAnalyst. В пакеті реалізован багатий інструментарій для вирішення задач класифікації та для знаходження правил віднесення записів до одного з двох або до одного з декількох класів [79].

Одним з таких інструментів є модуль Stepwise Linear Regression (LR) - покрокова багатопараметрична лінійна регресія (рис. 7.12).

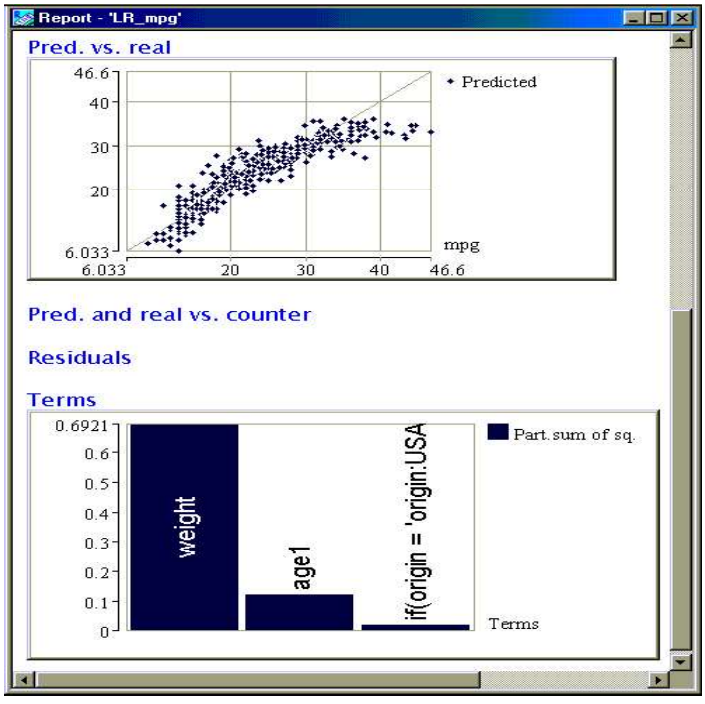


Рис. 7.12. Результати роботи модуля Stepwise Linear Regression.

Лінійна регресія як широко поширений метод статистичного дослідження, включена в багато статистичних пакетів і електронні таблиці. Проте, реалізація цього модуля в системі PolyAnalyst має свої особливості, а саме, автоматичний вибір найбільш значущих незалежних змінних і ретельна оцінка статистичної значущості результатів. Потрібно відмітити, що в даному випадку значущість відрізняється від значущості одиничної регресійної моделі, оскільки протягом одного запуску обчислювального процесу може бути перевірене велике число регресійних моделей. Алгоритм працює дуже швидко і може бути застосовний для побудови лінійних моделей на змішаних типах даних.

Модуль Memory based reasoning (MR) реалізує метод «найближчих сусідів». У системі PolyAnalyst використовується модифікація відомого алгоритму «метод найближчих сусідів» (рис. 7.13).

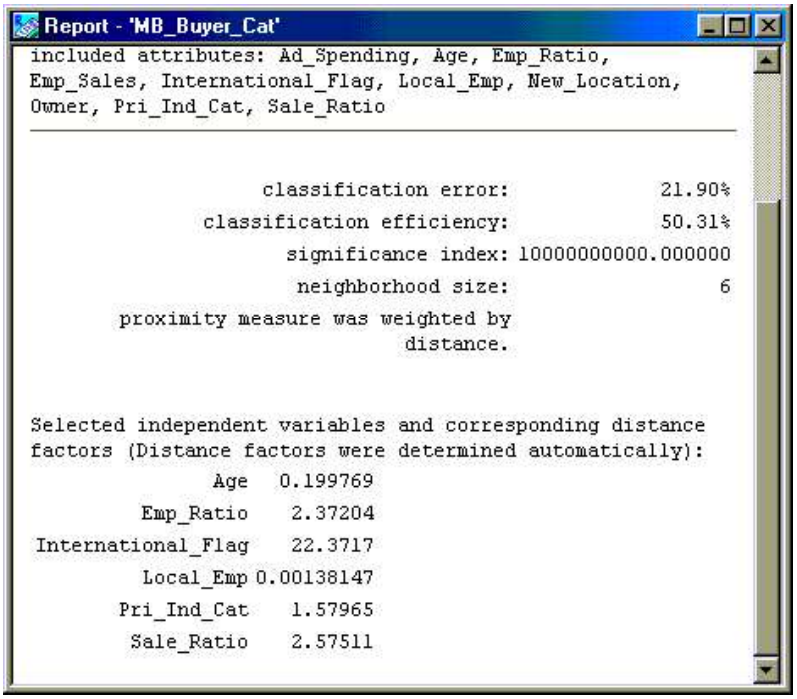


Рис. 7.13. Метод найближчих сусідів в системі PolyAnalyst.

Ідея методу дуже проста; для передбачення значення цільової змінної для даного запису, в навчальній таблиці з історичними даними, знаходяться «схожі» записи, для яких відомі значення цільової змінної, і обчислюється

середнє з цих значень, яке і вважається прогнозом. На практиці реалізація цієї ідеї зустрічається з трьома основними труднощами:

- що вважати мірою близькості записів;
- скільки записів брати для усереднювання;
- який метод усереднювання використовувати, звичайне або зважене усереднювання.

У системі PolyAnalyst оптимізація цих параметрів виконується на основі генетичних алгоритмів. У цьому і полягає відмінність даної реалізації алгоритму «найближчих сусідів» від відомих аналогів. Алгоритм MR використовується для передбачення значень числових змінних і категоріальних змінних, включаючи текстові (string data type), а також для класифікації на два або декілька класів.

Для задач класифікації система PolyAnalyst також застосовує модулі: Classify (CL) - класифікатор на основі нечіткої логіки (розділ 6), Discriminate (DS) – дикрімінація (розділ 6), Decision Tree (DT) - дерево рішень (розділ 4) та Decision Forest (DF) - ліси рішень.

Засоби аналізу STATISTICA Data Miner. Програмний комплекс STATISTICA включає величезний набір різних аналітичних процедур [76]. Для спрощення роботи користувача в пакет були вбудовані готові закінчені модулі аналізу даних, призначені для вирішення найбільш важливих і популярних задач: прогнозування, класифікації і так далі. Зокрема, це такі модулі як:

- General Classifier - класифікація. STATISTICA Data Miner включає повний пакет процедур класифікації: узагальнені лінійні моделі, дерева класифікації, регресійні дерева та інші.
- General Modeler/Multivariate Explorer - узагальнені лінійні, нелінійні і регресійні моделі. Даний елемент містить лінійні, нелінійні, узагальнені регресійні моделі і елементи аналізу дерев класифікації.

Окрім них, STATISTICA Data Miner містить набір спеціалізованих процедур Data Mining, які доповнюють лінійку інструментів Data Mining:

- General Classification and Regression Trees (GTrees) - узагальнені класифікаційні і регресійні дерева (GTrees). Модуль є повною реалізацією методів, розроблених Breiman, Friedman, Olshen і Stone. Окрім цього, модуль містить різного роду доопрацювання і доповнення, такі як оптимізації алгоритмів для великих об'ємів даних і так далі. Модуль є набором методів узагальненої класифікації і регресійних дерев.
- Interactive Classification and Regression Trees - інтерактивна класифікація і регресійні дерева. На додаток до модулів автоматичної побудови різного роду дерев, STATISTICA Data Miner також включає засоби для формування таких дерев в інтерактивному режимі.
- Boosted Trees - розширювані прості дерева. Останні дослідження аналітичних алгоритмів показують, що для деяких задач побудови «складних» оцінок, прогнозів і класифікацій використання послідовно збільшуваних простих дерев дає точніші результати, ніж нейроні мережі або складні цілісні дерева. Даний модуль реалізує алгоритм побудови простих збільшуваних (розширюваних) дерев.

Коротко проілюструємо схему роботи в Data Miner. Дії в Data Miner починаються з підміню «Добыча данных» в меню «Анализ» (рис. 7.14). Вибравши пункт «Добытчик данных – Мои процедуры», ми запустимо робоче середовище STATISTICA.

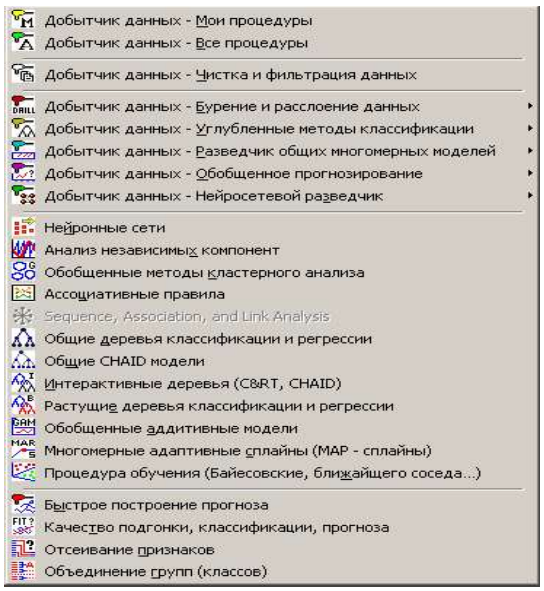


Рис. 7.14. Меню «Добытчик данных».

Після завантаження необхідного файлу у вікні діалогу «Виберите зависимые переменные и предикторы» вибираємо залежні змінні (неперервні і категоріальні) і предиктори (неперервні і категоріальні), виходячи із знань про структуру даних.

Запускаємо «Диспетчер узлов». У даному діалозі, показаному на рис. 7.15, можна вибрати вигляд аналізу або задати операцію перетворення даних.

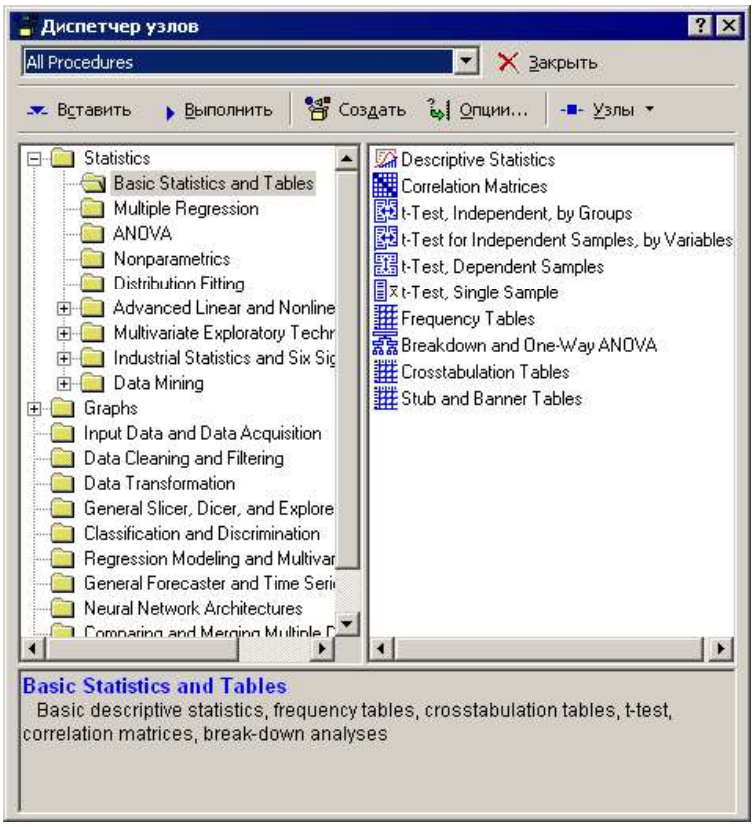


Рис. 7.15. Меню «Диспетчер узлов».

Диспетчер вузлів включає всі доступні процедури для видобутку даних. Всього доступні близько 260 методів фільтрації і очищення даних та методів аналізу. За умовчанням, процедури поміщені в папки і відсортовані відповідно до типа аналізу, який вони виконують. Проте користувач має можливість створити власну конфігурацію сортування методів. Для того, щоб вибрати необхідний аналіз, необхідно виділити його на правій панелі. У нижній частині діалогу дається опис вибираних методів.

Виберемо, для прикладу, Descriptive Statistics і Standard Classification Trees with Deployment (C And RT) . Вікно Data Miner виглядає таким чином (рис. 7.16).

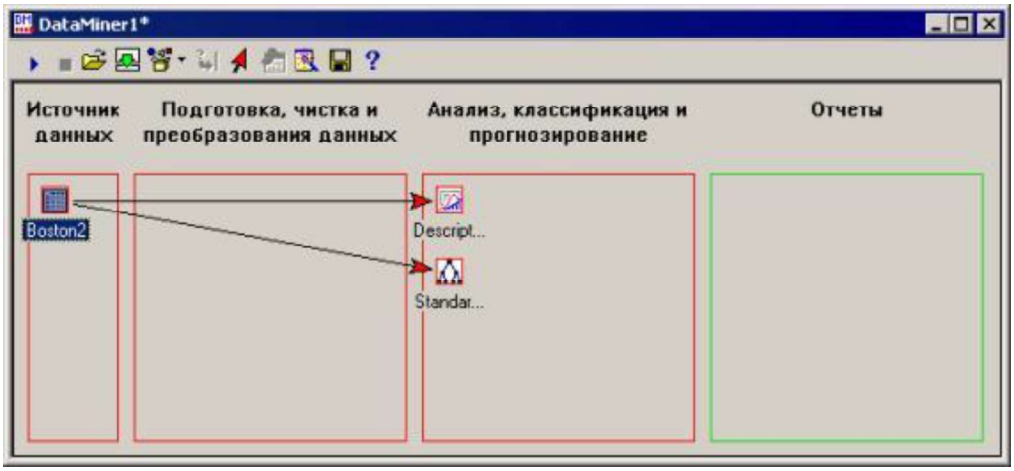


Рис. 7.16. Вікно Data Miner з вузлами вибраних аналізів.

Тепер виконаємо проект. Всі вузли, сполучені з джерелами даних активними стрілками, будуть проведені (рис. 7.17).

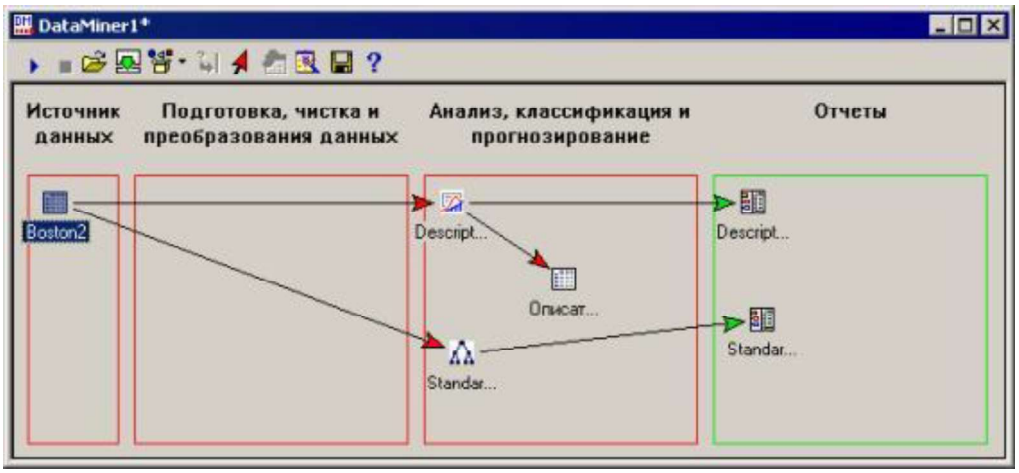


Рис. 7.17. Вікно Data Miner після виконання проекту.

Далі можна проглянути результати (у стовпці звітів). Детальні звіти створюються за умовчанням для кожного виду аналізу. Для робочих книг результатів доступна повна функціональність системи STATISTICA. Крім того, в диспетчерові вузлів STATISTICA Data Miner містяться всілякі процедури для класифікації і дискримінантного аналізу, регресійних моделей і багатовимірного аналізу, а також узагальнені часові ряди і прогнозування. Всі ці інструменти можна використовувати для проведення складного аналізу в автоматичному режимі, а також для оцінювання якості моделі.

Oracle Data Mining. Oracle Data Mining є модулем в Oracle Enterprise Edition. ODM підтримує всі етапи технології інтелектуального аналізу даних, включаючи постановку задачі, підготовку даних, автоматичну побудову моделей, аналіз і тестування результатів, використання моделей в реальних застосуваннях. Важливо, що моделі будуються автоматично на основі аналізу наявних даних про об'єкти, спостереження і ситуації за допомогою спеціальних алгоритмів. Основу модуля ODM складають процедури, що реалізовують різні алгоритми побудови моделей класифікації, регресії, прогнозування.

На етапі підготовки даних забезпечується доступ до будь-яких реляційних баз, текстових файлів, файлів формату SAS. Додаткові засоби перетворення і очищення даних дозволяють змінювати вигляд представлення, проводити нормалізацію значень, виявляти невизначені або відсутні значення. На основі підготовлених даних спеціальні процедури автоматично будують моделі для подальшого прогнозування, класифікації нових ситуацій, виявлення аналогій. ODM підтримує побудову п'яти різних типів моделей. Графічні засоби надають широкі можливості для аналізу отриманих результатів, верифікації моделей на тестових наборах даних, оцінки точності і стійкості результатів. Уточнені і перевірені моделі можна включати в існуючі додатки шляхом генерації їх описів на C, C++, Java, а також розробляти нові спеціалізовані додатки за допомогою засобу розробки Software Development Kit (SDK), що входить до складу середовища ODM .

Важливою особливістю системи ODM є його технічні характеристики: робота в архітектурі клієнт-сервер, широке використання техніки паралельних обчислень, висока міра масштабованості при збільшенні обчислювальних ресурсів. Oracle Data Mining підтримує наступний спектр алгоритмів класифікації:

- класифікаційні моделі - Na_ve Bayes, Adaptive Bayes Network;
- класифікації і регресійні моделі - Support Vector Machine.

Особливість алгоритмів, реалізованих в Oracle Data Mining, полягає в тому, що всі вони працюють безпосередньо з реляційними базами даних і не

вимагають вивантаження і збереження даних в спеціальних форматах. Окрім власне алгоритмів, в модуль ODM входять засоби підготовки даних, оцінки результатів, застосування моделей до нових наборів даних.

Засоби бізнес-аналізу в SQL Server. В програмному комплексі реалізовані засоби Data Mining, які доступні користувачам цієї СУБД. В якості прикладів алгоритмів розглянемо Microsoft Decision Trees і байєсівський алгоритм.

Алгоритм Microsoft Decision Tree є алгоритмом класифікації, що дозволяє прогнозувати як безперервні, так і дискретні атрибути на основі оцінки в процесі навчання моделі міри впливу вхідних атрибутів на прогнозований атрибут і побудови ієрархічної структури, яка базується на відповіді «так чи ні» на набір питань. Алгоритми побудови дерев рішень дозволяють передбачити значення якого-небудь параметра для заданого випадку (наприклад, чи поверне вчасно чоловік виданий йому кредит) на основі великої кількості даних про інші подібні випадки (зокрема, на основі відомостей про інших осіб, яким видавалися кредити).

Байєсівський алгоритм (Naive Bayes) дозволяє в процесі навчання моделі обчислити ймовірність, з якою кожен можливий стан кожного вхідного атрибуту приводить до кожного стану прогнозованого атрибуту, а потім використовувати результати розрахунків для прогнозування. Цей алгоритм підтримує лише дискретні атрибути.

Розробка моделей Data Mining здійснюється за допомогою SQL Server Business Intelligence Development Studio. Для виконання вказаної задачі використовується шаблон Analysis Services Project. Робота над моделлю починається з вказівки джерел даних і із створення представлення джерел даних, на основі якого генеруватиметься модель (рис. 7.18).

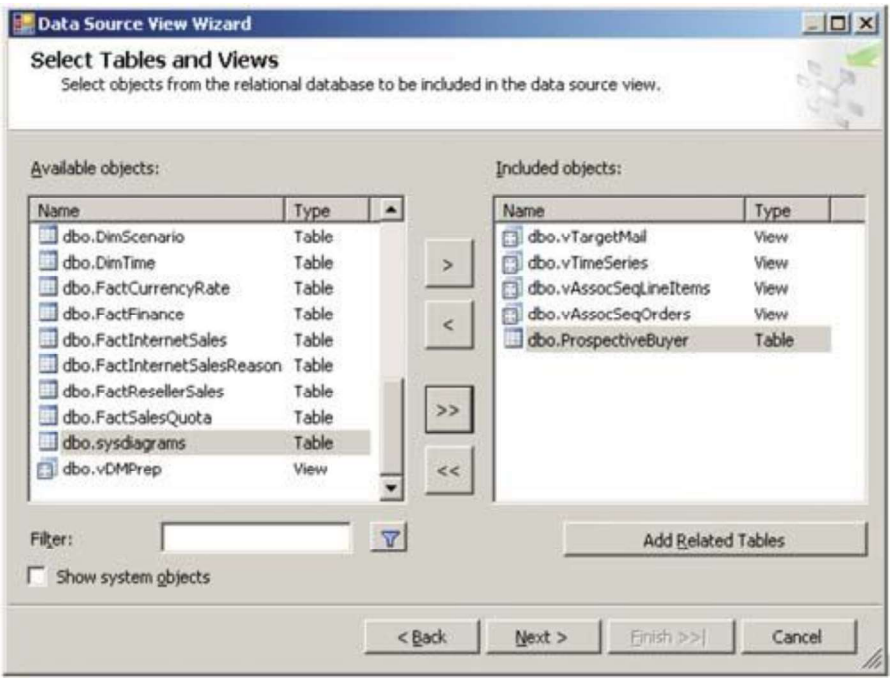


Рис. 7.18. Представлення джерел даних для подальшого створення моделей.

Створивши представлення джерел даних, можна приступати до розробки структури моделі (Mining Structure), вибравши відповідний пункт контекстного меню папки Mining structures. При відповіді на питання майстра вибираються модель Data Mining, використовуване представлення джерел даних, таблиця, що містить рядки, призначені для навчання моделі (кожна такий рядок в термінах Data Mining носить назву Case), прогнозовані атрибути і вхідні атрибути. Створеною структурою можна скористатися повторно, наприклад для розробки моделі, що використовує інший алгоритм. Це дозволяє змінити алгоритм Data Mining без повторного вибору таблиць і полів в тому випадку, якщо аналіз із застосуванням первинного вибраного алгоритму привів до виводу про необхідність його заміни. Описавши структуру моделей, ми можемо перенести моделі на сервер аналітичних служб і здійснити навчання моделей. Відзначимо, що процес навчання моделей виконується однократно, тоді як процес передбачення, що виконується багато разів, здійснюється досить швидко.

Коли один з алгоритмів побудови дерев рішень застосовується до набору вхідних даних, результат відображується у вигляді дерева. Подібні алгоритми дозволяють здійснити декілька рівнів такого розділення, ділячи отримані групи на дрібніші на підставі інших ознак до тих пір, поки значення, які передбачається передбачати, не стануть однаковими (або, в разі безперервного значення параметра, що передбачається, близькими) для всіх отриманих груп. Саме ці значення і застосовуються для здійснення передбачень на основі даної моделі.

