МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ЗАПОРІЗЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

ІНТЕЛЕКТУАЛЬНИЙ АНАЛІЗ ДАНИХ

Конспект лекцій для здобувачів ступеня вищої освіти бакалавра
спеціальності «Економіка» освітньо-професійної програми
«Економічна кібернетика»

Затверджено
вченою радою ЗНУ
Протокол №__ від _____2020

Запоріжжя
2020

УДК: 330.46:311:33(075.8)
І-20

Інтелектуальний аналіз даних: конспект лекцій для здобувачів ступеня вищої освіти бакалавра спеціальності «Економіка» освітньо-професійної програми «Економічна кібернетика» / Укладачі: Іванов С.М., Максишко Н.К., Бречко Д.О. Запоріжжя: ЗНУ, 2020. 156 с.

Зміст навчального видання спрямовано на опанування курсу «Інтелектуальний аналіз даних» – науки, метою якої є застосування теоретичних відомостей до процесу аналізу даних за допомогою технології Data Mining, вмінню оперувати при цьому комбінацією вивчених методів, а також вибору найбільш раціональних методів та підходів до аналізу даних. У виданні викладено сутність інтелектуального аналізу даних, мету та призначення класичних статистичних методів аналізу та OLAP-систем, особливості роботи методів Data Mining (дерева рішень, нейронні мережі, методи обмеженого перебору, кластерні моделі, комбіновані методи та інше).

Навчальне видання містить теоретичні відомості з кожної теми, питання для самоконтролю та контрольні питання для розділів.

Навчальне видання сприятиме кращому засвоєнню студентами програмного матеріалу курсу «Інтелектуальний аналіз даних». Призначене для здобувачів ступеня вищої освіти бакалавра спеціальності «Економіка» освітньо-професійної програми «Економічна кібернетика».

Рецензент — *Г.Ю. Кучерова, д.е.н., доцент професор кафедри економіки КПУ*

Відповідальний за випуск — *Н.К. Максишко, д.е.н., професор кафедри економічної кібернетики Запорізького національного університету*

ЗМІСТ

ВСТУП

Курс «Інтелектуальний аналіз даних» –важливий курс у процесі формування сучасного фахівця з економіки. Під час вивчення матеріалів курсу студенти знайомляться з технологією Data Mining, її методами, інструментальними засобами та особливостями застосування.

Розглянуті в курсі системи доцільно використовувати для розв'язання задач соціально-економічного прогнозування і планування розвитку промислових галузей, підприємств і в інших службах, що утворюють інфраструктуру міст, областей і регіонів. Наукову основу курсу складають теоретичні моделі, математичний апарат, сучасні концепції та парадигми, які визначають