Дія алгоритмів побудови дерев рішень базується на застосуванні методів регресійного і кореляційного аналізу. У реалізації Microsoft Decision Tree розділення виконується на основі найбільш високого для описуваних даних коефіцієнта кореляції між параметром, згідно якому відбувається розділення, і параметром, який надалі має бути передбачений. Проглянути отримане дерево рішень можна за допомогою закладки Decision Tree засобу Mining Model Viewer (мал. 7.19).

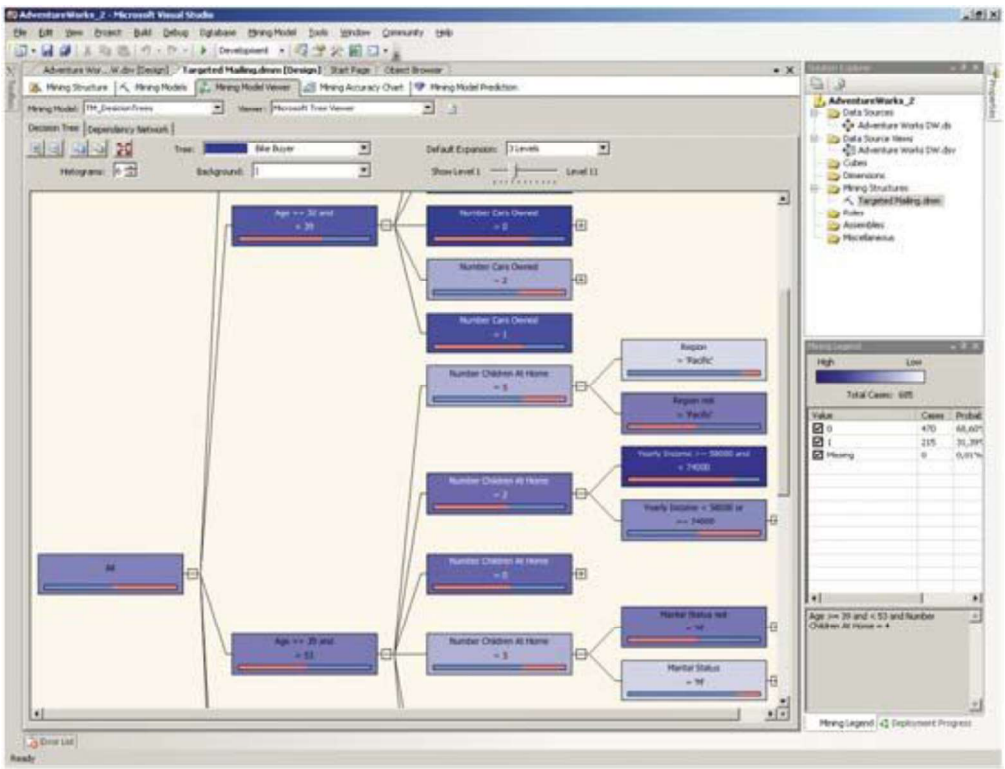


Рис. 7.19. Отримане дерево класифікації.

Ієрархія, що представлена в отриманому дереві, створена на основі класифікації даних за правилом «Якщо, то...», причому насиченість кольору гілок залежить від кількості вхідних даних, що попали у вказану гілку. При цьому на гістограмі в окремому вікні можна побачити процентний розподіл можливих результатів передбачення для вибраної гілки. Гілці дерева можна розкривати до самого нижнього рівня, при цьому на нижньому рівні в розподілі можливих результатів передбачення зазвичай переважає якесь одне значення. Іншими словами, алгоритм побудови дерев рішень дозволяє визначити набір значень характеристик, що дозволяють відокремити одну категорію даних від іншої, - цей процес називають *сегментацією*.

Для вивчення взаємозалежності атрибутів в даних, використаних для навчання моделі, можна скористатися закладкою Dependency Network засобу Mining Model Viewer (рис. 7.20).

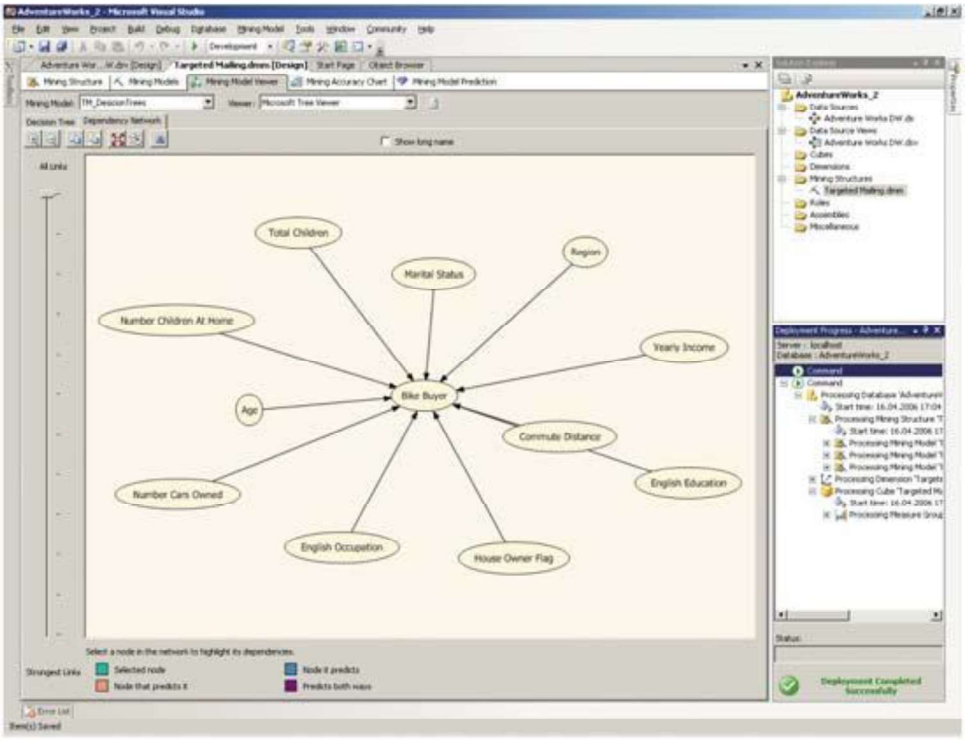


Рис. 7. 20. Вивчення взаємозалежності атрибутів.

У центральній частині представленої схеми розташований атрибут, що передбачається, довкола нього - атрибути, що використовуються для прогнозування. При цьому, переміщаючи покажчик в лівій частині вікна,

можна оцінити міру впливу кожного атрибуту на результат, що передбачається: чим нижче покажчик, тим менше зв'язків відображуватиметься на представленій схемі.

За допомогою байєсівського алгоритму можна виявити відмінності у впливі, що накладається на прогнозований атрибут різними станами вхідного атрибуту. Для вивчення взаємозалежності атрибутів, визначеної за допомогою байєсівського алгоритму, можна скористатися закладкою Dependency Network засобу Mining Model Viewer.

Закладка Attribute Profiles призначена для оцінки того, яким чином різні значення вхідних атрибутів впливають на атрибут, що передбачається (рис. 7.21).

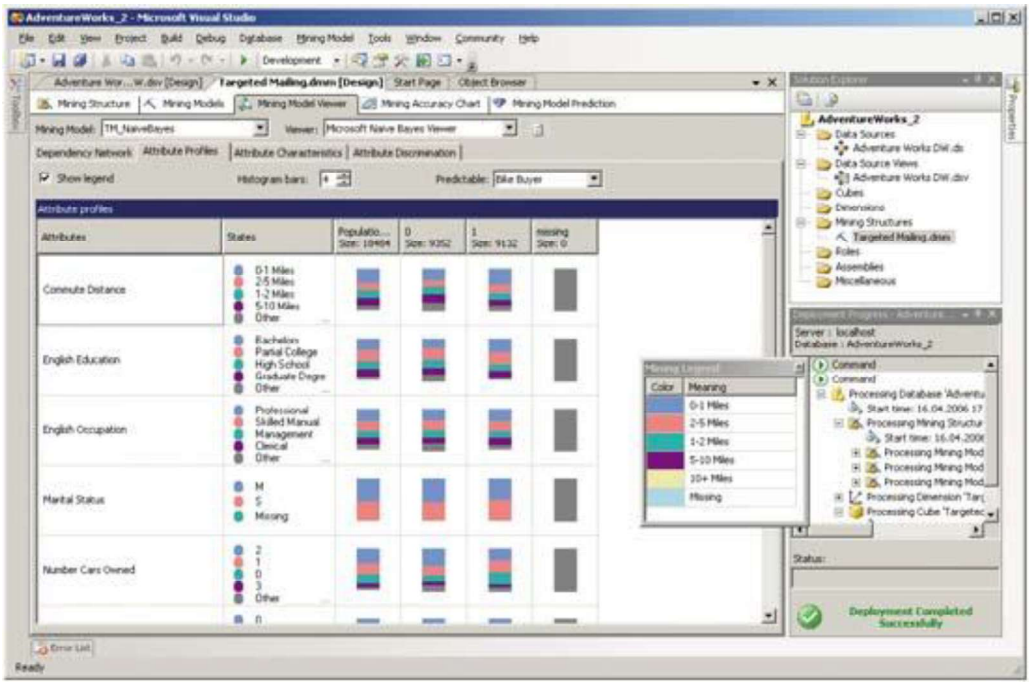


Рис. 7.21. Профілі атрибутів.

Побудувавши моделі і оцінивши їх точність, в тому випадку, якщо остання представляється задовільною, можна перейти безпосередньо до прогнозів. Для цієї мети зазвичай створюються запити на мові Data Mining Extensions (DMX). Код запиту можна як написати вручну, так і згенерувати за допомогою інструменту Prediction Query Builder на закладці Mining Model

Prediction інструменту Data Mining Designer. Типові результати класифікації для заздалегідь сформованого набору даних представлені на рис. 7.22.

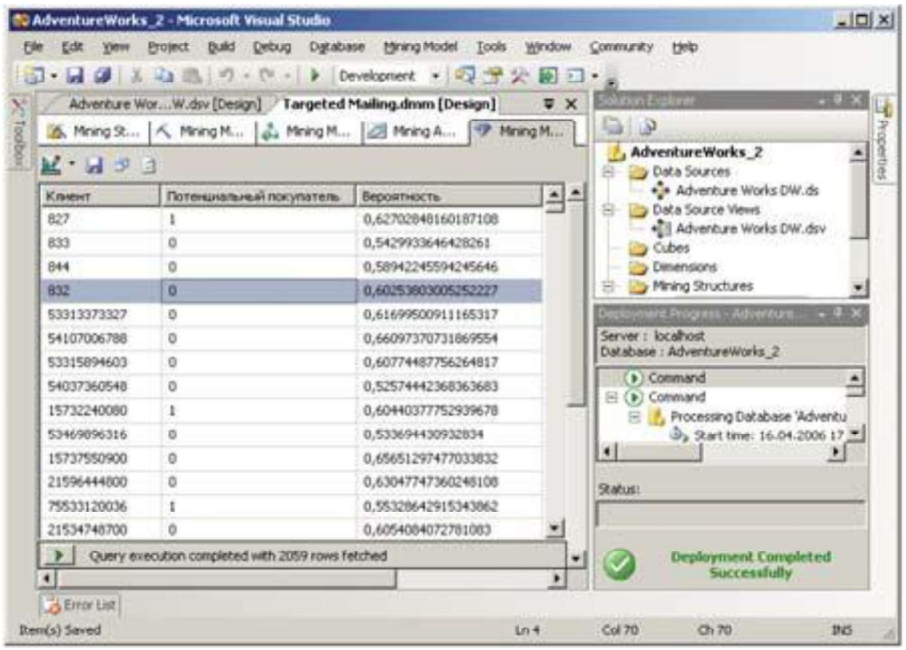
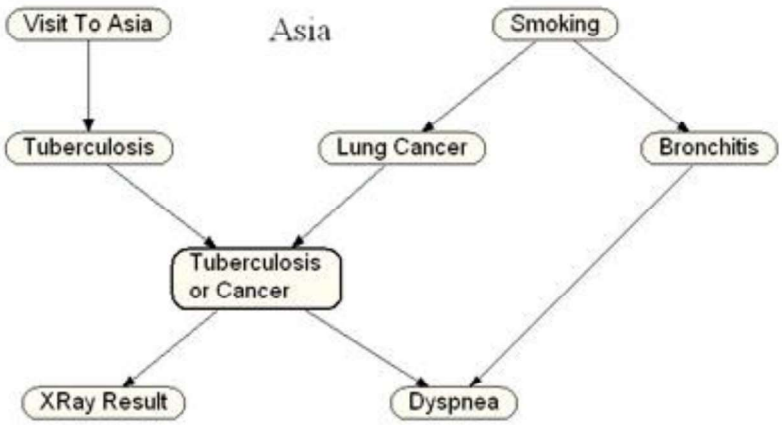


Рис. 7.22. Результати роботи моделі.

Байєсівські алгоритми. Компанія Norsys Software Corp. - спеціалізується в розробці програмного забезпечення для байєсівських мереж. Програма Netica - основне досягнення компанії, яка стала комерційно доступною в 1995 році. В даний час Netica є одним з найпоширених інструментів для розробки байєсівських мереж (рис. 7.23).



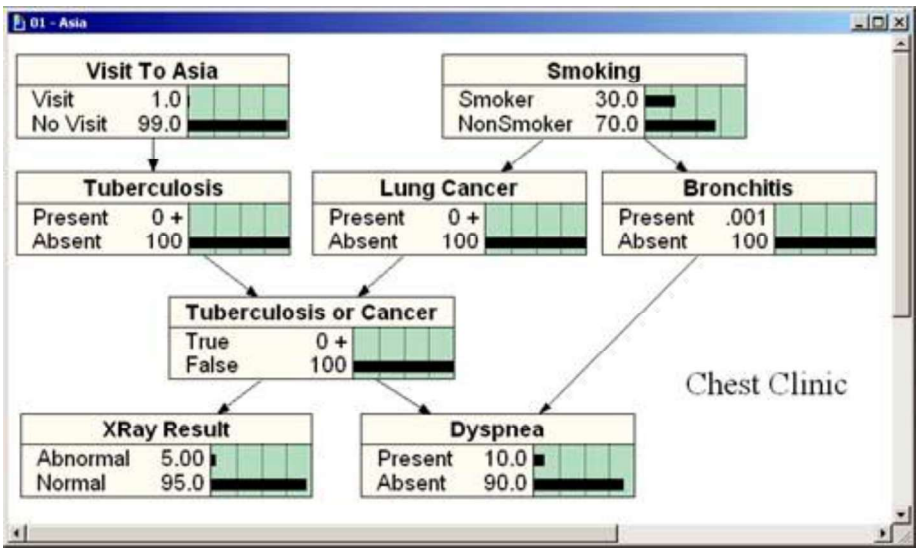


Рис. 7.23. Приклад байєсівської мережі в додатку Netica.

Netica - потужна, зручна в роботі програма для роботи з графовими імовірнісними моделями. Вона має інтуїтивний і приємний інтерфейс користувача для введення топології мережі. Співвідношення між змінними можуть бути задані, як індивідуальна ймовірність, у формі рівнянь, або шляхом автоматичного навчання з файлів даних. Створені мережі можуть бути використані незалежно, і як фрагменти крупніших моделей, формуючи тим самим бібліотеку модулів. При створенні мережевих моделей доступний широкий спектр функцій і інструментів. Багато операцій можуть бути зроблені декількома клацаннями миші, що робить систему Netica зручною для пошукових досліджень, для навчання і для простого перегляду, а також для навчання моделі байєсівської мережі.

Однією з успішних розробок компанії BayesWare Ltd є програмній комплекс Bayesware Discoverer, оснований на моделях байєсівських мереж. Програма здатна автоматично будувати причинну ймовірнісну модель на основі баз даних, використовуючи байєсівську мережу. Bayesware Discoverer автоматично визначає змінні, створює граф і визначає кількість залежностей як розподіл ймовірностей. Програма може бути застосована для дослідження різних сценаріїв відповідно до розробленої моделі, як наприклад, наявність

різних ймовірнісних розподілів. Вона має вбудовану діалогову 3D графіку, яка підтримує візуалізацію складних взаємодій (рис. 7.24).

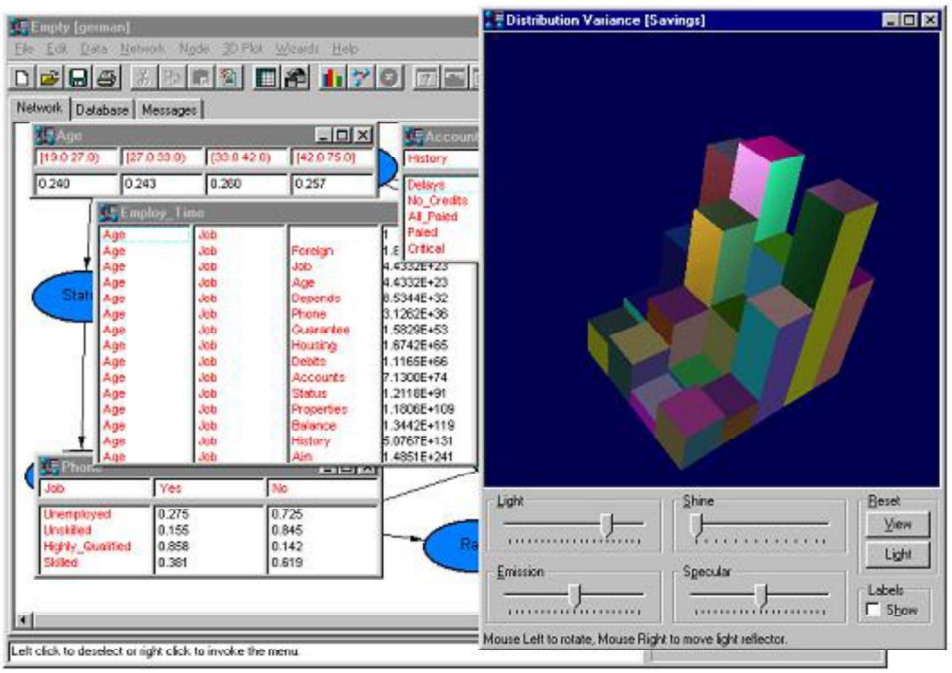


Рис. 7.24. Приклад роботи Bayesware Discoverer.

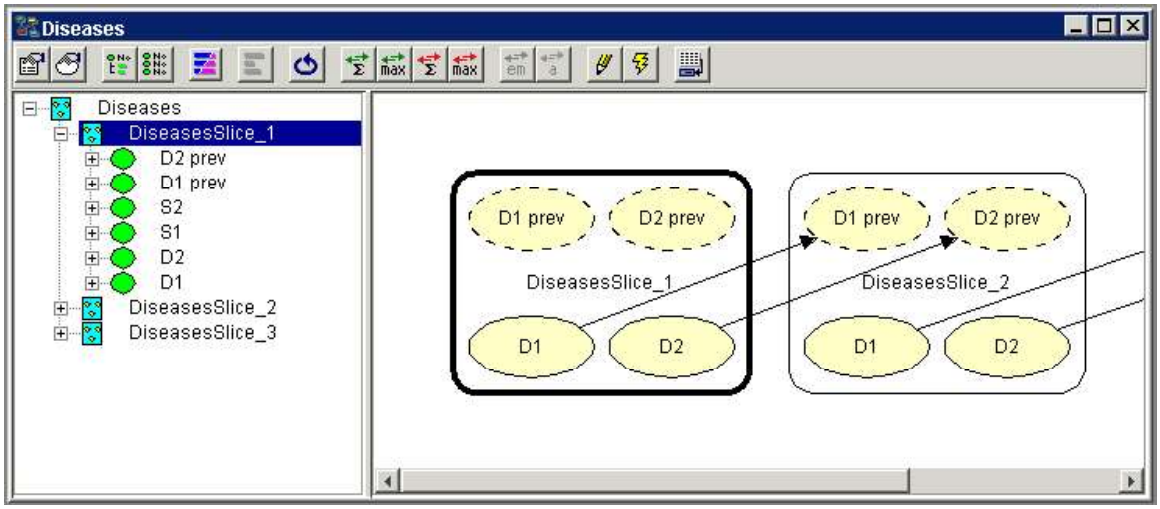
Основний продукт компанія Hugin Expert програмний комплекс Hugin почав створюватися під час робіт за проектом ESPRIT, в якому системи, засновані на знаннях, використовувалися для проблеми діагностування нервово-м'язових захворювань. До теперішнього моменту Hugin адаптована в 25 різних країнах, вона використовується у ряді різних областей, пов'язаних з аналізом рішень, підтримкою ухвалення рішень, передбаченням, діагностикою, управлінням ризиками і оцінками безпеки технологій.

Hugin є програмою реалізацією системи ухвалення рішень на основі байєсівських мереж довіри. Ця система має розвинений інтерфейс і дозволяє досить просто створювати бази знань і фактів. Використовує два основні режими роботи:

- режим редагування і побудови причинно-наслідкової мережі, а також заповнення таблиць умовної вірогідності, що є кількісним описом бази знань;
- режим розрахунку ймовірнісних оцінок для ухвалення рішення по всіх подіях, що входять в причинно-наслідкову мережу. Розрахунки можуть

здійснюватися як на основі класичної теорії Байєса, так і на основі методів теорії можливостей.

Програмний комплекс Hugin має можливість зв'язку з основними найбільш поширеними програмними засобами фірми Microsoft. Дана система має всі основні функції будь-якої інформаційної системи, включаючи такі як: зберігання даних, діагностика помилок в роботі і т. д (рис. 7.25).



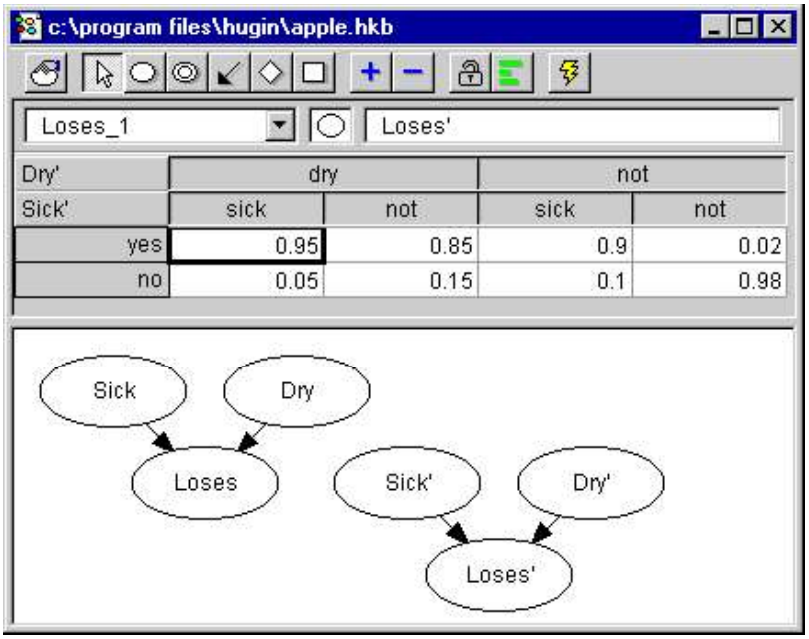


Рис. 7. 25. Приклад роботи програмного комплексу Hugin.

Байєсівські імовірнісні методи є істотним кроком вперед, в розвитку технологій інтелектуального аналіза даних. Вони дають зрозуміле пояснення

своїх виводів, допускають логічну інтерпретацію і модифікацію структури стосунків між змінними задачі, а також дозволяють в явній формі врахувати апріорний досвід експертів відповідної предметної області. Завдяки вдалому представленню у вигляді графів, байєсівські мережі вельми зручні в призначених для користувача застосуваннях.

Байєсівська методологія, насправді, ширше, ніж сімейство способів операції з умовною вірогідністю в орієнтованих графах. Вона включає також моделі з симетричними зв'язками (випадкові поля і решітки), моделі динамічних процесів (марківські ланцюги), а також широкий клас моделей з прихованими змінними, що дозволяють вирішувати імовірнісні задачі класифікації, розпізнавання образів і прогнозування. В найближчому майбутньому передбачається значно розширити вживання байєсівських мереж довіри. Наприклад, на одному з сайтів пошукових систем конструюються байєсівські мережі для моделювання успішних запитів, що поступають від користувачів. Ці мережі можуть поповнювати реєстраційний файл пошукового сервера категоріями передбачуваних цілей інформації, що призначаються, для забезпечення можливості передбачення модифікацій запитів.

Аналіз ринку програмних засобів для використання байєсівських методів демонструє як наявність безкоштовних і Open-Source продуктів (GeNIe & SMILE, OpenBayes, RISO, BANSY3, SamIam) так і комерційних продуктів (AgenaRisk Bayesian network tool, Bayesian network application library, Bayesia, BNet, Dezide, MSBNx, Bayes Net Toolbox for MatLab, dVelox).

Алгоритми міркувань на основі прецедентів. Потужний імпульс технології міркувань на основі прецедентів дала розробка комп'ютерної системи SMART. Система SMART призначена для технічної підтримки замовників корпорації COMPAQ. Коли замовник стикається з деякою проблемою, подробиці передаються в систему. Виконується початковий пошук в бібліотеці прецедентів, аби знайти випадки з подібними ознаками. При недоліку інформації система ставить додаткові питання. Як тільки певний поріг досягнутий (скажімо, прецедент збігається не менше, чим на 80%),

пропонується рішення від прецеденту. На додаток до цього, система може бути використана як інструмент навчання. Надалі COMPAQ розширила цю систему, просунувши її безпосередньо до покупців. Система QUICKSOURCE дозволяє користувачеві самому справлятися з проблемами і звертатися в центр підтримки в якості останнього притулку.

У системі KATE TOOLS компанії Acknosoft підтримується спрощений погляд на процес виводу. Вхідна інформація для KATE – це файл, який містить описи ознак і їх значення на спеціальній мові CASUAL. KATE може працювати із складними даними, представленими у вигляді структурованих об'єктів, відношень або навіть загальними знаннями про проблемної області. Але для виявлення схожості між прецедентами використовується одна проста метрика. Основний акцент робиться на відбір прецедентів за допомогою алгоритму «найближчого сусіда». KATE використовує версію алгоритму «найближчого сусіда» для обчислення метрики подібності. Близькість між двома випадками x і y, що мають p ознак обчислюється за формулою

$$Similarity(x,y) = \sqrt{\sum_{i=1}^{p} f(x_i, y_i)},$$

$$f(x,y) = \begin{cases} (x_i - y_i)^2, if\_x_i, y_i\_numeric; \\ (x_i \neq y_i), if\_x_i, y_i\_symbolic. \end{cases}$$

Система KATE не пропонує можливостей для автоматичної адаптації рішення. Перевірка коректності рішення неможлива, але є перевірка бази прецедентів на наявність контрприкладів. Все ж, KATE – це ефективна індустріальна система, яка дозволяє використовувати зважені ознаки при обчисленні метрики подібності, а також використовувати визначувану користувачем метрику. Її легко розширювати, тому що всі функції KATE доступні при підключенні супутніх динамічних бібліотек.

Інструментів Data Mining, що реалізовують метод k найближчих сусідів і CBR-метод, не надто багато. Серед найбільш відомих: CBR Express і Case Point (Inference Corp.), Apriori (Answer Systems), DP Umbrella (VYCOR Corp.),

KATE tools (Acknosoft, Франція), Pattern Recognition Workbench (Unica, США), а також деякі статистичні пакети, наприклад, Statistica.

CBR Express і CasePoint – продукти, призначені для розробки експертних систем, заснованих на прецедентах. CBR Express теж нагромаджує «досвід», забезпечуючи введення, супровід і динамічне додавання прецедентів, а також простий доступ до них за допомогою питань і відповідей. Обоє системи використовуються при автоматизації інформаційно-довідкових служб і «гарячих ліній», а також при створенні інтелектуальних програмних продуктів, систем доступу до інформації, систем публікації знань і так далі. При спілкуванні з системою спочатку вводиться простий запит, наприклад: «Мій комп'ютер не працює». Далі відбувається виділення ключових слів, пошук в базі прецедентів, і генерується перелік потенційних рішень. Користувачеві можуть бути також поставлені уточнюючі питання. Пропоновані варіанти вирішення проблеми можуть включати відео- або фотоматеріали. Технологія виводу по прецедентах є основою для практично безмежних застосувань, які нарощуються за рахунок постійного збору інформації (причому забезпечується поєднання структурованих і неструктурованих даних, включаючи мультимедіа). На думку компаній, що активно використовують цю технологію, таких як Nippon Steel, Lockheed і деяких інших, створюється самонавчальна колективна пам'ять, виключно зручна для накопичення і передачі професійного досвіду.

На останок зупинимося на двох важливих практичних додатках технологій класифікації.

В останнє десятиліття кількість електронних документів різко зросла, у зв'язку з чим виникла необхідність вирішення різних задач для зручної роботи з ними. Класифікація текстів (text categorization) та сортування текстових документів по заздалегідь визначених категоріях - одна з таких задач.

Методи класифікації текстів лежать на стику двох областей - інформаційного пошуку (information retrieval) і машинного навчання (machine learning). Загальні частини два цих підходів - способи представлення

документів і способи оцінки якості класифікації текстів, а відмінності полягають лише в способах власне пошуку. Згідно парадигмі машинного навчання, класифікуюче правило будується поступово на основі тренувальної колекції. Такий підхід забезпечує якість класифікації, порівнянну з якістю класифікації, вироблюваної людиною.

Сферами застосування цього підходу є фільтрація спаму, сортування новин, перевірка авторства, підбір ключових слів, складання інтернет-каталогів, контекстна реклама, зняття неоднозначності (автоматичні перекладачі), автоматичне анотування.

Текст представляється у вигляді мультимножини термів (слів). Кожному слову зіставлене деяке число – вага, яке є характеристикою зустрічаємості цього слова в тексті. Порядок слів, як правило, не враховується, враховується лише частота зустрічається слова в тексті і, можливо, інші ознаки, такі як «слово трапилося в заголовку», «слово виділене іншим кольором» і так далі. На підставі цих ознак кожному слову в тексті зіставляється його вага. Інколи проводиться нормалізація по документу для того, щоб сума квадратів всіх вагів в нім дорівнювала 1. Кожен текст - це вектор в багатовимірному просторі, координати - номери слів, значення координат - значення вагів. Розмірність вектора - це кількість слів, які зустрічаються в документах. Через те, що враховуються всі слова, які будь-коли зустрілися в документах, вектора виходять з величезною кількістю координат, більшість з яких нулі.

Допустимо, що існує такий лінійно незалежний набір документів D, що всі інші вектора документів виражаються як лінійна комбінація векторів цього набору. Тоді можна взяти множину D за базис; у такому базисі в кожного документа буде вже k координат. Таким чином, можна істотно скоротити розмірність.

Побудова класифікатора зазвичай полягає у визначенні функції $CSV_i : D \to [0,1]$, яка для кожного документа d_j повертає значення приналежності (categorization status value) d_j до c_i. Побудову точного класифікатора можна робити двома способами: відразу будувати функцію

$CSV_i : D \to \{T, F\}$ або спочатку обчислити аналогічну ранжируванню функцію $, CSV_i : D \to [0,1]$ а потім визначити поріг (threshold) τ_i такий, що $CSV_i \geq \tau_i$ і інтерпретується як T, а $CSV_i < \tau_i$ інтерпретується як F. Вибирати поріг можна декількома способами:

- пропорційний метод. Для кожної категорії c_i на колекції $T_к$ обчислюється, яка доля документів їй належить. Порогове значення вибирається так, щоб на інших колекціях доля документів, віднесених до c_i, була такою ж.
- метод k найближчих категорій. Кожен документ d_j вважається таким, що належить до k найближчим категоріям і відповідно цього вибирається порогове значення.

Іншою важливою проблемою сучасної інформатики є оцінка релевантності. Релевантність як відповідність між пошуковим запитом і знайденою інформацією є одним з фундаментальних понять при пошуку інформації. Розробка будь-якого пошукового двигуна в мережі Інтернет також передбачає необхідність рішення задачі оцінки релевантності. Зростання об'єму інформації в мережі Інтернет останнім часом істотно утрудняє процес виявлення релевантних документів і фільтрацію не релевантних документів. Особливо це стосується пошуку в певної предметної області. Пошукові двигуни загального призначення, по-перше, не надають користувачеві можливості звуження зони пошуку, а по-друге, через свою орієнтацію на роботу із слабо структурованими документами не в змозі виробляти оцінку релевантності з врахуванням специфіки конкретної області. Необхідність вирішення цих проблем привела до створення пошукових двигунів вузького призначення – вертикальних пошукових двигунів. Ці пошукові системи обмежуються пошуком в певної предметній області (вертикалі), намагаючись поліпшити якість надаваємих користувачеві результатів.

Для вирішення вказаної проблемі зручно використовувати байєсівські мережі довіри. Одна із задач, для якої успішно застосовуються байєсівські

мережі - задачі класифікації. Задачу оцінки релевантності також можна розглядати як задачу класифікації. Дійсно, можна розглянути кожен документ як об’єкт, що належить до однієї з двох областей, що не перетинаються: релевантні документи і не релевантні документи. В такому разі, задача оцінки релевантності документа запиту представляється у вигляді задачі віднесення його до одного з двох класів. В цьому випадку, приналежність документа до першого класу дозволяє свідчити, що цей документ є релевантним запиту.

Така класифікація здійснюється на підставі розрахунку ймовірності приналежності документа до тієї або іншої категорії. Ця ж вірогідність виступає і мірою релевантності, що дозволяє відсікати документі з низькою релевантністю, сортувати множину отриманих результатів та надавати користувачеві можливість вибору порогу релевантності.

7.3. Класичні технології кластеризації в Data Mining

Задача кластеризації схожа із задачею класифікації, є її логічним продовженням, але її відмінність в тому, що класи вивчаємого набору даних, заздалегідь не зумовлені. Синонімами терміну «кластеризація» є «автоматична класифікація», «навчання без вчителя» і «таксономія» [11, 13, 26].

Кластеризація призначена для розбиття сукупності об'єктів на однорідні групи (кластери або класи). Якщо дані вибірки представити як точки в ознаковому просторі, то задача кластеризації зводиться до визначення «згущувань точок». Мета кластеризації - пошук існуючих структур.

Кластеризація є описовою процедурою, вона не робить жодних статистичних висновків, але дає можливість провести розвідувальний аналіз і вивчити «структуру даних». Само поняття «кластер» визначене неоднозначно: у кожному дослідженні свої «кластери». Переводиться поняття кластер (cluster) як «скупчення», «гроздь». Кластер можна охарактеризувати як групу об'єктів,

що мають загальні властивості. Характеристиками кластера можна назвати дві ознаки:

- внутрішня однорідність;
- зовнішня ізольованість.

Питання, що задається аналітиками при вирішенні багатьох задач, полягає в тому, як організувати дані в наглядні структури, тобто розвернути таксономії. Результатом таксономії є деревоподібна ієрархічна структура. При цьому кожен об'єкт характеризується перерахуванням всіх кластерів, яким він належить, зазвичай від великого до дрібного. Візуально таксономія представляється у вигляді графіка, названого *дендрограмою*. Класичним прикладом таксономії на основі схожості є біноміальна номенклатура живих істот, запропонована Карлом Ліннеєм в середині XVIII століття. Аналогічні систематизації будуються в багатьох областях знання, аби упорядкувати інформацію про велику кількість об'єктів. Найбільше застосування кластеризація спочатку отримала в таких науках як біологія, антропологія, психологія. Для вирішення економічних задач кластеризація тривалий час мало використовувалася із-за специфіки економічних даних і явищ.

У 1925 р. гідробіолог П. В. Терентьєв розробив так званий «метод кореляційних плеяд», призначений для угрупування корелюючих ознак. Цей метод дав поштовх розвитку методів угрупування за допомогою графів. Термін «кластерний аналіз» вперше був запропонований Тріоном. На початку 50-х років з'явилися публікації Р. Люїса, Е. Фікса і Дж. Ходжеса по ієрархічних алгоритмах кластерного аналізу. Помітний поштовх розвитку робіт по кластерному аналізу дали роботи Р. Розенблатта по розпізнаючому пристрою (персептрону), які поклали початок розвитку теорії «розпізнавання образів без вчителя».

Важливим кроком до розробки методів кластеризації з'явилася книга «Принципи чисельної таксономії», опублікована в 1963г. двома біологами - Робертом Сокелом і Пітером Снітом. Автори цієї книги виходили з того, що для створення ефективних біологічних класифікацій процедура кластеризації

повинна забезпечувати використання всіляких показників, що характеризують досліджувані організми, виробляти оцінку міри схожості між цими організмами і забезпечувати розміщення схожих організмів в одну і ту ж групу. При цьому сформовані групи мають бути досить «локальні», тобто схожість об'єктів усередині груп повинна перевершувати схожість груп між собою. Подальший аналіз виділених угрупувань, на думку авторів, може з'ясувати, чи відповідають ці групи різним біологічним видам. Іншими словами, Сокел і Сніт передбачали, що виявлення структури розподілу об'єктів в групи, допомагає встановити процес утворення цих структур. А відмінність і схожість організмів різних кластерів (груп) можуть служити базою для осмислення еволюційного процесу, що відбувався, і з'ясування його механізму.

В цей ж час була запропонована множина алгоритмів таких авторів, як Дж. Мак-Кин, Г. Болл і Д. Холл за методами *k* середніх; Г. Ланса і В. Уїльямса, Н. Джардайна та ін. - за ієрархічними методами. Помітний внесок у розвиток методів кластерного аналізу внесли і вітчизняні учені - Е. М. Браверман, А. А. Дорофеюк, Л. А, Растригін, Ю. І. Журавльов та ін. Зокрема, в 60-70 рр. великою популярністю користувалися багаточисельні алгоритми розроблені математиками Н. Г. Загоруйко і Г. С. Лобовим. Це такі широко відомі алгоритми, як FOREL, BIGFOR, KRAB, NTTP, DRET, TRF та ін. На їх основі був створений спеціалізований пакет програм ОТЕКС. По приблизних оцінках фахівців число публікацій по кластерному аналізу і його застосуванням в різних областях знання подвоюється кожні три роки.

Які ж причини настільки бурхливого інтересу до цього виду аналізу? Об'єктивно існують три основні причини цього явища. Це поява потужної обчислювальної техніки, без якої кластерний аналіз реальних даних практично не реалізовується. Друга причина полягає в тому, що сучасна наука все сильніше спирається в своїх побудовах на класифікацію. Причому цей процес усе більш поглиблюється, оскільки паралельно цьому йде все більша спеціалізація знань, яка неможлива без досить об'єктивної класифікації. Третя причина - поглиблення спеціальних знань неминуче приводить до збільшення

кількості змінних, що враховуються при аналізі тих або інших об'єктів і явищ. Внаслідок цього суб'єктивна класифікація, яка раніше спиралася на досить малу кількість ознак, що враховувалися, часто виявляється вже ненадійною. А об'єктивна класифікація, зі все зростаючим набором характеристик об'єкту, вимагає використання складних алгоритмів кластеризації, які можуть бути реалізовані лише на базі сучасних комп'ютерів. Саме ці причини і породили «кластерний бум».

Формальна постановка задачі *кластеризації* (або навчання без вчителя) полягає в наступному. Є навчальна вибірка $X^l = (x_1,...,x_l) \subset X$ і функція відстані між об'єктами $\rho(x,x')$. Потрібно розбити вибірку на підмножини, що не перетинаються, які назвемо *кластерами*, так, щоб кожен кластер складався з об'єктів, близьких по метриці ρ, а об'єкти різних кластерів істотно відрізнялися. При цьому кожному об'єкту $x_i \in X^l$ приписується мітка (номер) кластера y_i.

Алгоритм кластеризації - це функція $a: X \to Y$ а: X > Y, яка будь-якому об'єкту $x \in X$ ставить у відповідність мітку кластера $y \in Y$. Множина міток Y в деяких випадках відома заздалегідь, проте частіше ставиться задача визначити оптимальне число кластерів, з точки зору того або іншого критерію якості кластеризації.

Потреба в обробці великих масивів даних в Data Mining привела до формулювання вимог, яким, по можливості, повинен задовольняти алгоритм кластеризації. Розглянемо їх:

- мінімальна можлива кількість проходів по базі даних;
- робота в обмеженому об'ємі оперативної пам'яті комп'ютера;
- роботу алгоритму можна перервати із збереженням проміжних результатів, аби продовжити обчислення пізніше;
- алгоритм повинен працювати, коли об'єкти з бази даних можуть витягуватися лише в режимі однонаправленого курсора (тобто в режимі навігації по записах).

Рішення задачі кластеризації принципове неоднозначно, і тому є декілька причин. По-перше, не існує однозначно найкращого критерію якості кластеризації. Відомий цілий ряд досить розумних критеріїв, а також ряд алгоритмів, що не мають чітко вираженого критерію, але що здійснюють досить розумну кластеризацію «по побудові». Всі вони можуть давати різні результати. По-друге, число кластерів, як правило, невідоме заздалегідь і встановлюється відповідно до деякого суб'єктивного критерію. По-третє, результат кластеризації істотно залежить від метрики ρ, вибір якої, як правило, також суб'єктивний і визначається експертом.

Вибір відстані між об'єктами є вузловим моментом дослідження, від нього багато в чому залежить остаточний варіант розбиття об'єктів на класи при даному алгоритмі розбиття. Існує декілька методів визначення функції відстані.

Відстань Евкліда. Найбільш пряма дорога обчислення відстаней між об'єктами полягає в обчисленні відстаней Евкліда. Вона є геометричною відстанню в багатовимірному просторі і обчислюється таким чином

$$d(X_j, X_i) = \left[\sum_{k=1}^{N} (x_{ki} - x_{kj})^2 \right]^{1/2}.$$

Відмітимо, що відстань Евкліда обчислюється по початкових, а не за стандартизованими даними. Це звичайний спосіб її обчислення, який має певні переваги (наприклад, відстань між двома об'єктами не змінюється при введенні в аналіз нового об'єкту, який може виявитися викидом). Проте, на відстані можуть сильно впливати відмінності між осями, по координатах яких обчислюються ці відстані.

Відстань міських кварталів (манхэттенська відстань). Ця відстань є просто середньою різниць по координатах. В більшості випадків ця міра відстані приводить до таких же результатів, як і звичайна відстань Евкліда. Проте відзначимо, що для цієї міри вплив окремих великих різниць (викидів) зменшується (оскільки вони не зводяться в квадрат). Манхеттенська відстань обчислюється за формулою

$$d(X_j, X_i) = \sum_{k=1}^{N} | x_{ki} - x_{kj} |.$$

Відстань Чебишева. Ця відстань може виявитися корисною, коли бажають визначити два об'єкти як «різні», якщо вони розрізняються по якій-небудь одній координаті (яким-небудь одним виміром). Відстань Чебишева обчислюється за формулою

$$d(X_j, X_i) = \max | x_{ki} - x_{kj} |$$

Степенна відстань (відстань Мінковського). Інколи виникає необхідність прогресивно збільшити або зменшити вагу, що відноситься до розмірності, для якої відповідні об'єкти сильно відрізняються. Це може бути досягнуто з використанням степенної відстані, яка обчислюється за формулою:

$$d(X_j, X_i) = (\sum_{k=1}^{N} | x_{ki} - x_{kj} |^p)^{1/r}$$

де r і p - параметри, що визначаються користувачем. Параметр p відповідальний за поступове зважування різниць по окремих координатах, параметр r відповідальний за прогресивне зважування великих відстаней між об'єктами. Якщо обоє параметра рівні двом, то ця відстань збігається з відстанню Евкліда.

Відсоток незгоди. Ця міра використовується в тих випадках, коли дані є категоріальними. Ця відстань обчислюється за формулою

$$d(X_j, X_i) = (Кількість\_x_{ki} \neq x_{kj}) / k.$$

Коли кожен об'єкт є окремим кластером, відстані між цими об'єктами визначаються вибраною мірою. Виникає наступне питання - як визначити відстані між кластерами? Існують різні правила, названі методами об'єднання або зв'язки для двох кластерів.

Метод ближнього сусіда або одиночний зв'язок. Тут відстань між двома кластерами визначається відстанню між двома найбільш близькими об'єктами (найближчими сусідами) в різних кластерах. Цей метод дозволяє виділяти кластери скільки завгодно складної форми за умови, що різні частини таких кластерів сполучені ланцюжками близьких один до одного елементів. В

результаті роботи цього методу кластери представляються довгими «ланцюжками» або «волокнистими» кластерами, «зчепленими разом» лише окремими елементами, які випадково виявилися ближчими за останніх один до одного.

Метод найбільш віддалених сусідів або повний зв'язок. Тут відстані між кластерами визначаються найбільшою відстанню між будь-якими двома об'єктами в різних кластерах (тобто «найбільш віддаленими сусідами»). Метод добре використовувати, коли об'єкти дійсно походять з різних «гаїв». Якщо ж кластери мають в деякому роді подовжену форму або їх природний тип є «ланцюжковим», то цей метод не слід використовувати.

Метод Варда (Ward's method). В якості відстані між кластерами береться приріст суми квадратів відстаней об'єктів до центрів кластерів, отримуваний в результаті їх об'єднання. На відміну від інших методів кластерного аналізу для оцінки відстаней між кластерами, тут використовуються методи дисперсійного аналізу. На кожному кроці алгоритму об'єднуються такі два кластери, які приводять до мінімального збільшення цільової функції, тобто внутрішньогрупової суми квадратів. Цей метод направлений на об'єднання близько розташованих кластерів і «прагне» створювати кластери малого розміру.

Метод незваженого попарного середнього (unweighted pair-group method using arithmetic averages, UPGMA). В якості відстані між двома кластерами береться середня відстань між всіма парами об'єктів в них. Цей метод слід використовувати, якщо об'єкти дійсно походять з різних «гаїв», у випадках присутності кластерів типа «ланцюжка», при припущенні нерівних розмірів кластерів.

Метод зваженого попарного середнього (weighted pair-group method using arithmetic averages, WPGMA). Цей метод схожий на метод незваженого попарного середнього, різниця полягає лише в тому, що тут в якості вагового коефіцієнту використовується розмір кластера (число об'єктів, що містяться в

кластері). Цей метод рекомендується використовувати саме за наявності припущення про кластери різних розмірів.

Кластеризація (навчання без вчителя) відрізняється від класифікації (навчання з вчителем) тим, що мітки вхідних об'єктів y_i спочатку не задані, і навіть може бути невідома само множина Y. У цьому сенсі задача кластеризації ще в більшій мірі некоректно поставлене, чим задача класифікації (рис. 7.26).

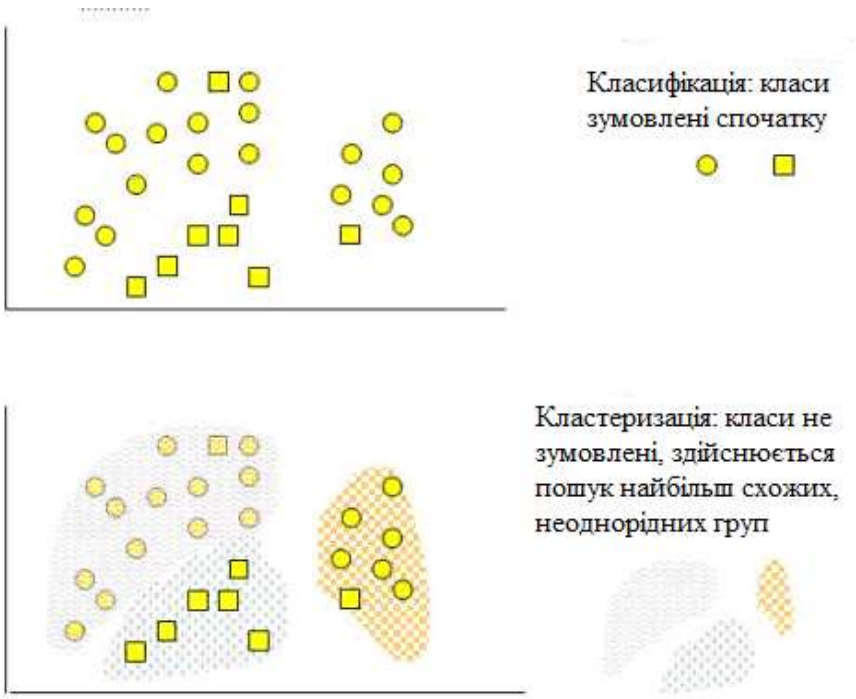


Рис. 7.26. Порівняння задач класифікації і кластеризації.

Цілі кластеризації можуть бути різними залежно від особливостей конкретної прикладної задачі:

- зрозуміти структуру множини об'єктів X^l, розбивши її на групи схожих об'єктів. Спростити подальшу обробку даних і ухвалення рішень, працюючи з кожним кластером окремо (стратегія «розділяй і володарюй»);
- скоротити об'єм зберігаємих даних в разі надвеликої вибірки X^l, залишивши по одному найбільш типовому представникові від кожного кластера;

- виділити нетипові об'єкти, які не підходять ні до одного з кластерів. Цю задачу називають однокласовою класифікацією, виявленням нетиповості або новизни (novelty detection).

У першому випадку число кластерів прагнуть зробити поменше. У другому випадку важливіше забезпечити високу міру схожості об'єктів усередині кожного кластера, а кластерів може бути скільки завгодно. У третьому випадку найбільший інтерес представляють окремі об'єкти, що не вписуються ні в один з кластерів.

Кластери можуть бути такими, що не перетинаються, або ексклюзивними (non-overlapping, exclusive), і такими, що перетинаються, (overlapping) (мал. 7.27).

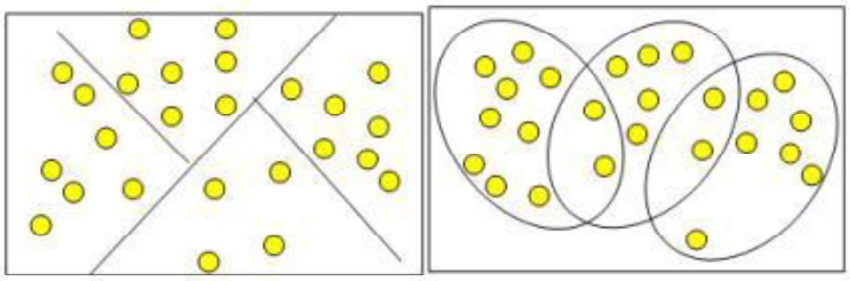


Рис. 7.27. Кластери, що перетинаються і не перетинаються.

Слід зазначити, що в результаті застосування різних методів кластерного аналізу можуть бути отримані кластери різної форми. Наприклад, можливі кластери типа «ланцюжка», тобто кластери представлені довгими «ланцюжками», кластери подовженої форми і так далі, а деякі методи можуть створювати кластери довільної форми.

На сьогоднішній день розроблена більше сотні різних алгоритмів кластеризації. Приведемо коротку характеристику підходів до кластеризації.

1. Алгоритми, засновані на розділенні даних (Partitioning Algorithms), в т.ч. ітеративні:
   - розділення об'єктів на k кластерів;
   - ітеративний перерозподіл об'єктів для поліпшення кластеризації.
2. Ієрархічні алгоритми (Hierarchy Algorithms):

- агломерація (Agglomerative Nesting): кожен об'єкт спочатку є кластером, кластери, з'єднуючись один з одним, формують більший кластер і так далі;
- дивізімні методи (Divisive Analysis): характеризуються послідовним розділенням кластера, що складається зі всіх об'єктів, і відповідним збільшенням числа кластерів, що в результаті приводить до створення послідовність розщеплюючих груп.

3. Методи, засновані на концентрації об'єктів (Density-based methods):
   - засновані на можливості з'єднання об'єктів;
   - ігнорують шуми, знаходження кластерів довільної форми.
4. Грід-методи (Grid-based methods):
   - квантування об'єктів в грід-структури.
5. Модельні методи (Model-based):
   - використання моделі для знаходження кластерів, найбільш відповідних даним.
6. Методи за способом аналізу даних:
   - чіткі;
   - нечіткі.
7. Методи по кількості застосування алгоритмів кластеризації:
   - з одноетапною кластеризацією;
   - з багатоетапною кластеризацією.
8. Методи по можливості розширення об'єму оброблюваних даних:
   - масштабовані;
   - не масштабовані.
9. Методи за часом виконання кластеризації:
   - потокові (on-line);
   - не потокові (off-line).

Важливою проблемою кластеризації є оцінка її якості. Задачу кластеризації можна ставити як задачу дискретної оптимізації: необхідно так

приписати номери кластерів y_i об'єктам x_i, аби значення вибраного функціонала якості прийняло найкраще значення. Існує багато різновидів функціоналів якості кластеризації, але немає «найправильнішого» функціонала. По суті справи, кожен метод кластеризації можна розглядати як точний або наближений алгоритм пошуку оптимуму деякого функціонала.

Середня внутрішньокластерна відстань має бути якомога менше

$$F_0 = \frac{\sum_{i<j}[y_i = y_j]\rho(x_i, x_j)}{\sum_{i<j}[y_i = y_j]} \to \min .$$

Середня міжкластерна відстань має бути якомога більше

$$F_1 = \frac{\sum_{i<j}[y_i \neq y_j]\rho(x_i, x_j)}{\sum_{i<j}[y_i \neq y_j]} \to \max .$$

Якщо алгоритм кластеризації обчислює центри кластерів μ_y, $y \in Y$, то можна визначити обчислювально ефективніші функціонали.

На практиці обчислюють відношення пари функціоналів, аби врахувати як міжкластерні, так і внутрішньокластерні відстані

$$F_o / F_1 \to \min .$$

Оцінка якості кластеризації може бути проведена на основі наступних процедур:

- ручна перевірка;
- встановлення контрольних точок і перевірка на отриманих кластерах;
- визначення стабільності кластеризації шляхом додавання в модель нових змінних;
- створення і порівняння кластерів з використанням різних методів. Різні методи кластеризації можуть створювати різні кластери, і це є нормальним явищем. Проте створення схожих кластерів різними методами вказує на правильність кластеризації.

Кластерний аналіз застосовується в різних областях. Він корисний, коли потрібно класифікувати велику кількість інформації. Так, в медицині використовується кластеризація захворювань, лікування захворювань або їх

симптомів, а також таксономія пацієнтів, препаратів і так далі. У археології встановлюються таксономії кам'яних споруд і древніх об'єктів. У менеджменті прикладом задачі кластеризації буде розбиття персоналу на різні групи, класифікація споживачів і постачальників, виявлення схожих виробничих ситуацій, при яких виникає брак. У соціології задача кластеризації - розбиття респондентів на однорідні групи. Загалом, всякий раз, коли необхідно класифікувати «гори» інформації до придатних для подальшої обробки груп, кластерний аналіз виявляється вельми корисним і ефективним.

Досить широко кластерний аналіз застосовується в маркетингових дослідженнях - як в теоретичних дослідженнях, так і практиці вирішення проблем групування різних об'єктів. При цьому вирішуються питання про групи клієнтів, продуктів і так далі. Так, однією з найбільш важливих задач при використанні кластерного аналізу в маркетингових дослідженнях є аналіз поведінки споживача [48], а саме: групування споживачів в однорідні класи для здобуття максимально повного уявлення про поведінку клієнта з кожної групи і про чинники, що впливають на його поведінку. Ця проблема детально описана в роботах Клакстона, Фрая, Портіса, Кіля і Лейтона. Важливу задачу, яку може вирішити кластерний аналіз, є позиціювання, тобто визначення ніші, в якій слід позиціювати новий продукт, пропонований на ринку. В результаті вживання кластерного аналізу будується карта, по якій можна визначити рівень конкуренції в різних сегментах ринку і відповідні характеристики товару для можливості попадання в цей сегмент. За допомогою аналізу такої карти можливе визначення нових, незайнятих ніш на ринку, в яких можна пропонувати існуючі товари або розробляти нові. Кластерний аналіз також може бути зручний, наприклад, для аналізу клієнтів компанії. Для цього всі клієнти групуються в кластери, і для кожного кластера виробляється індивідуальна політика. Такий підхід дозволяє істотно скоротити об'єкти аналізу, і, в той же час, індивідуально підійти до кожної групи клієнтів.

Задачі кластерного аналізу можна об'єднати в наступні групи:

- розробка типології або класифікації;

- дослідження корисних концептуальних схем групування об'єктів;
- представлення гіпотез на основі дослідження даних;
- перевірка гіпотез або досліджень для визначення, чи дійсно типи (групи), виділені тим або іншим способом, присутні в наявних даних.

Як правило, при практичному використанні кластерного аналізу одночасно вирішується декілька з вказаних задач.

Розглянемо приклад процедури кластерного аналізу. Допустимо, ми маємо набір даних A, що складається з 14-ти прикладів, в яких є по дві ознаки X та Y. Дані по ним приведені в таблиці 7.1.

Табл.. 7.1

Набір даних A

№ приклада	Ознака X	Ознака Y
1	27	19
2	11	46
3	25	15
4	36	27
5	35	45
6	10	43
7	11	44
8	36	24
9	26	14
10	26	45
11	9	23
12	33	15
13	27	16
14	10	48

Представимо змінні X та Y у вигляді діаграми розсіювання (рис. 7. 28).

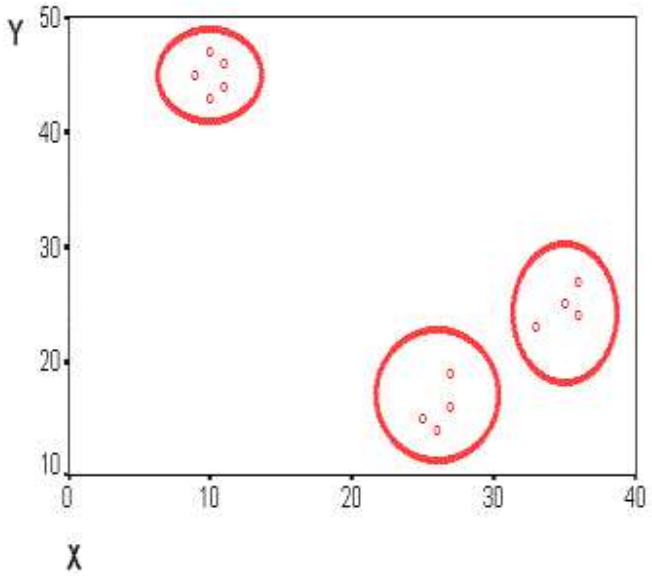


Рис. 7.28. Діаграма розсіювання змінних X та Y.

На рисунку бачимо декілька груп «схожих» прикладів. Приклади (об'єкти), які по значеннях X і Y «схожі» один на одного, належать до однієї групи (кластеру); об'єкти з різних кластерів не схожі один на одного. Критерієм для визначення схожості і відмінності кластерів є відстань між точками на діаграмі розсіювання. Цю схожість можна «виміряти», вона дорівнює відстані між точками на графіку.

Коли осей більше, ніж дві, відстань розраховується таким чином: сума квадратів різниці координат складається із стількох доданків, скільки осей (вимірів) присутньо в нашому просторі (рис. 7.29).

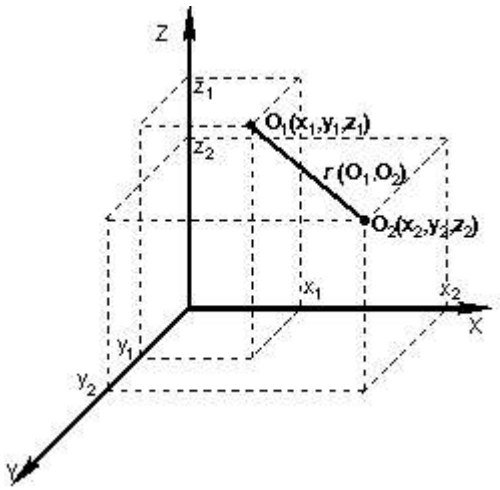


Рис. 7.29. Відстань між двома точками в просторі трьох вимірів.

У загальному випадку всі етапи кластерного аналізу взаємозв'язані, і рішення, прийняті на одному з них, визначають дії на подальших етапах. Основними етапами кластерного аналізу є:

1. Вибір метрики і методу стандартизації вихідних даних.
2. Визначення кількості кластерів (для ітеративного кластерного аналізу).
3. Визначення методу кластеризації (правила об'єднання або зв'язку). На думку багатьох фахівців, вибір методу кластеризації є вирішальним при визначенні форми і специфіки кластерів.
4. Аналіз результатів кластеризації. Цей етап передбачає вирішення таких питань: чи не є отримане розбиття на кластери випадковим; чи є розбиття надійним і стабільним на підвибірках даних; чи існує взаємозв'язок між результатами кластеризації і змінними, які не брали участь в процесі кластеризації; чи можна інтерпретувати отримані результати кластеризації.
5. Перевірка результатів кластеризації. Результати кластеризації також мають бути перевірені формальними і неформальними методами. Формальні методи залежать від того методу, який використовувався для кластеризації. Неформальні включають наступні процедури перевірки якості кластеризації:

- аналіз результатів кластеризації, отриманих на певних вибірках набору даних;
- крос-перевірка;
- проведення кластеризації при зміні порядку спостережень в наборі даних;
- проведення кластеризації при видаленні деяких спостережень;
- проведення кластеризації на невеликих вибірках.

Один з варіантів перевірки якості кластеризації - використання декількох методів і порівняння отриманих результатів. Відсутність подібності не означатиме некоректність результатів, але присутність схожих груп вважається ознакою якісної кластеризації.

Кластер має наступні математичні характеристики: центр, радіус, середньоквадратичне відхилення, розмір кластера. *Центр кластера* - це

середнє геометричне місце точок в просторі змінних. *Радіус кластера* - максимальна відстань точок від центру кластера.

Як було відмічено раніше, кластери можуть бути такими, що перекриваються. В цьому випадку неможливо за допомогою математичних процедур однозначно віднести об'єкт до одного з двох кластерів. Такі об'єкти називають *спірними*. Спірний об'єкт - це об'єкт, який по мірі схожості може бути віднесений до декількох кластерів.

Розмір кластера може бути визначений або по радіусу кластера, або по середньоквадратичному відхиленню об'єктів для цього кластера. Об'єкт відноситься до кластера, якщо відстань від об'єкта до центра кластера менше радіуса кластера. Якщо ця умова виконується для двох і більше кластерів, об'єкт є спірним.

Вибір масштабу в кластерному аналізі має велике значення. Розглянемо приклад. Уявимо собі, що дані ознаки X в наборі даних A на два порядки більше даних ознаки Y: значення змінної X знаходяться в діапазоні від 100 до 700, а значення змінної Y - в діапазоні від 0 до 1. Тоді, при розрахунку величини відстані між точками, що відображають положення об'єктів в просторі їх властивостей, змінна, що має великі значення, тобто змінна X, буде практично повністю домінувати над змінною з малими значеннями, тобто змінною Y. Таким чином із-за неоднорідності одиниць виміру ознак стає неможливо коректно розрахувати відстані між точками. Ця проблема вирішується за допомогою попередньої *стандартизації змінних*. Стандартизація (standardization) або нормування (normalization) приводить значення всіх перетворених змінних до єдиного діапазону значень шляхом вираження через відношення цих значень до деякої величини, що відображає певні властивості конкретної ознаки.

Існують різні способи нормування вихідних даних. Два найбільш поширених способа:

- ділення даних на середньоквадратичне відхилення відповідних змінних;
- обчислення Z вкладу або стандартизованого вкладу.

Разом із стандартизацією змінних, існує варіант додання кожній з них певного коефіцієнта важливості, або ваги, який би відображав значущість відповідної змінної. Як ваги можуть виступати експертні оцінки, отримані в ході опиту експертів - фахівців предметної області. Отримані множення нормованих змінних на відповідних ваги дозволяють отримувати відстані між точками в багатовимірному просторі з врахуванням неоднакової ваги змінних. В ході експериментів можливе порівняння результатів, отриманих з врахуванням експертних оцінок і без них, і вибір кращого з них.

Як і будь-які інші методи, методи кластерного аналізу мають певні слабкі сторони, тобто деякі складнощі, проблеми і обмеження. При проведенні кластерного аналізу слід враховувати, що результати кластеризації залежать від критеріїв розбиття сукупності вихідних даних. При пониженні розмірності даних можуть виникнути певні спотворення, за рахунок узагальнень можуть загубитися деякі індивідуальні характеристики об'єктів.

Існує ряд складнощів, які слід продумати перед проведенням кластеризації.

1. Складність вибору характеристик, на основі яких проводиться кластеризація. Необдуманий вибір приводить до неадекватного розбиття на кластери і, як наслідок, - до невірного рішення задачі.

2. Складність вибору методу кластеризації. Цей вибір вимагає непоганого знання методів і передумов їх використання. Аби перевірити ефективність конкретного методу певної предметної області, доцільно застосувати наступну процедуру: розглядають декілька апріорі різних між собою груп і перемішують їх представників між собою випадковим чином. Далі проводиться кластеризація для відновлення вхідного розбиття на кластери. Доля збігів об'єктів у виявлених і вхідних групах є показником ефективності роботи методу.

3. Проблема вибору числа кластерів. Якщо немає жодних відомостей відносно можливого числа кластерів, необхідно провести ряд експериментів і, в результаті перебору різного числа кластерів, вибрати оптимальне їх число.

4. Проблема інтерпретації результатів кластеризації. Форма кластерів в більшості випадків визначається вибором методу об'єднання. Проте слід враховувати, що конкретні методи прагнуть створювати кластери певних форм, навіть якщо в досліджуваному наборі даних кластерів насправді немає.

Перед проведенням кластеризації у аналітика може виникнути питання, якій групі методів кластерного аналізу віддати перевагу. Методи кластерного аналізу у загальному випадку можна розділити на дві групи: ієрархічні та неієрархічні. Вибираючи між ними, необхідно враховувати наступні їх особливості.

Неієрархічні методи виявляють вищу стійкість по відношенню до шумів і викидів, некоректного вибору метрики, включення незначимих змінних в набір, що бере участь в кластеризації. Ціною, яку доводиться платити за ці достоїнства методу, є слово «апріорі». Аналітик повинен заздалегідь визначити кількість кластерів, кількість ітерацій або правило зупинки, а також деякі інші параметри кластеризації.

Якщо немає припущень відносно числа кластерів, рекомендують використовувати ієрархічні алгоритми кластерного аналізу. Проте якщо об'єм вибірки не дозволяє це зробити, можлива дорога - проведення ряду експериментів з різною кількістю кластерів, наприклад, почати розбиття сукупності даних з двох груп і, поступово збільшуючи їх кількість, порівнювати результати. За рахунок такого «варіювання» результатів досягається чимала гнучкість кластеризації.

Ієрархічні методи, на відміну від неієрархічних, відмовляються від визначення числа кластерів, а будують повне дерево вкладених кластерів. Складнощі ієрархічних методів кластеризації: обмеження об'єму набору даних; вибір міри близькості; негнучкість отриманих класифікацій. Перевага цієї групи методів порівняно з неієрархічними методами - їх наочність і можливість отримати детальне уявлення про структуру даних. При використанні ієрархічних методів існує можливість досить легко ідентифікувати викиди в наборі даних і, в результаті, підвищити якість даних. Ця процедура лежить в

основі двокрокового алгоритму кластеризації. Такий набір даних надалі може бути використаний для проведення неієрархічної кластеризації.

Існує ще один аспект, який полягає в можливості кластеризації всієї сукупності даних або ж її вибірки. Названий аспект істотний для обох груп методів, проте він критичніший для ієрархічних методів. Ієрархічні методи не можуть працювати з великими наборами даних, а використання деякої вибірки, тобто частини даних, могло б дозволити застосовувати ці методи.

Результати кластеризації можуть не мати достатнього статистичного обґрунтування. З іншого боку, при вирішенні задач кластеризації допустима нестатистична інтерпретація отриманих результатів, а також чимала різноманітність варіантів поняття кластера. Така нестатистична інтерпретація дає можливість аналітикові отримати результати кластеризації, які задовольняють його, що при використанні інших методів часто буває скрутним. Розглянемо ієрархічні і неієрархічні методи більш детальніше.

Ієрархічні методи кластерного аналізу. При ієрархічній кластеризації виконується послідовне об'єднання менших кластерів у великі або розділення великих кластерів на менші.

Агломеративні методи AGNES (Agglomerative Nesting). Ця група методів характеризується послідовним об'єднанням елементів і відповідним зменшенням числа кластерів. На початку роботи алгоритму всі об'єкти є окремими кластерами. На першому кроці найбільш схожі об'єкти об'єднуються в кластер. На подальших кроках об'єднання продовжується до тих пір, поки всі об'єкти не складатимуть один кластер.

Single Link, Complete Link, Group Average. Одні із перших алгоритмів кластеризації даних. Особливістю цих методів, є те, що вони розбивають об'єкти на кластери шляхом розбиття їх на ієрархічні групи. Основна суть цих методів полягає у виконанні наступних кроків:

- обчислення значень близькості між елементами і здобуття матриці близькості;
- визначення кожного елементу в свій окремий кластер;

- злиття в один кластер найбільш близьких пар елементів;
- оновлення матриці близькості шляхом видалення колонок і рядків для кластерів, які злилися з іншими і подальшого перерахунку матриці;
- перехід на крок 3 до тих пір, поки не спрацює зупинний критерій.

Поняттям, протилежним до поняття *відстані* між об'єктами X_i та X_j, є поняття *близькості* (схожість) між X_i і X_j. Точніше, міра близькості між об'єктами X_i та X_j - це дійсна функція $\mu(X_i, X_j) = \mu_{ij}$ з властивостями:

$$0 \le \mu(X_i, X_j) < 1, \text{ якщо } X_i \ne X_j,$$

$$\mu(X_i, X_j) = 1,\ \mu(X_i, X_j) = \mu(X_j, X_i).$$

Пари значень мір близькості можна об'єднати в матрицю близькості

$$\mu = \begin{vmatrix} 1 & \mu_{12} & \dots & \mu_{1n} \\ \mu_{21} & 1 & \dots & \mu_{2n} \\ \dots & \dots & \dots & \dots \\ \mu_{n1} & \mu_{n2} & \dots & 1 \end{vmatrix},\ \mu_{ii} = 0 \text{ для } i = 1,2,...,n.$$

Величину μ_{ij} називають *коефіцієнтом близькості*. Прикладом лінійною близькості є коефіцієнт кореляції.

Вказані алгоритми відрізняються між собою в 4-му кроці. І саме завдяки способам оновлення матриці близькості різні алгоритми мають різну точність. Перевірка точності алгоритмів була проведена на спеціальних тестових наборах і показала, що алгоритм Single Link має найменшу точність, а останні два - приблизно однакову, але більш високу, чим Single Link. В якості зупинного критерію вибирається максимальна кількість об’єктів в кластері (рис. 7.30). Швидкість роботи алгоритмів Single Link і Group Average - $O(n^2)$, а Complete Link - $O(n^3)$, де n - кількість елементів.

Перевагами методів є те, що алгоритми не потребують навчання та використовують матрицю близькості між елементами. Недоліками цих методів є: необхідно задавати поріг - максимальну кількість елементів в кластері; для здобуття добрих результатів кластеризації значення близькості між парами

елементів повинні приходити в певному порядку, тобто робота алгоритму не детермінована; кластери не перетинаються.

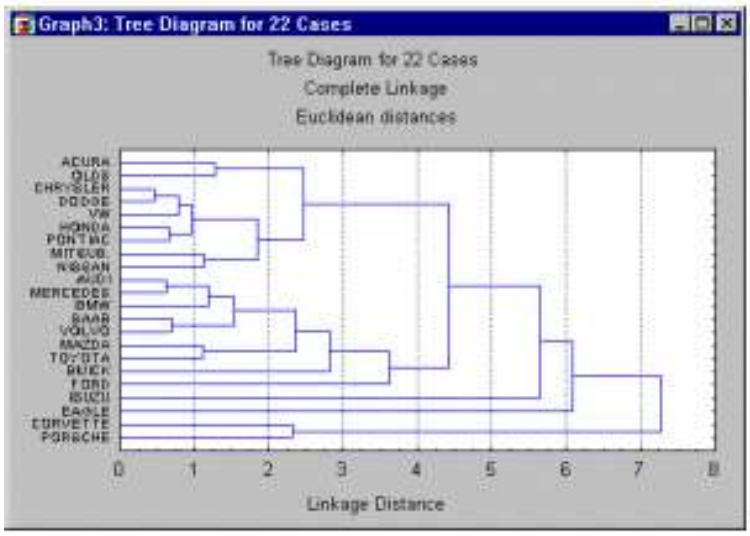


Рис. 7.30. Результати роботи алгоритму Complete Link.

Алгоритм CURE (Clustering Using REpresentatives). Виконує ієрархічну кластеризацію з використанням набору визначальних точок для визначення об'єкту в кластер. Призначення: кластеризація дуже великих наборів числових даних. Обмеження: ефективний для даних низької розмірності, працює лише на числових даних. Достоїнства: виконує кластеризацію на високому рівні навіть за наявності викидів, виділяє кластери складної форми і різних розмірів, володіє лінійно залежними вимогами до місця зберігання даних і часову складність для даних високої розмірності. Недоліки: є необхідність в задані порогових значень і кількості кластерів. Робота алгоритму складається з наступних кроків:

Крок 1: Побудова дерева кластерів, яке складається з кожного рядка вхідного набору даних.

Крок 2: Формування «купи» в оперативній пам'яті, розрахунок відстані до найближчого кластера (рядка даних) для кожного кластера. При формуванні «купи» кластери сортуються за збільшенням дистанції від кластера до найближчого кластера. Відстань між кластерами визначається по двох найближчих елементах з сусідніх кластерів. Для визначення відстані між

кластерами використовуються «манхэттенська», «Евклідова» метрики або схожі на них функції.

Крок 3: Злиття ближніх кластерів в один кластер. Новий кластер отримує всі точки вхідних даних, які опиняються в ньому. Виконується розрахунок відстані до інших кластерів для новоутвореного кластера. Для розрахунку відстані кластери діляться на дві групи: перша група – кластери, в яких найближчими кластерами вважаються кластери, що входять в новоутворений кластер, останні кластери – друга група. При цьому для кластерів з першої групи, якщо відстань до новоутвореного кластера менше ніж до попереднього найближчого кластера, то найближчий кластер міняється на новоутворений кластер. Інакше шукається новий найближчий кластер, але при цьому не беруться кластери, відстані до яких більше, ніж до новоутвореного кластера. Для кластерів другої групи виконується наступне: якщо відстань до новоутвореного кластера ближча, ніж попередній найближчий кластер, то найближчий кластер міняється. Інакше нічого не відбувається.

Крок 4: Перехід на крок 3, якщо не отримана необхідна кількість кластерів.

Дивізімні методи DIANA (Divisive Analysis). Ці методи є логічною протилежністю агломеративним методам. На початку роботи алгоритму всі об'єкти належать одному кластеру, який на подальших кроках ділиться на менші кластери, в результаті утворюється послідовність розщеплюючих груп.

Алгоритм BIRCH (Balanced Iterative Reducing and Clustering using Hierarchies). У цьому алгоритмі передбачений двох етапний процес кластеризації. Призначення: кластеризація дуже великих наборів числових даних. Обмеження: робота з лише числовими даними. Достоїнства: двоступінчата кластеризація, кластеризація великих об'ємів даних, працює на обмеженому об'ємі пам'яті, є локальним алгоритмом, може працювати при одному скануванні вхідного набору даних, використовує той факт, що дані неоднаково розподілені по простору, і обробляє області з великою щільністю як єдиний кластер. Недоліки: робота з лише числовими даними, добре виділяє

лише кластери сферичної форми, є необхідність в задані порогових значень. Робота алгоритму здійснюється таким чином:

Фаза 1: Завантаження даних в пам'ять. Побудова початкового кластерного дерева за даними в пам'яті. Кластерне дерево – це зважено збалансоване дерево з двома параметрами: B – коефіцієнт розгалуження, T – порогова величина. Кожен не листовий вузол дерева має не більше ніж B входжень вузлів (рис. 7.31).

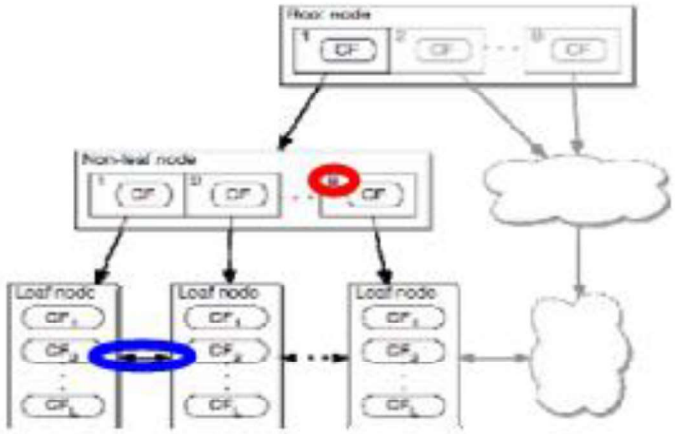


Рис. 7.31. Побудова кластерного дерева.

Кожен листовий вузол має посилання на два сусідні вузли. Кластер, що складається з елементів листового вузла, повинен задовольняти наступній умові: діаметр або радіус отриманого кластера має бути не більше порогової величини T.

Фаза 2: Стискування (ущільнення) даних. Стискування даних до прийнятних розмірів за допомогою перестроювання і зменшення кластерного дерева із збільшенням порогової величини T.

Фаза 3: Глобальна кластеризація. Застосовується вибраний алгоритм кластеризації на листових компонентах кластерного дерева.

Фаза 4: Поліпшення кластерів. Використовує центри тяжесті кластерів, отримані у фазі 3, як основи. Перерозподіляє дані між «близькими» кластерами. Дана фаза гарантує попадання однакових даних в один кластер.

Алгоритм MST (Algorithm based on Minimum Spanning Trees). Призначення: кластеризація великих наборів довільних даних. Достоїнства: виділяє кластери довільної форми, вибирає з декількох оптимальних рішень найоптимальніше. Опис алгоритму:

Крок 1: Побудова мінімального остовного дерева.

Алгоритм Борувки:

1. Для кожної вершини графа знаходимо ребро з мінімальною вагою.
2. Додаємо знайдені ребра до остовного дерева, за умови їх безпеки.
3. Знаходимо і додаємо безпечні ребра для незв'язаних вершин до остовного дерева.

Алгоритм Крускала:

1. Обхід ребер за збільшенням їх ваги.
2. За умови безпеки ребра додаємо його до остовного дерева.

Алгоритм Пріма:

1. Вибір кореневої вершини.
2. Починаючи з кореня додаємо безпечні ребра до остовного дерева.

Крок 2: Розділення на кластери. Дуги з найбільшими вагами розділяють кластери.

Алгоритм Форель. Алгоритм є прикладом евристичного дивізімного алгоритму класифікації. В основі роботи алгоритму Форель лежить використання гіпотези компактності: близьким в змістовному сенсі об'єктам в геометричному просторі ознак відповідають відособлена множина точок, так звані «згустки». Мета роботи алгоритму Форель полягає в тому, щоб знайти таке розбиття множини об'єктів, аби величина відстані між ними була мінімальною.

Робота алгоритму полягає в переміщенні гіперсфери певного радіусу в геометричному просторі до здобуття стійкого центру тяжіння спостережень, що попали в цю гіперсферу. До початку роботи алгоритму ознаки об'єктів нормуються так, щоб їх значення знаходилися між нулем і одиницею. Процедура алгоритму Форель є такою, що сходиться за кінцеве число кроків в

просторі Евкліда будь-якої розмірності при довільному розташуванні точок і будь-якому виборі гіперсфери (рис. 7.32).

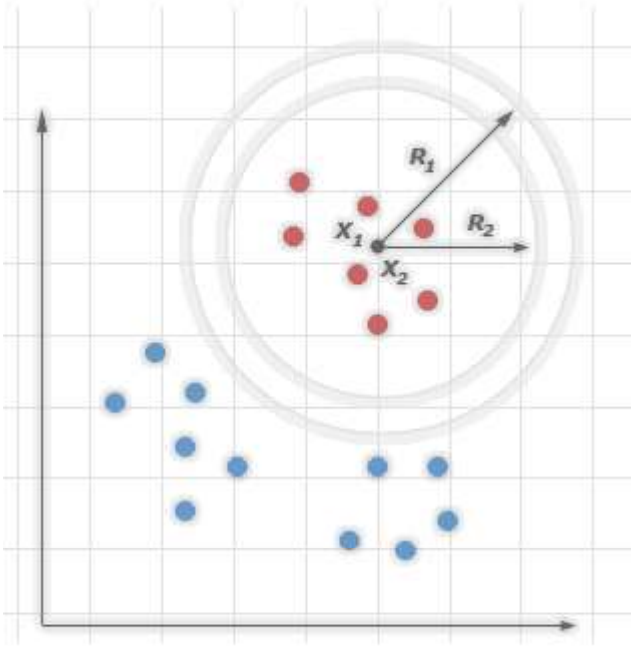


Рис. 7.32. Пошук кластерів алгоритмом Форель.

Алгоритм Форель 2 є модифікацією алгоритму Форель і застосовується в тих випадках, коли необхідно отримати спочатку задану кількість кластерів. Радіус сфери у міру потреби може змінюватися на задану величину, яка від ітерації до ітерації зменшуватиметься.

Принцип роботи описаних вище груп методів у вигляді дендрограмми показаний на рис. 7.33.

Ієрархічні методи кластеризації розрізняються правилами побудови кластерів. В якості правила виступають критерії, які використовуються при рішенні питання про «схожість» об'єктів при їх об'єднанні в групу (агломеративні методи) або розділення на групи (дивізімні методи). Ці методи кластерного аналізу застосовуються при невеликих об'ємах наборів даних..

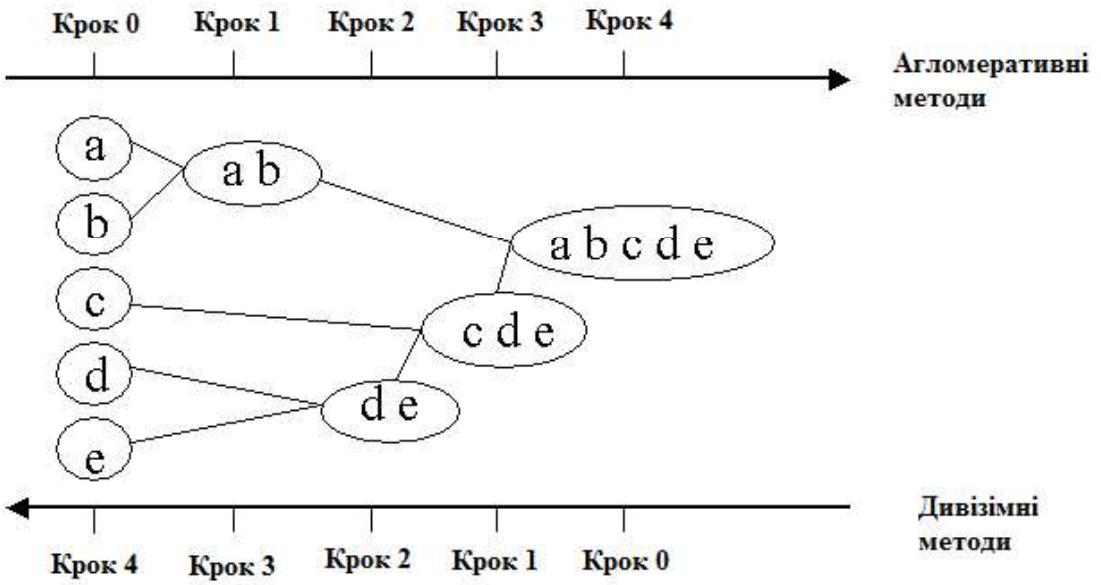


Рис. 7.33. Дендрограмма агломеративних і дивізімних методів.

Неієрархічні методи кластерного аналізу. При великій кількості спостережень ієрархічні методи кластерного аналізу не придатні. У таких випадках використовують неієрархічні методи, засновані на розділенні, які є ітеративними методами дроблення вхідної сукупності. В процесі ділення нові кластери формуються до тих пір, поки не буде виконано правило зупинки. Така неієрархічна кластеризація полягає в розділенні набору даних на певну кількість окремих кластерів. Існує два підходи. Перший полягає у визначенні кордонів кластерів як найбільш щільних ділянок в багатовимірному просторі даних, тобто визначення кластера там, де є велике «згущення точок». Другий підхід полягає в мінімізації міри відмінності об'єктів.

Ітераційні алгоритми. Такі алгоритми засновані на оптимізації деякої цільової функції, що визначає оптимальне в певному значенні розбиття множини об'єктів на кластери. Вони носять ітеративний характер, і на кожній ітерації потрібно розраховувати матрицю відстаней між об'єктами. При великому числі об'єктів це неефективно і вимагає серйозних обчислювальних ресурсів.

Алгоритм k-середніх (k-means). Найбільш поширений серед неієрархічних методів алгоритм k-середніх, також названий швидким

кластерним аналізом. На відміну від ієрархічних методів, які не вимагають попередніх припущень відносно числа кластерів, для можливості використання цього методу необхідно мати гіпотезу про найбільш вірогідну кількість кластерів. Алгоритм k-середніх будує k кластерів, розташованих на можливо великих відстанях один від одного. Основний тип задач, які вирішує алгоритм k-середніх, - наявність припущень (гіпотез) відносно числа кластерів, при цьому вони мають бути різні настільки, наскільки це можливо. Вибір числа k може базуватися на результатах попередніх досліджень, теоретичних міркуваннях або інтуїції. Загальна ідея алгоритму полягає в наступному: задане фіксоване число k кластерів спостереження зіставляються кластерам так, що середні в кластері (для всіх змінних) максимально можливо відрізняються один від одного. Робота алгоритму полягає в наступному (рис. 7.34):

1. Первинний розподіл об'єктів по кластерах. Вибирається число k, і на першому кроці ці точки вважаються «центрами» кластерів. Кожному кластеру відповідає один центр.

2. Ітеративний процес. Обчислюються центри кластерів, якими потім і далі вважаються покоординатні середні кластерів. Об'єкти знову перерозподіляються. Процес обчислення центрів і перерозподілу об'єктів продовжується до тих пір, поки не виконана одна з умов:

- кластерні центри стабілізувалися, тобто всі спостереження належать кластеру, якому належали до поточної ітерації;
- число ітерацій дорівнює максимальному числу ітерацій.

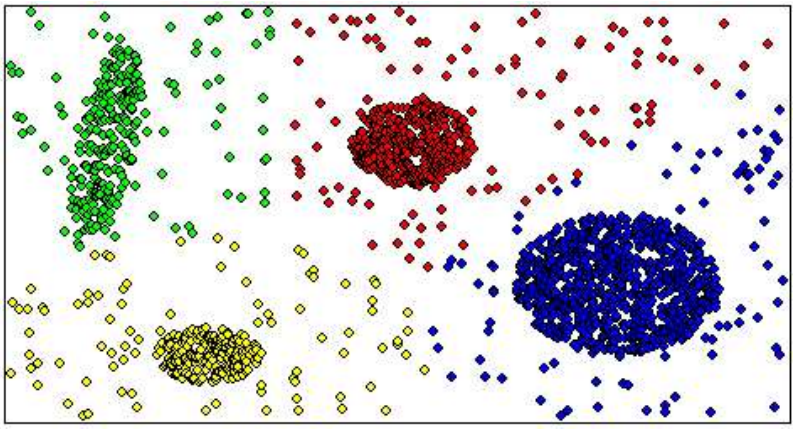


Рис. 7.34. Результат кластеризації алгоритмом k-means.

Після здобуття результатів кластерного аналізу методом k-середніх слід перевірити правильність кластеризації (тобто оцінити, наскільки кластери відрізняються один від одного). Для цього розраховуються середні значення для кожного кластера. При хорошій кластеризації мають бути отримані середні, що сильно відрізняються, для всіх вимірів або хоч би більшої їх частини.

Перевагами алгоритму k-середніх є: простота та швидкість використання, зрозумілість і прозорість алгоритму. Недоліки алгоритму k-середніх:

- алгоритм дуже чутливий до викидів, які можуть спотворювати середнє. Можливим вирішенням цієї проблеми є використання модифікації алгоритму - алгоритм k-медіани;
- алгоритм може повільно працювати на великих базах даних. Можливим вирішенням даної проблеми є використання вибірки даних.

З метою підвищення ефективності роботи алгоритму розроблені цікаві його розширення для роботи з категорійними атрибутами (k-modes) і змішаними атрибутами (k-prototypes). Наприклад, в к-prototypes розрахунок відстаней між об'єктами здійснюється по-різному залежно від типа атрибуту.

Алгоритм PAM (partitioning around Medoids). PAM є модифікацією алгоритму k-середніх алгоритмом k-медіани (k-medoids).Алгоритм менш чутливий до шумів і викидів даних, чим алгоритм k-means, оскільки медіана менше схильна до впливів викидів. PAM ефективний для невеликих баз даних, але його не слід використовувати для великих наборів даних.

Алгоритм EM (Expectation - Maximization). В основі ідеї EM алгоритму лежить припущення, що досліджувана множина даних може бути змодельована за допомогою лінійної комбінації багатовимірних нормальних розподілів, а метою є оцінка параметрів розподілів, які максимізували логарифмічну функцію правдоподібності, використовувану як міра якості моделі. Іншими словами, передбачається, що дані в кожному кластері підкоряються певному закону розподілу, а саме, нормальному розподілу. З врахуванням цього припущення можна визначити параметри - математичне сподівання і

дисперсію, які відповідають закону розподілу елементів в кластері, щонайкраще «відповідному» до спостережуваних даних. Таким чином, ми передбачаємо, що будь-яке спостереження належить до всіх кластерів, але з різною ймовірністю. Тоді задача полягатиме в «підгонці» розподілів суміші до даних, а потім у визначенні ймовірності приналежності спостереження до кожного кластера. Вочевидь, що спостереження має бути віднесене до того кластера, для якого дана вірогідність вища.

Алгоритм EM заснований на обчисленні відстаней. Він може розглядатися як узагальнення кластеризації на основі аналізу суміші імовірнісних розподілів. В процесі роботи алгоритму відбувається ітеративне поліпшення рішення, а зупинка здійснюється в мить, коли досягається необхідний рівень точності моделі. Мірою в даному випадку є статистична величина, що монотонно збільшується, названа логарифмічною правдоподібністю. Метою алгоритму є оцінка середніх значень, коваріацій і вагів суміші для функції розподілу ймовірностей (рис. 7.35).

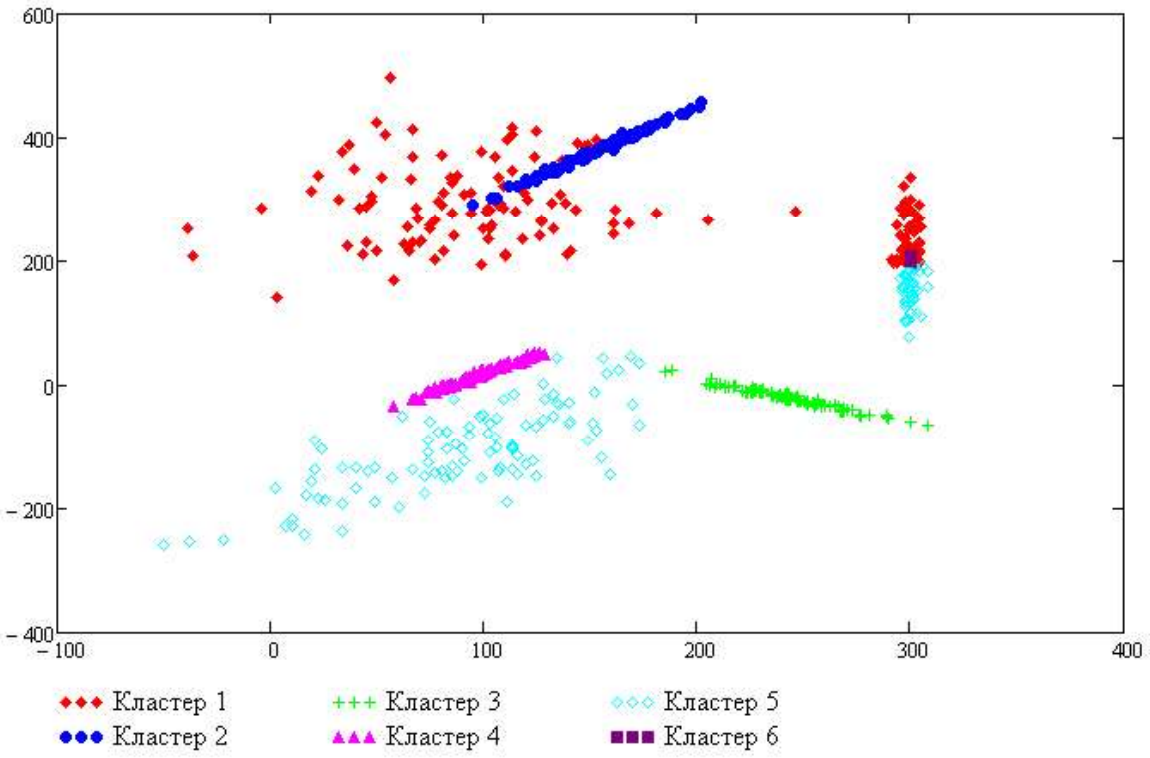


Рис. 7.35. Результати кластеризації алгоритмом EM.

Серед переваг EM алгоритму можна виділити наступні: потужна статистична основа, лінійне збільшення складності при зростанні об'єму даних, стійкість до шумів і пропусків в даних, можливість побудови бажаного числа кластерів. Проте алгоритм має і ряд недоліків. По-перше, припущення про нормальність всіх вимірів даних не завжди виконується. По-друге, при невдалій ініціалізації збіжність алгоритму може виявитися повільною. Окрім цього, алгоритм може зупинитися в локальному мінімумі і дати квазіоптимальне рішення.

Кластеризація категорійних даних: алгоритм CLOPE. Категорійні дані зустрічаються в будь-яких областях: виробництво, комерція, маркетинг, медицина. Вони включають і так звані транзакційні дані: чеки в супермаркетах, логи відвідин веб-ресурсів. Взагалі під категорійними даними розуміють якісні характеристики об'єктів, виміряні в шкалі найменувань.

Застосовувати для кластеризації об'єктів з категорійними ознаками традиційні алгоритми неефективно, а часто – неможливо. Основні труднощі пов'язані з високою розмірністю і гігантським об'ємом, якими часто характеризуються такі бази даних. На сьогоднішній день запропоновано понад десяток методів для роботи з категорійними даними. Одним з ефективних вважається алгоритм LargeItem, який заснований на оптимізації деякого глобального критерію. Цей глобальний критерій використовує параметр підтримки. Взагалі, обчислення глобального критерію робить алгоритм кластеризації у багато разів швидше, ніж при використанні локального критерію при парному порівнянні об'єктів, тому «глобалізація» оціночної функції – один з шляхів здобуття масштабованих алгоритмів.

Алгоритм CLOPE запропонований в 2002 році групою китайських вчених. При цьому він забезпечує вищу продуктивність і кращу якість кластеризації порівняно з алгоритмом LargeItem і багатьма ієрархічними алгоритмами. Призначення: кластеризація величезних наборів категорійних даних. Переваги: висока масштабованість і швидкість роботи, якість кластеризації, що досягається використанням глобального критерію оптимізації

на основі максимізації градієнта висоти гістограми кластера. Під час роботи алгоритм зберігає в пам'яті невелику кількість інформації по кожному кластеру і вимагає мінімальне число сканувань набору даних. CLOPE автоматично підбирає кількість кластерів, причому це регулюється одним єдиним параметром – коефіцієнтом відштовхування.

Алгоритм функціонує на основі слідуючих процедур. Хай є база транзакцій D, що складається з множини транзакцій $\{t_1,t_2,...,t_n\}$. Кожна транзакція є набір об'єктів $\{i_1,...,i_m\}$. Множина кластерів $\{C_1,...,C_k\}$ є розбиття множини $\{t_1,t_2,...,t_n\}$, таке, що $C_1 \cup ... \cup C_k = \{t_1,t_2,...,t_n\}$ і $C_i \cap C_j = \varnothing$, $\forall i \geq 1$, $k \geq j$. Кожен елемент C_i називається кластером, а n,m,k – кількість транзакцій, кількість об'єктів в базі транзакцій і число кластерів відповідно. Кожен кластер C_i має наступні характеристики:

- $D(C)$ – множина унікальних об'єктів;
- $Occ(i,C)$ – кількість входжень (частота) об'єкту i в кластер C_i;

$$S(C) = \sum_{i \in D(C)} Occ(i,C) = \sum_{t_i \in C} |t_i|,$$

$$W(C) = |D(C)|,\ H(C) = S(C)/W(C)$$

- функція вартості

$$\Pr ofit(C) = \frac{\sum_{i=1}^{k} G(C_i) * |C_i|}{\sum_{i=1}^{k} |C_i|} = \frac{\sum_{i=1}^{k} \frac{S(C_i)}{W(C_i)^r} * |C_i|}{\sum_{i=1}^{k} |C_i|},$$

де $|C_i|$ – кількість об'єктів в i кластері, k – кількість кластерів, r – коефіцієнт відштовхування. За допомогою параметра r регулюється рівень схожості транзакцій усередині кластера, і, як наслідок, фінальна кількість кластерів. Цей коефіцієнт підбирається користувачем. Чим більше r, тим нижче рівень схожості і тим більше кластерів згенерується.

Формальна постановка задачі кластеризації алгоритмом CLOPE виглядає таким чином: для заданих D та r знайти розбиття C: $\Pr ofit(C,r) \to \max$.

Нові алгоритми кластерного аналізу. Методи, які ми розглянули, є «класикою» кластерного аналізу. До останнього часу основним критерієм, по якому оцінювався алгоритм кластеризації, була якість кластеризації: вважалося, аби весь набір даних уміщався в оперативній пам'яті. Проте зараз, у зв'язку з появою надвеликих баз даних, з'явилися нові вимоги, яким повинен задовольняти алгоритм кластеризації. Основне з них - це масштабованість алгоритму. Відзначимо також інші властивості, яким повинен задовольняти алгоритм кластеризації: незалежність результатів від порядку вхідних даних; незалежність параметрів алгоритму від вхідних даних. Останнім часом ведуться активні розробки нових алгоритмів кластеризації, здатних обробляти надвеликі бази даних. У них основна увага приділяється масштабованості. До таких алгоритмів відноситься узагальнене представлення кластерів (summarized cluster representation), а також вибірка і використання структур даних, підтримуваних нижче лежачих СУБД.

Алгоритми нечіткої кластеризації - алгоритм Fuzzy C-means. Призначення: кластеризація великих наборів числових даних. Достоїнства: нечіткість при визначенні об'єкту в кластер дозволяє визначати об'єкти, які знаходяться на кордоні, в кластери. Недоліки: обчислювальна складність, задання кількості кластерів, виникає невизначеність з об'єктами, які віддалені від центрів всіх кластерів.

Принцип роботи алгоритму полягає в наступному. Хай нечіткі кластери задаються матрицею розбиття $F=[\mu_{ki}]$, $\mu_{ki}\in[0,1]$, $k=\overline{1,M}$, $i=\overline{1,c}$, де μ_{ki} - міра приналежності об'єкту k до кластера i, c – кількість кластерів, M – кількість елементів. При цьому

$$\sum_{i=1}^{c}\mu_{ki}=1,\ k=\overline{1,M},\ 0<\sum_{k=1}^{M}\mu_{ki}<M,\ i=\overline{1,c}.$$

Етап 1. Встановити параметри алгоритму: c - кількість кластерів; m - експоненціальна вага, що визначає нечіткість, розмазаність кластерів ($m\in[1,\infty)$), ε - параметр зупинки алгоритму.

Етап 2. Генерація випадковим чином матриці нечіткого розбиття з врахуванням вказаних умов.

Етап 3. Розрахунок центрів кластерів $V_i = \dfrac{\sum_{k=1}^{M} \mu_{ki}^{\ m} * |X_k|}{\sum_{k=1}^{M} \mu_{ki}^{\ m}}$, $i = \overline{1,c}$.

Етап 4. Розрахунок відстані між об'єктами X і центрами кластерів

$$D_{ki} = \sqrt{\|X_k - V_i\|^2}\ ,\ k = \overline{1,M}\ ,\ i = \overline{1,c}\ .$$

Етап 5. Перерахунок елементів матриці розбиття з врахуванням наступних умов:

якщо $D_{ki} > 0$: $\mu_{kj} = \dfrac{1}{\left(D_{jk}^2 * \sum_{j=1}^{c} \dfrac{1}{D_{jk}^2} \right)^{1/(m-1)}}$, $j = \overline{1,c}$

якщо $D_{ki} = 0 : \mu_{kj} = \begin{cases} 1, j = i \\ 0, j \neq i \end{cases}$, $j = \overline{1,c}$.

Етап 6. Перевірити умову $\|F - F^*\| < \varepsilon$, де F^* - матриця нечіткого розбиття на попередній ітерації алгоритму. Якщо «Так», то перехід до етапу 7, інакше до етапу 3.

Етап 7. Кінець роботи алгоритму.

Алгоритм WaveCluster. WaveCluster є алгоритмом кластеризації на основі хвилевих перетворень. На початку роботи алгоритму дані узагальнюються шляхом накладання на простір даних багатовимірних грат. На подальших кроках алгоритму аналізуються не окремі точки, а узагальнені характеристики точок, що попали в одну чарунку грат. В результаті такого узагальнення необхідна інформація уміщається в оперативній пам'яті. На подальших кроках для визначення кластерів алгоритм застосовує хвилеве перетворення до узагальнених даних. Головні особливості WaveCluster: складність реалізації, алгоритм може виявляти кластери довільних форм, алгоритм не чутливий до шумів, алгоритм застосовний лише до даних низької розмірності.

Алгоритм CLARA (Clustering LARge Applications). Алгоритм CLARA був розроблений Kaufmann і Rousseeuw в 1990 році для кластеризації даних у великих базах даних. Даний алгоритм виконується в статистичних аналітичних пакетах, наприклад, таких як S+.

Викладемо коротко суть алгоритму. Алгоритм CLARA витягує множину зразків з бази даних. Кластеризація застосовується до кожного із зразків, на виході алгоритму пропонується краща кластеризація. Для великих баз даних цей алгоритм ефективніший, ніж алгоритм PAM. Ефективність алгоритму залежить від вибраного як зразок набору даних. Хороша кластеризація на вибраному наборі може не дати хорошу кластеризацію на всій множині даних.

Алгоритми Clarans, CURE, DBScan. Алгоритм Clarans (Clustering Large Applications based upon RANdomized Search) формулює задачу кластеризації як випадковий пошук в графі. В результаті роботи цього алгоритму сукупність вузлів графа є розбиттям множини даних на число кластерів, визначеним користувачем. «Якість» отриманих кластерів визначається за допомогою критеріальної функції. Алгоритм Clarans сортує все можливе розбиття множини даних у пошуках прийнятного рішення. Пошук рішення зупиняється в тому вузлі, де досягається мінімум серед зумовленого числа локальних мінімумів.

Серед нових масштабованих алгоритмів також можна відзначити алгоритм CURE - алгоритм ієрархічної кластеризації, і алгоритм DBScan, де поняття кластера формулюється з використанням концепції щільності (density). Основним недоліком алгоритмів Clarans, CURE, DBScan є та обставина, що вони вимагають задання деяких порогів щільності точок, а це не завжди прийнятно. Ці обмеження обумовлені тим, що описані алгоритми орієнтовані на надвеликі бази даних і не можуть користуватися великими обчислювальними ресурсами.

Над масштабованими методами зараз активно працюють багато дослідників, основне завдання яких - здолати недоліки алгоритмів, що існують на сьогоднішній день.

7.4. Програмне забезпечення задач кластеризації

Одним з основних підходів в «виявленні знань в даних» (Data Mining) є кластеризація. Кластерний аналіз дозволяє відкрити в даних раніше невідомі закономірності, які практично неможливо досліджувати іншими способами і представити їх в зручній для користувача формі. Методи кластерного аналізу використовуються як самостійні інструменти досліджень, так і у складі інших засобів Data Mining. До теперішнього часу розроблена велика кількість програмних продуктів, що застосовуються до даних різного типа. Розглянемо основні з них [17, 25, 72, 76, 79].

Система PolyAnalyst. В програмному комплексі реалізовано два модулі, які відповідають за проведення кластерного аналізу: Find Dependecies (FD) - N-мірний аналіз розподілів та Find Clusters (FC) - N-мірний кластеризатор.

Алгоритм Find Dependecies виявляє у вхідній таблиці групи записів, для яких характерна наявність функціонального зв'язку між цільовою змінною і незалежними змінними, оцінює міру (силу) цієї залежності в термінах стандартної помилки, визначає набір найбільш впливових чинників, відсіває точки, що відскочили. Цільова змінна для FD має бути числового типа, тоді як незалежні змінні можуть бути і числовими і категоріями і логічними (рис. 7.36).

Рис. 7.36. Кластеризація в модулі Find Dependecies.

Алгоритм працює дуже швидко і здатний обробляти великі об'єми даних. Його можна використовувати як препроцесор для інших алгоритмів, оскільки він зменшує простір пошуку, а також як фільтр точок, що відскочили, або в зворотній постановці, як детектор виключень. FD створює правило табличного вигляду, проте як і всі правила PolyAnalyst воно може бути обчислене для будь-якого запису таблиці.

Модуль Find Clusters застосовується тоді, коли треба виділити в деякій множині даних компактні типові підгрупи (кластери), що складаються з близьких по своїх характеристиках записів. Причому заздалегідь може бути невідомо які змінні потрібно використовувати для такого розбиття. Алгоритм FC сам визначає набір змінних, для яких розбиття найзначиміше. Результатом роботи алгоритму є опис областей (діапазонів значень змінних), що характеризують кожен виявлений кластер і розбиття досліджуваної таблиці на підмножини, відповідні кластерам. Якщо дані є досить однорідними по всіх своїх змінних і не містять «згущувань» точок в якихось областях, цей метод не дасть результатів (рис. 7.37).

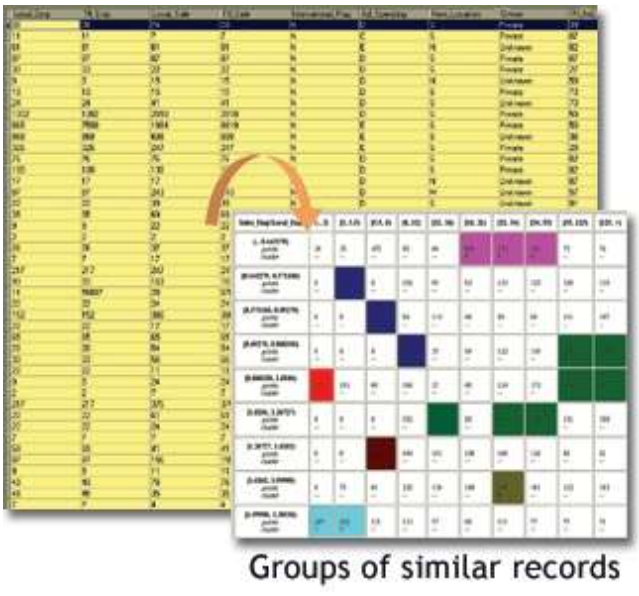


Рис. 7.37. Кластеризація в модуліFind Clusters.

Треба відзначити, що мінімальне число кластерів, що виявляються, рівне двом - згущування точок лише в одному місці в даному алгоритмі не розглядається як кластер. Крім того, цей метод пред'являє вимоги до наявності

достатньої кількості записів в досліджуваній таблиці, а саме, мінімальна кількість записів в таблиці, в якій може бути виявлено N кластерів, рівне $(2N-1)4$.

Система «Багатовимірні розвідувальні технології аналізу STATISTICA». У модулі «Кластерний аналіз» реалізований повний набір методів кластерного аналізу даних, включаючи методи k-середніх, ієрархічної кластеризації і двовходового об'єднання. Дані можуть поступати як у ісходному вигляді, так і у вигляді матриці відстаней між об'єктами. Спостереження і змінні можна кластеризувати, використовуючи різні міри відстані (евклідову, квадрат евклідова, міських кварталів (манхэттенське), Чебишева, степенне, відсоток незгоди і коефіцієнта кореляції Пірсона), а також різні правила об'єднання кластерів. Матриці відстаней можна зберігати для подальшого аналізу в інших модулях системи STATISTICA (рис. 7.38).

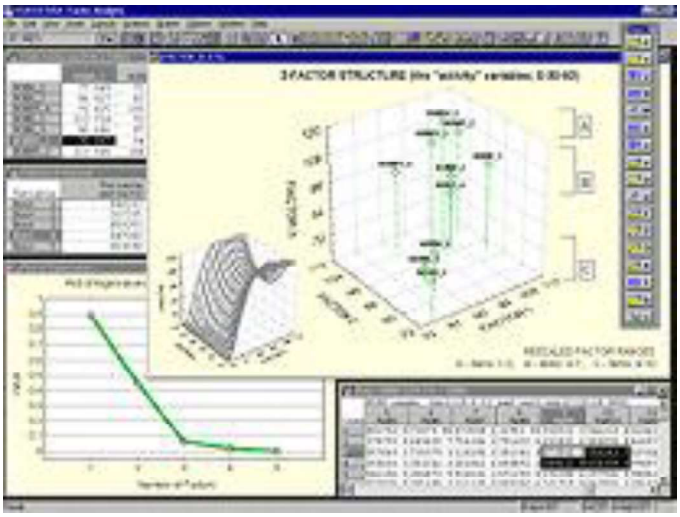


Рис. 7.38. Кластерний аналіз в пакеті STATISTICA.

При проведенні кластерного аналізу методом k-середніх користувач має повний контроль над початковим розташуванням центрів кластерів. Можуть бути виконані надзвичайно великі плани аналізу: так наприклад, при ієрархічному аналізі можна працювати з матрицею з 90 тис. відстаней. Окрім стандартних результатів кластерного аналізу, в модулі доступний також всілякий набір описових статистик і розширених діагностичних методів (повна схема об'єднання з пороговими рівнями при ієрархічній кластеризації, таблиця

дисперсійного аналізу при кластеризації методом k-середніх). Інформація про приналежність об'єктів до кластерів може бути додана до файлу даних і використовуватися в подальшому аналізі. Графічні можливості модуля «Кластерний аналіз» включають дендрограмми, двовходові діаграми об'єднань, графічне представлення схеми об'єднання, діаграму середніх при кластеризації по методу k-середніх і багато що інше.

Розглянемо процедуру рішення практичної задачі методом кластерного аналізу в системі STATISTICA. Задачею кластерного аналізу є організація спостережуваних даних в наочні структури. Для вирішення даної задачі в кластерному аналізі скористаємося наступними методами - *Joining* (tree clustering - ієрархічні агломеративні методи або деревовидна кластеризація), *K-means clustering* (метод k-середніх), *Two-way joining* (двохвходове об'єднання). Для цього за допомогою перемикача модулів STATISTICA відкриємо модуль *Cluster Analysis* (Кластерний Аналіз). На екрані з'явиться стартова панель модуля Clustering Method (методи кластерного аналізу), яка дозволяє вибрати необхідний метод кластерного аналізу. Розглянемо кожен з цих методів.

Joining (tree clustering) (ієрархічні агломеративні методи). Відкриємо файл даних. Після вибору *Joining* (tree clustering) з'являється вікно *Cluster Analysis: Joing (Tree Clustering)* (вікно введення режимів роботи для ієрархічних агломеративних методів) (рис. 7.39), в якому опція *Variables* дозволяє вибрати змінні, що беруть участь в класифікації. Виберемо всі змінні *Select All* .

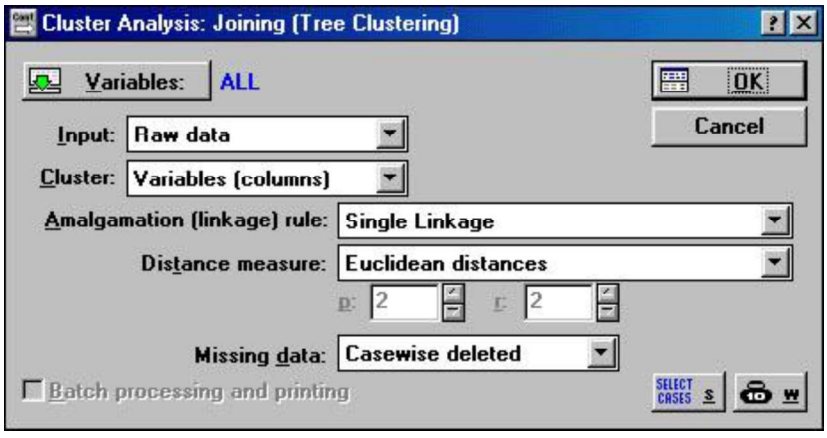


Рис.7.39. Cluster Analysis - Joing (Tree Clustering).

Також можна задати *Input* (тип вхідної інформації) і *Cluster* (режим класифікації (по ознаках або об'єктах)). Можна вказати *Amalgamation rule* (правило об'єднання) і *Distance measure* (метрика відстаней). Опція *Codes for grouping variable* (коди для груп змінної) вказуватимуть кількість аналізованих груп об'єктів, а опція *Missing data* (пропущені змінні) дозволяє вибрати або порядкове видалення змінних із списку, або замінити їх на середні значення. Після задання всіх необхідних параметрів будуть виконані обчислення, а на екрані з'явиться вікно, що містить результати кластерного аналізу «Joining Results» (рис.7.40).

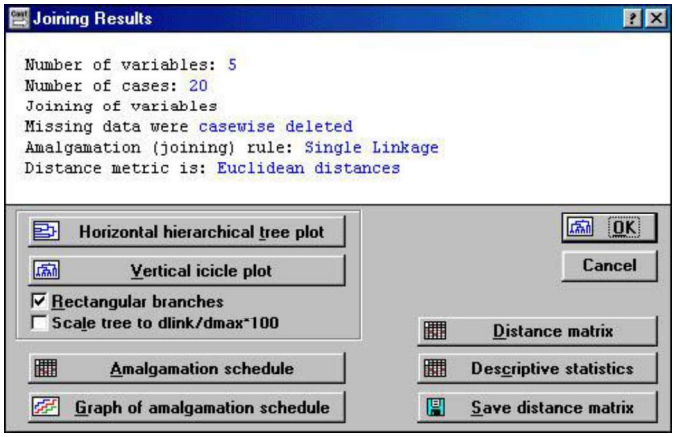


Рис.7.40. Вікно результатів кластерного аналізу «Joining Results».

Користувач може викликати на екран горизонтальну і вертикальну діаграму (Horizontal hierachical plot або Vertical icicle plot) (рис. 7.41).

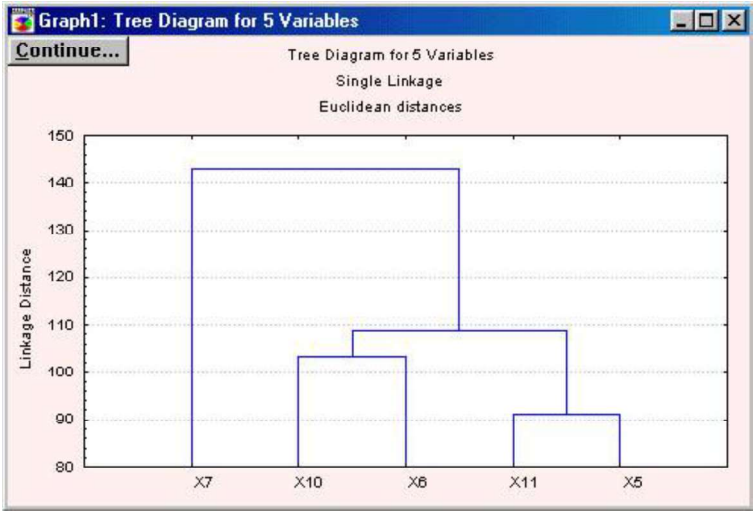


Рис. 7.41. Vertical icicle plot.

Аби повернутися у вікно, що містить інші результати кластерного аналізу, необхідно використовувати опцію *Continue*. Опція *Discriptive statistics* містить такі важливі описові статистики, як середнє (means) і середньоквадратичне відхилення (standart deviations) для кожного спостереження.

K - means clustering (метод k середніх). Із стартової панелі модуля *Clustering Method* (методи кластерного аналізу) виберемо *K - means clustering* (метод *k* -середніх). Відкриємо файл даних. У вікні *Cluster Analysis*: *K - means clustering* (рис. 7.42) опція *Variables* дозволяє вибрати змінні, що беруть участь в кластеризації. Виберемо всі змінні *Select All* . Опція *Cluster* вказує як ведеться кластеризація: при запуску встановлений режим *Variables* (colums) - кластеризуються змінні на підставі їх спостережень, проте в переважній більшості випадків використовується режим Cases (rows) – класифікуються спостереження.

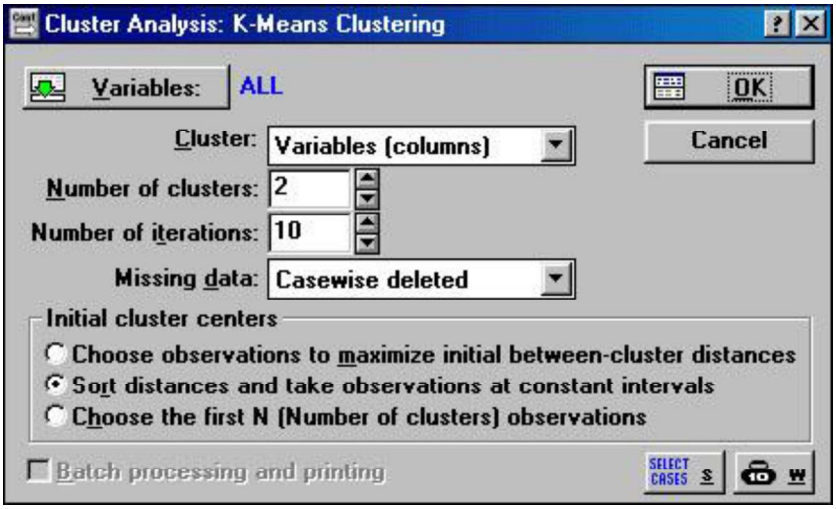


Рис.7.42. Cluster Analysis: K - means clustering.

Опція *Number of iterations* вказує кількість ітерацій в розрахунках кластерів. Як правило, встановлених за умовчанням 10 ітерацій цілком достатньо. Опція *Missing data* встановлює режим роботи з тими спостереженнями, в яких пропущені дані. Якщо встановити режим *Subsituted by means* (Замінювати на середнє), то замість пропущеного числа буде використано середнє по цій

змінній (або спостереженню). Після проведення обчислень з'явиться нове вікно: «K - Means Clustering Results» (рис. 7. 43).

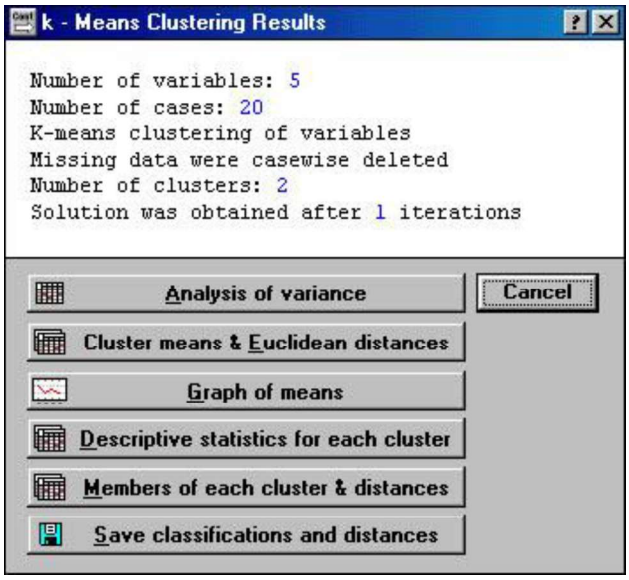


Рис. 7.43. K - Means Clustering Results.

У нижній частині вікна розташовані опції для виведення різної інформації по кластерах.

1. *Analysis of Variance* (аналіз дисперсії). Після застосування опції з'являється таблиця, в якій приведена міжгрупова і внутрішньогрупова дисперсії, в якій рядки - змінні (спостереження), а стовпці - показники для кожної змінної: дисперсія між кластерами, число мір свободи для міжкласової дисперсії, дисперсія всередині кластерів, число мір свободи для внутрікласової дисперсії, F - критерій, для перевірки гіпотези про нерівність дисперсій. Перевірка даної гіпотези схожа на перевірку гіпотези в дисперсійному аналізі, коли робиться припущення про те, що рівні чинника не впливають на результат.

2. *Cluster Means & Euclidean Distances* (середні значення в кластерах і відстань Евкліда). Виводяться дві таблиці. У першій вказані середні величини класу по всіх змінних (спостереженням). По вертикалі вказані номери класів, а по горизонталі змінні (спостереження). У другій таблиці приведені відстані між

класами, а по вертикалі і по горизонталі вказані номери кластерів. Таким чином при пересіченні рядків і стовпців вказані відстані між відповідними класами.

3. *Graph of means* є графічним зображенням (рис. 7.44) тієї інформації, що міститься в таблиці, яка виводиться при застосуванні опції *Analysis of Variance*. На графіці показані середні значення змінних для кожного кластера.

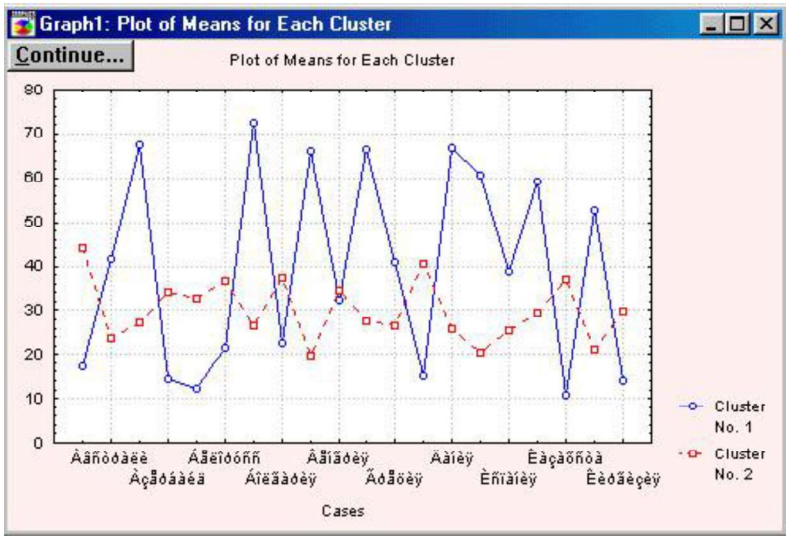


Рис. 7.44. Graph of means.

4. *Descriptive Statistics for each cluster* (описова статистика для кожного кластера). Після використання цієї опції виводяться вікна, кількість яких дорівнює кількості кластерів. У кожному такому вікні в рядках вказані змінні (спостереження), а по горизонталі їх характеристики, розраховані для даного класу: середнє, незміщене середньоквадратичне відхилення, незміщена дисперсія.

5. *Members for each cluster & distances*. Виводиться стільки вікон, скільки задано класів. У кожному вікні вказується загальне число елементів, віднесених до цього кластера, у верхньому рядку вказаний номер спостереження (змінної), віднесеного до даного класу і відстань Евкліда від центру класу до цього спостереження (змінної). Центр класу - середні величини по всіх змінних (спостереженням) для цього класу.

У системі STATISTICA реалізовані також і інші методи кластеризації, наприклад *Two-way joining*, в якому кластеризуються випадки і змінні

одночасно. На рис. 7.45 показаний результат кластеризації. Трудність з інтерпретацією отриманих результатів цим методом виникає внаслідок того, що схожість між різними кластерами може походити з деякої відмінності підмножин змінних. Тому отримані кластери є за своєю природою неоднорідними.

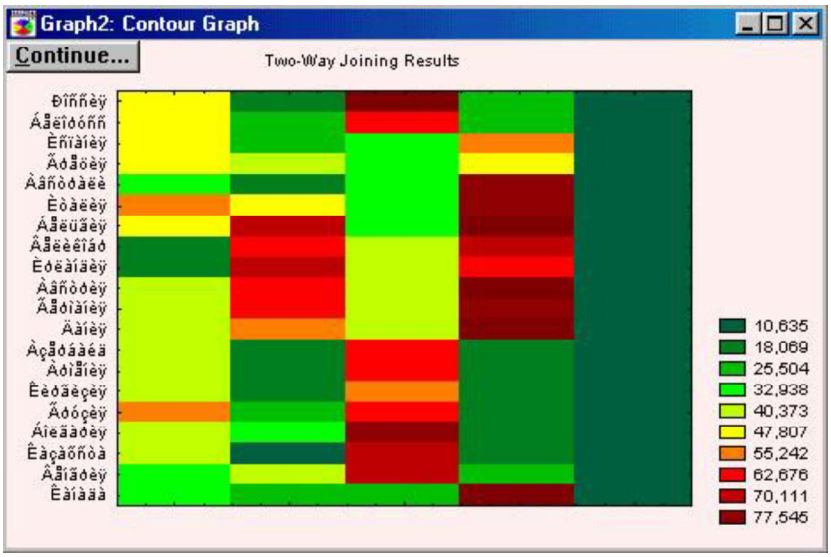


Рис. 7.45. Результат кластеризації Two-way joining методом.

Пакет програм SPSS (Statistical Package for Social Science). Пакет програм є одним з поширених, потужних і зручних інструментів статистичного аналізу. Пакет SPSS користується популярністю у економістів, соціологів, маркетологів, надає користувачеві широкі можливості по статистичній обробці емпіричних даних, по формуванню і модифікації баз даних, а також по створенню звітів, надаючи широкі можливості по представленню результатів статистичної обробки в текстовій, табличній і графічній формах. Пакет орієнтований, головним чином, на аналіз просторових даних і на кластерний аналіз. Інтерфейс програми інтуїтивно зрозумілий користувачеві і дозволяє застосувати різні варіанти статистичного аналізу.

Опишемо практичний підхід до виділення сегментів споживачів методом ієрархічного кластерного аналізу. Хай досліджується поведінка споживачів пельменів. В ході опиту респондентам задавалася ціла низка

запитань, призначених для вирішення задач, що стоять перед дослідженням. Також з'ясовувалися соціально-демографічні параметри респондентів: стать, вік і рівень доходів. В якості критерію сегментації виділимо 4 змінні: частота покупки, кратність покупки, наявність марочних переваг і орієнтація на ціну/якість. Як дескриптори сегментів використовуватимемо стать, вік і рівень доходів.

Ієрархічний кластерний аналіз проводиться в два етапи. Єдиним результатом першого етапу повинно стати число кластерів (цільових сегментів), на які слід розділити досліджувану вибірку респондентів. На другому етапі виконується власне кластеризація респондентів по тому числу кластерів, яке було визначено в ході першого етапу аналізу.

Процедура кластерного аналізу в SPSS запускається за допомогою меню: *Analyze Classify Hierarchical Cluster* (Аналіз Класифікація Ієрархічний кластерний аналіз). У діалоговому вікні (рис. 7.46) необхідно з лівого списку всіх наявних у файлі даних змінних вибрати змінні, що є критеріями сегментації.

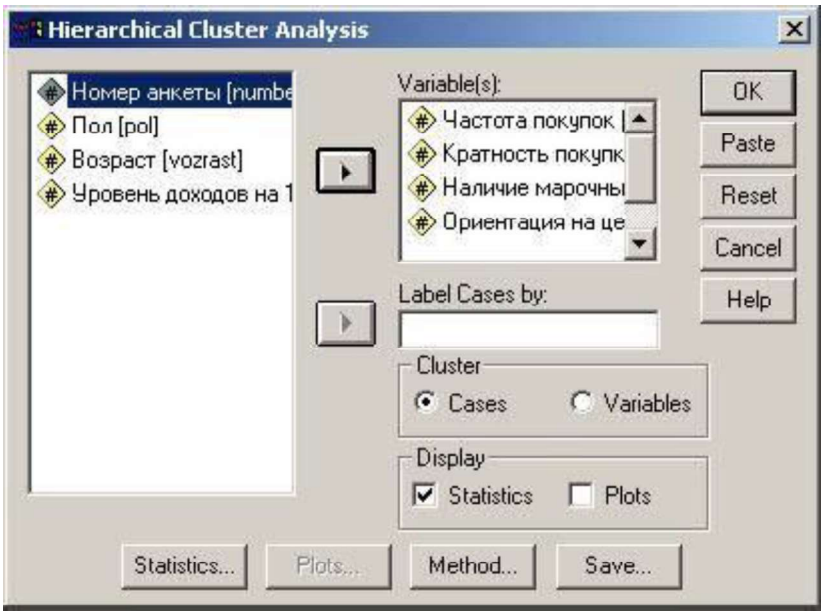


Рис. 7.46. Діалогове вікно «Hierarchical Cluster Analysis».

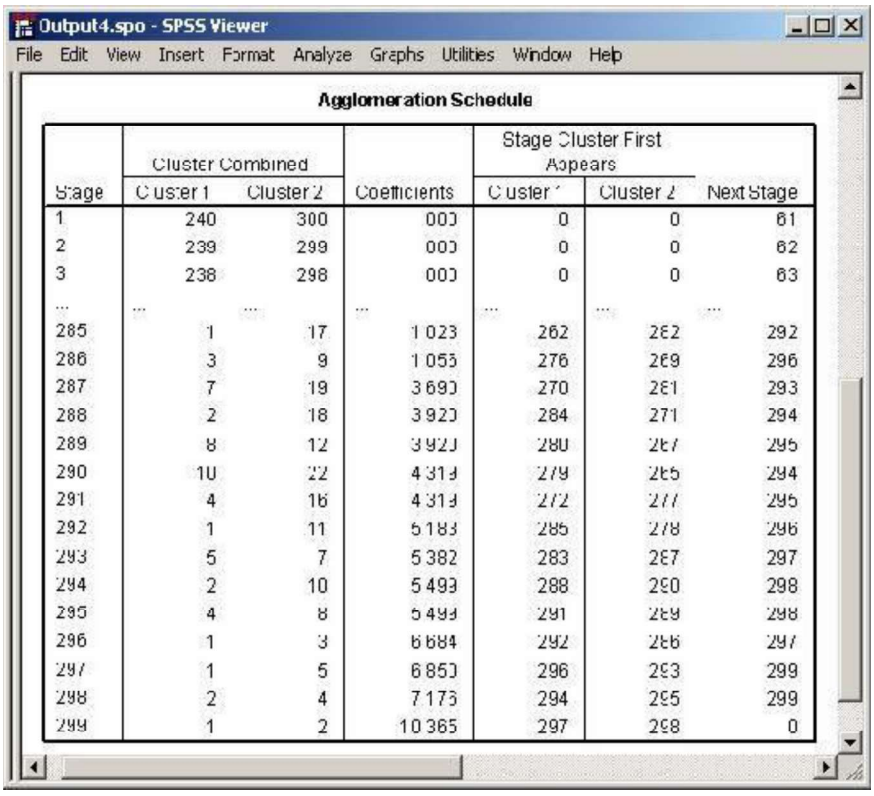
За умовчанням SPSS виводить також спеціальну перевернуту гістограму, названу «Icicle» (сосульковидна діаграма). За задумом творців

програми, вона допомагає визначити оптимальну кількість кластерів (виведення спеціальних видів діаграм здійснюється за допомогою опції «Plots» (діаграми)). Окрім «Icicle» SPSS дозволяє вибрати швидку лінійчату діаграму «Dendogram» (дендограмма). Теоретично при невеликій (до 50—100) кількості респондентів дана діаграма допомагає вибрати оптимальне рішення відносно необхідного числа кластерів. Проте практично у всіх прикладах з реальних маркетингових досліджень розмір вибірки перевищує це значення. Дендограмма в даному випадку стає абсолютно даремною, оскільки навіть при відносно невеликому числі спостережень є дуже довгою послідовністю номерів рядків вихідного файлу даних, сполучених між собою горизонтальними і вертикальними лініями.

Після вказівки критеріїв сегментації слід вибрати метод проведення кластерного аналізу. Це дозволяє зробити спеціальне діалогове вікно *Hierarchical Cluster Analysis: Method*, що викликається опцією «Method». Серед всіх можливих варіантів статистичних методик, пропонованих SPSS, рекомендується вибирати або встановлений за умовчанням метод «Between-groups linkage» (зв'язок між групами), або процедуру Ward'а («Ward's method»). При цьому перший метод використовується найчастіше зважаючи на його універсальність і відносну простоту статистичної процедури, на якій він заснований. При цьому методі відстань між кластерами обчислюється як середнє значення відстаней між всіма можливими парами спостережень, причому в кожній ітерації бере участь одне спостереження з одного кластера, а інше - з іншого. Метод Ward'а складається з множини етапів і заснований на усереднюванні значень всіх змінних для кожного спостереження і подальшому підсумовуванні квадратів відстаней від обчислених середніх до кожного спостереження. Також слід вибрати метод для обчислення відстаней між спостереженнями (область «Measure» (шкала) в даному діалоговому вікні). Найбільш часто використовуваним методом визначення відстаней для інтервальних змінних є квадрат відстані («Squared Euclidean Distance») Евкліда, що встановлюється за умовчанням. Саме даний метод найкраще

зарекомендував себе в маркетингових дослідженнях як найбільш точний і універсальний.

Після задання всіх необхідних параметрів виконуємо розрахунок першого етапу кластерного аналізу, результати якого з'являться у вікні *SPSS Viewer* (звіт SPSS). Єдиним значимим підсумком першого етапу аналізу буде таблиця «Average Linkage (Between Groups)» (усереднені зв'язки між групами), представлена на рис 7.47. По ній і повинне визначитися оптимальне число кластерів.



Stage	Cluster Combined		Coefficients	Stage Cluster First Appears		Next Stage
	Cluster 1	Cluster 2		Cluster 1	Cluster 2	
1	240	300	000	0	0	61
2	239	299	000	0	0	62
3	238	298	000	0	0	63
...	...	...	...	...	...	...
285	1	17	1 023	262	282	292
286	3	9	1 055	276	269	296
287	7	19	3 690	270	281	293
288	2	18	3 920	284	271	294
289	8	12	3 920	280	267	295
290	10	22	4 319	279	265	294
291	4	16	4 319	272	277	295
292	1	11	5 183	285	278	296
293	5	7	5 382	283	267	297
294	2	10	5 493	288	260	298
295	4	8	5 499	291	269	298
296	1	3	6 684	292	266	297
297	1	5	6 850	296	293	299
298	2	4	7 175	294	295	299
299	1	2	10 365	297	298	0

Рис. 7.47. Таблиця «Average Linkage (Between Groups)».

Перш за все спробуємо застосувати найбільш поширений, стандартний метод для визначення числа кластерів. Спочатку по таблиці «Average Linkage (Between Groups)» визначимо, на якому кроці процесу формування кластерів (колонка «Stage») відбувається перший порівняно великий стрибок коефіцієнта агломерації (колонка «Coefficients»). Даний стрибок означає, що до нього в кластери об'єднувалися спостереження, що знаходяться на досить малих відстанях одне від одного, а з цього етапу починає відбуватися об'єднання

більш далеких спостережень. У нашому випадку коефіцієнти плавно зростають від 0 до 1,056, тобто різниця між коефіцієнтами на кроках з першого по 286 включно була вельми мала (наприклад, між 286 і 285 кроками — всього 0,033). Проте починаючи з 287 кроку відбувається перший істотний стрибок коефіцієнта: з 1,056 до 3,690 (на 2,634). Таким чином, ми визначили крок, на якому відбувається перший стрибок коефіцієнта - 287. Тепер, аби визначити оптимальну кількість кластерів, необхідно відняти отримане значення із загального числа спостережень (розміру вибірки). Загальний розмір вибірки в нашому випадку складає 300 споживачів пельменів, отже, розрахункова оптимальна кількість кластерів складає: 300 – 287 = 13. Отримано досить велике число кластерів, яке надалі складно інтерпретуватиме. Тому необхідно досліджувати отримані кластери і визначити, які з них є значимими, а які слід спробувати скоротити. Ця задача вирішується на другому етапі кластерного аналізу.

Знов відкриємо головне діалогове вікно процедури кластерного аналізу (меню: *Analyze Classify Hierarchical Cluster*). У полі для аналізованих змінних вже є необхідні нам чотири параметри. Діалогове вікно, що відкрилося, дозволяє створити у вхідному файлі даних нову змінну, що розподіляє всіх респондентів на цільові групи. Виберемо параметр «Single Solution» (єдине рішення) і вкажемо у відповідному полі необхідне нам число кластерів - 13. Тепер слід знов запустити процедуру кластерного аналізу. В результаті у вхідному файлі даних SPSS буде створена нова змінна з назвою «clu13_1».

Аби дослідити, наскільки вірно ми визначили оптимальне число кластерів, побудуємо лінійний розподіл змінної «clu13_1» (меню: *Analyze Descriptive Statistics Frequencies* (Аналіз Описова статистика Лінійні розподіли)). Як видно з рис 7.48, в кластерах з 7 по 13 число спостережень вагається від 1 до 2. Подібна ситуація зустрічається практично завжди, тому число кластерів, визначене на першому етапі аналізу, майже ніколи не буває істинно оптимальним. Тому разом з вищеописаним універсальним методом визначення оптимальної кількості кластерів існує також додаткове обмеження:

розмір кластерів має бути статистично значимим і практично прийнятним. Наприклад, при нашому розмірі вибірки таке критичне значення можна встановити хоч би на рівні 10 респондентів на один кластер. Оскільки даній умові відповідають 6 кластерів, нам необхідно перерахувати процедуру кластерного аналізу із збереженням 6-кластерного рішення (буде створена нова змінна «clu6_1»).

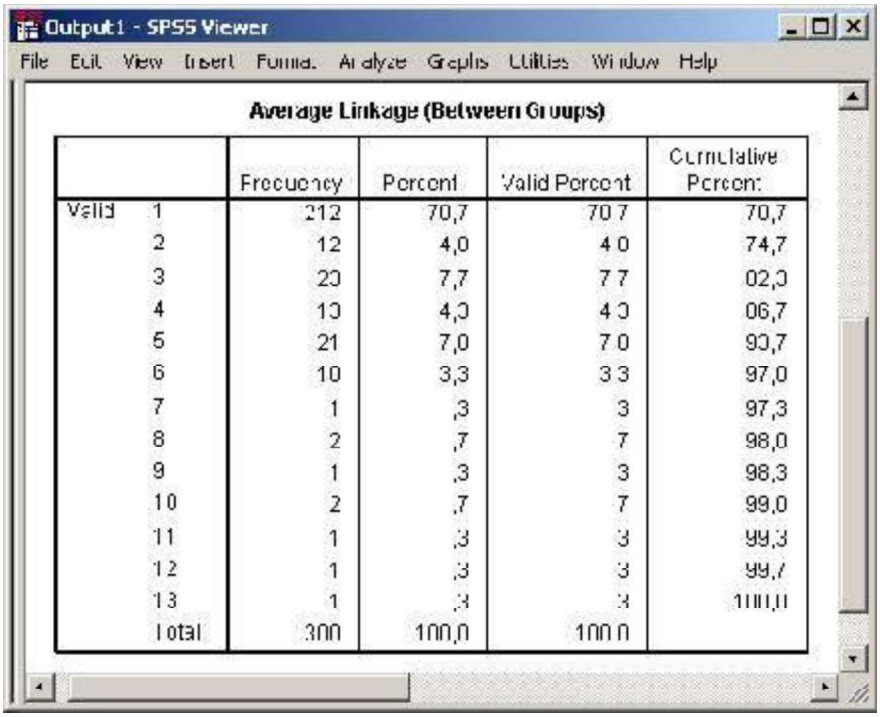


Рис. 7.48. Лінійний розподіл для 13-кластерного рішення.

Побудувавши лінійний розподіл по знов створеній змінній «clu6_1», ми побачимо, що лише в трьох кластерах число респондентів більше 10. Отже, нам необхідно знов перебудувати кластерну модель, тепер для 3-кластерного рішення. Після цього знову побудуємо розподіл по змінній «clu3_1». У загальному випадку дану процедуру слід продовжувати до тих пір, поки не вийде рішення, в якому на кожен кластер доводитиметься статистично значиме число респондентів. У нашому випадку 3-кластерне рішення виявилося оптимальним.

Приступимо до завершуючого етапу кластерного аналізу: інтерпретації отриманих цільових груп (сегментів). Опис отриманих сегментів проводиться в

два етапи: опис з точки зору критеріїв сегментації і опис з точки зору дескрипторів сегментів. Сегменти, виділені в результаті кластерного аналізу, характеризуються однорідністю значень критеріїв сегментації усередині кожного кластера і відмінністю між кластерами. Тому, по-перше, слід визначити, якими конкретно значеннями змінних, вибраних як критерії сегментації, характеризуються отримані кластери. Для цього найчастіше будують перехресний розподіл, в якому по стовпцях розташовується кластеризуюча змінна (у нашому випадку це «clu3_1»), а по рядках - критерії сегментації. Таким чином, можна бачити, в який кластер потрапляють респонденти з тим або іншим значенням критерію сегментації. Наприклад, в нашому випадку ми отримали 3 сегменти, які (за допомогою перехресного розподілу) описуються таким чином.

Сегмент 1. Часті покупці пельменів (1 раз на тиждень і частіше), які за один прихід в магазин купують невелику кількість продукту (до 1 кг); при цьому вони не звертає уваги на марку і орієнтуються в основному на ціну.

Сегмент 2. Відносно рідкі покупці пельменів (рідше за 1 раз в тиждень), які за один прихід в магазин купують значну кількість продукту (більше 1 кг); при цьому вони не звертає уваги на марку і орієнтуються в основному на ціну.

Сегмент 3. Відносно рідкі покупці пельменів (рідше за 1 раз в тиждень), які за один прихід в магазин купують значну кількість продукту (більше 1 кг); при цьому вони звертають увагу на марку і орієнтуються в основному на якість.

Отже, після побудови перехресного розподілу стає очевидною різниця в ключових характеристиках сегментів. Виділеним сегментам стає можливим дати вербальні назви. Крім того, з процентного співвідношення отриманих сегментів можна оцінити частку ринку, що займає кожен з них, і виявити найпривабливіші цільові групи. Другим, завершальним етапом в інтерпретації результатів кластерного аналізу є поглиблений опис отриманих сегментів за допомогою дескрипторних змінних. Таким чином, сегменти все більше знаходять «людське обличчя». Опис сегментів дескрипторними змінними також проводиться за допомогою побудови перехресних розподілів способом,

аналогічним описаному вище. В результаті виходить повна картина сегментації ринку. З такими даними можна аргументовано вибирати найпривабливіші цільові сегменти і розробляти стратегію позиціювання для кожного з них.

Кластеризація в програмному комплексі «1С: Предприятие 8.0». Метою кластеризації в програмному комплексі є виділення з множини об'єктів однієї природи деякої кількості відносно однорідних груп - сегментів або кластерів. Об'єкти розподіляються по групах так, щоб внутрішньогрупові відмінності були мінімальними, а міжгрупові - максимальними (рис. 7.49). Методи кластеризації дозволяють перейти від пооб'єктного до групового представлення сукупності довільних об'єктів, що істотно спрощує операцію ними.

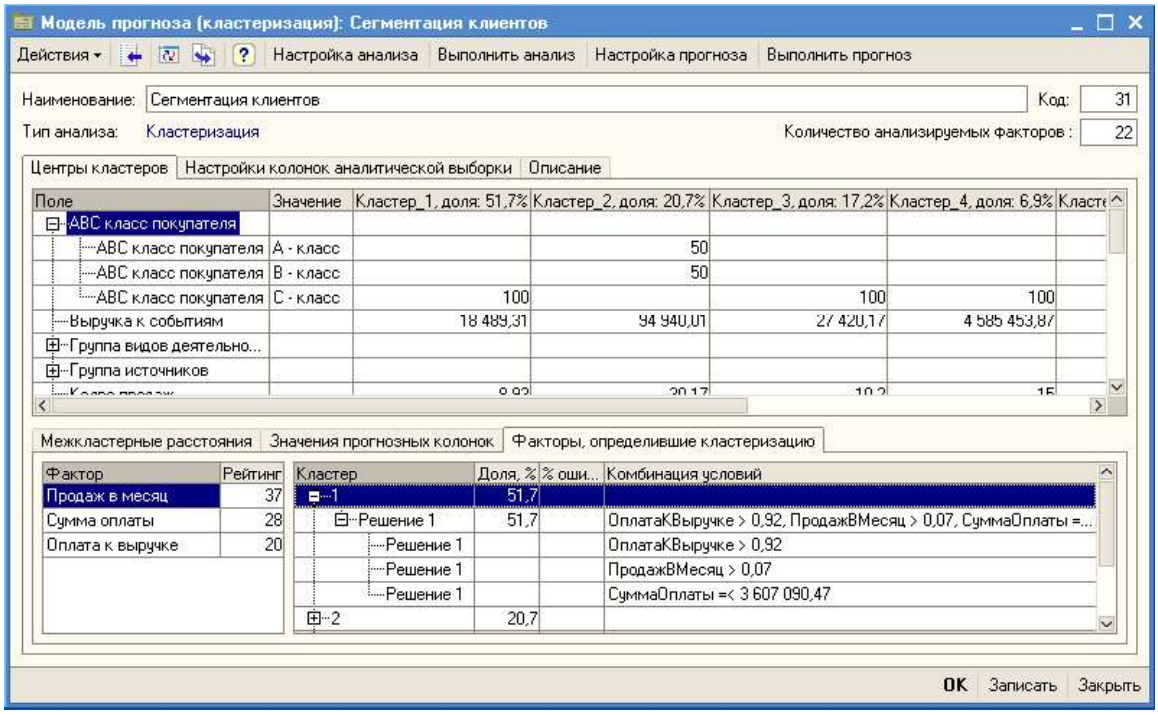


Рис. 7.49. Аналіз даних методом кластеризації.

Можливі сценарії вживання кластеризації на практиці:

1. Сегментація клієнтів по певній сукупності параметрів дозволяє виділити серед них стійкі групи, що мають схожі купівельні переваги, рівні продажів і платоспроможності, що істотно спрощує управління взаєминами з клієнтами.

2. При класифікації товарів часто використовуються досить умовні принципи класифікації. Виділення сегментів на основі групи формальних

критеріїв дозволяє визначити дійсно однорідні групи товарів. В умовах широкої і досить різнорідної номенклатури товарів управління асортиментом на рівні сегментів, в порівнянні з управлінням на рівні номенклатури, істотно підвищує ефективність просування, ціноутворення, мерчендайзинга, управління ланцюжками постачань.

3. Сегментація менеджерів дозволяє ефективніше спланувати організаційні зміни, поліпшити мотиваційні схеми, скоректувати вимоги до найманого персоналу, що кінець кінцем дозволяє підвищити керованість компанії і стабільність бізнесу в цілому.

Результатами аналізу за допомогою кластеризації є:

- центри кластерів, що є сукупністю усереднених значень вхідних колонок в кожному кластері;
- таблиця міжкластерних відстаней (між центрами кластерів), що визначають міру відмінності між ними;
- значення прогнозних колонок для кожного кластера;
- рейтинг чинників і дерево умов, що визначили розподіл об'єктів на кластери.

Алгоритми кластеризації дозволяють не лише провести кластерний аналіз об'єктів на множині заданих атрибутів, але і спрогнозувати значення одного або декількох з них для актуальної вибірки на підставі віднесення об'єктів цієї вибірки до того або іншого кластера.

В програмному комплексі на основі застосування кластерного аналізу реалізован типовий бізнес - сценарій «Управління взаємовідношеннями з клієнтами».

Сценарій - "Планування рекламної кампанії". Планерування майбутньої рекламної кампанії розглядається з точки зору оптимізації розподілу виділеного бюджету по рекламних каналах виходячи з регіонального, продуктового, клієнтського і інших показників цільового сегменту, а також ефективності рекламних каналів у вказаних розрізах в деякому, попередньому плановому періоді.

Прогнозні атрибути - долі відгуків на рекламний канал умовно однорідних сегментів, виділених алгоритмом.

Обчислювані колонки: долі рекламних каналів в бюджеті рекламної кампанії з врахуванням вірогідної долі відгуків і ефективності (у сенсі результуючої виручки) кожного рекламного каналу.

На останок розглянемо найбільш суттєві практичні напрями застосування технології кластерного аналізу.

Кластеризація текстової інформації. Величезні об'єми інформації в мережі Internet приводять до того, що кількість об'єктів, що видаються по запиту користувача, дуже велика. Це утрудняє процес огляду результатів і вибору відповідних матеріалів з множини знайдених. Проте, в більшості випадків величезні об'єми інформації можна зробити доступними для сприйняття, якщо уміти розбивати джерела інформації (наприклад, WEB-сторінки) на тематичні групи. Тоді, користувач відразу може відкидати множину документів з мало релевантних груп. Такий процес угрупування даних здійснюється за допомогою кластеризації корпусу текстів. На даний момент існує декілька методів, що здійснюють кластеризацію документів:

- LSA/LSI – Latent Semantic Analysis/Indexing. Шляхом факторного аналізу множини документів виявляються латентні (приховані) чинники, які надалі є основою для утворення кластерів документів;
- STC – Suffix Tree Clustering. Кластери утворюються у вузлах спеціального вигляду дерева – суффіксного дерева, яке будується із слів і фраз вхідних документів;
- Single Link, Complete Link, Group Average – ці методи розбивають множину документів на кластери, розташовані в деревовидній структурі, – dendrogramm, яка отримується за допомогою ієрархічної агломеративної кластеризації;
- K-means. Кластери представлені у вигляді центроїдів, що є «центром маси» всіх документів, що входять в кластер;

- Scatter/Gather. Представляється як ітеративний процес, що спочатку розбиває (scatter) множину документів на групи і представленні потім цих груп користувачеві (gather) для подальшого аналізу. Далі процес повторюється знову над конкретними групами.

На фазі розбиття метод може використовувати два алгоритми: Buckshot і Fractionation. Алгоритм Buckshot швидший і годиться для швидкої рекластеризації при виконанні ітерацій в Scatter/Gather. Fractionation же є точнішим і повільнішим алгоритмом і використовується в Scatter/Gather для попереднього розбиття на групи множини документів і виконується в режимі off-line. Обоє алгоритму відносяться до алгоритмів восходящої (bottom-up) кластеризації. Система Scatter/Gather існує у вигляді комерційного продукту і використовується для кластеризації різних текстових ресурсів (рис. 7.50).

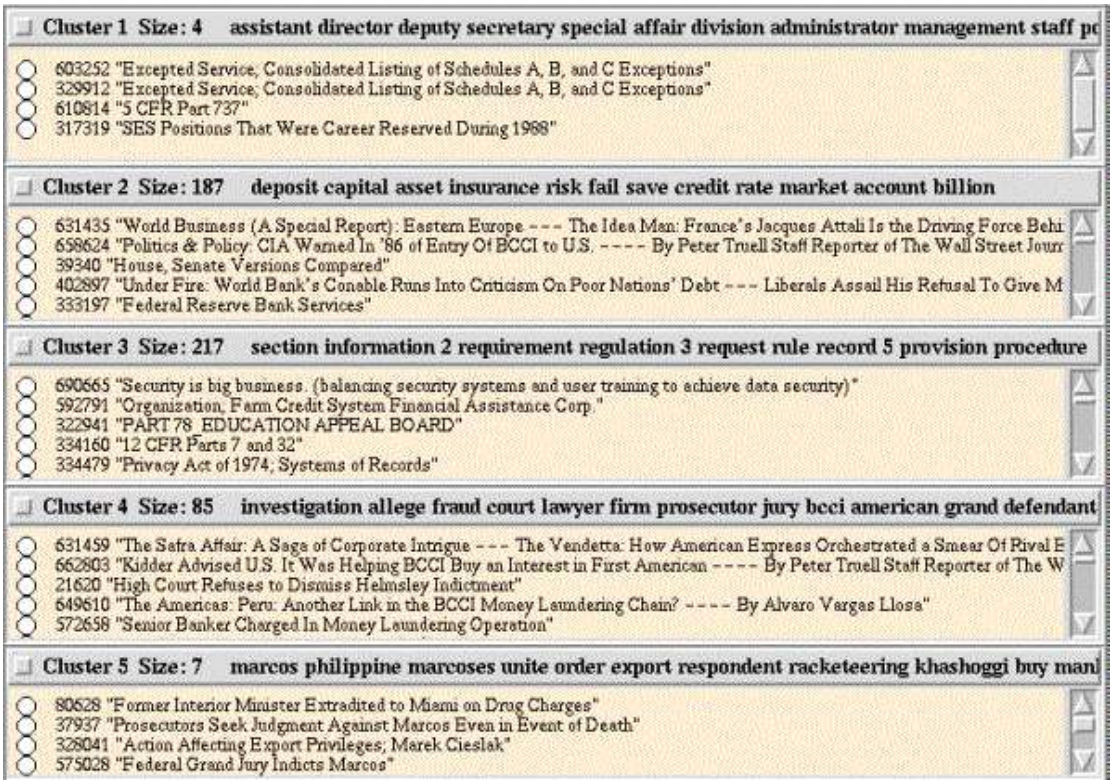


Рис.7.50. Результат запиту, розподілений по кластерах, в системі Scatter/Gather.

Кластерний аналіз в маркетингових дослідженнях. Кластерний аналіз все частіше знаходить застосування в маркетингових дослідженнях. Серед найбільш популярних напрямів маркетингу виділяють такі:

1. *Сегментація*. Кластерний аналіз застосовується для вирішення широкого спектру задач, але найчастіше йдеться саме про задачу сегментації. Всі дослідження, присвячені проблемі сегментації, безвідносно того, який використовується метод, мають на меті ідентифікувати стійкі групи (люди, ринки, організації), кожна з яких об'єднує в собі об'єкти з схожими характеристиками.

2. *Аналіз поведінки споживача.* Другим, але не менш важливі напрямом використання апарату кластерного аналізу, є побудова однорідних груп споживачів з метою отримати максимально повне уявлення про те, як поводиться клієнт з кожного сегменту, які ознаки визначають його поведінку.

3. *Позиціонування.* Кластерний аналіз застосовується також для того, щоб визначити, в якій ніші краще позиціонувати продукт, що виводиться на ринок. Кластерний аналіз дозволяє побудувати карту, на основі якої можна буде визначити рівень конкуренції в різних сегментах і характеристики, якими повинен володіти товар для того, щоб попасти в цільовий сегмент. Така карта дозволяє, наприклад, виявити нові ринки, для яких можна розробляти і просувати свої рішення.

4. *Вибір тестових ринків.* Багато маркетологів застосовують кластерний аналіз для того, щоб, визначити, які ринки (магазини, продукти, ...) можна об'єднати в одну групу за релевантними характеристиками. Річ у тому, що, висунувши припущення про існування певної закономірності необхідно запропонувати новий, не використаний в аналізі, ринок, на якому вона має бути перевірена, перш ніж застосовувати на практиці.

Кластеризація в фінансовій діяльності. Величезна кількість інвестиційних інструментів, що надається сучасним фінансовим ринком, заставляє корпоративних інвесторів з кожним днем аналізувати все більшу кількість фінансової інформації. Часом успіх інвестування залежить від об'єму аналізованих фінансових даних, часу, витраченого на аналіз, і вигляду, в якому представлені результати. Більше, швидше, зручніше - ось основні вимоги, що пред'являються постійно зміняющимся фінансовим ринком до методів аналізу

фінансових даних. Основу для того, щоб упевнено відповідати на виклики ринку, дають методи кластеризації. Процедура кластеризації вирішує питання про схожість фінансових активів, що характеризуються значеннями багатьох параметрів, на основі формальних математичних критеріїв. Це дозволяє замінити тривалий і трудомісткий процес вивчення і порівняння активів швидшим обчислювальним алгоритмом. Крім того, будучи засобом аналізу багатовимірних даних, кластеризація дозволяє виділити активи з близькими значеннями всіх параметрів.

Використання кластеризації виправдане скрізь, де потрібний аналіз багатовимірних фінансових даних. Один з прикладів - вибір фінансових інструментів відповідно до багатофакторної моделі оцінки прибутковості активів. Допустимо, що для формування диверсифікованого портфеля акцій потрібно проаналізувати чутливість 150 компаній NASDAQ до чотирьох чинників. Мірою чутливості до кожного чинника є показники b_i багатофакторної моделі. Таким чином, кожна компанія характеризується набором з чотирьох значень b_1, b_2, b_3, b_4. Виходить, що необхідно розглянути більше 600 чисел, а потім якимсь чином відранжирувати компанії. Метод звичайного сортування не дасть жодних результатів - якщо сортування виконується по показнику b_1, що характеризує чутливість до першого чинника, то після виділення активів з близьким значенням b_1 може виявитися, що ці компанії істотно відрізняються по значеннях b_2, b_3 та b_4.

Як видно з рим. 7.51, в результаті кластеризації все 150 компаній розподілено по трьох групах. У кожній групі автоматично зібрані компанії з близькими значеннями показників b. Замість вивчення 600 значень b тепер для здобуття повного уявлення про ситуацію, що склалася в даний момент на ринку, досить проаналізувати показники b лише по трьох компаніях (центрам груп). Показники b центру групи найбільш близькі до середніх по всій групі.

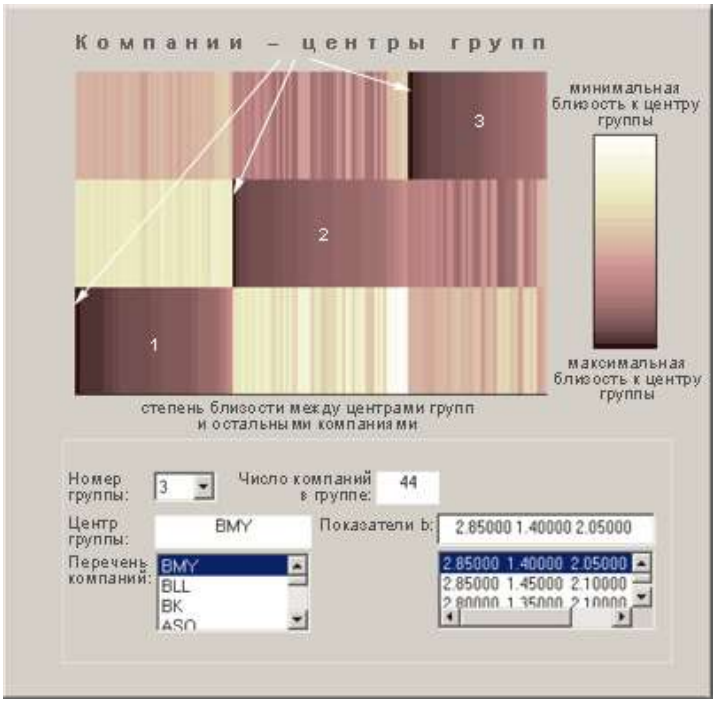


Рис. 7.51. Результати кластеризації компаній.

Аналіз стовпців в таблиці показує, що перша група найбільш чутлива до другого чинника, але значно менш чутлива до всіх інших. Друга - найбільш чутлива до всіх чинників, за винятком другого. По відношенню до нього спостерігається середня чутливість. Остання група об'єднує компанії, що демонструють середню чутливість до 1-го, 3-го і 4-го чинника і мінімальну, - до 2-го. Таким чином, якщо портфельний менеджер ставить перед собою завдання підняти очікувану прибутковість портфеля без збільшення ризику, пов'язаного з чинником b_4, наприклад, цінами на нафту, і підвищеною чутливістю до чинника b_1, наприклад, рівню довіри інвесторів, то далі з одновимірного масиву першої групи легко вибираються компанії з необхідною нормою прибутковості.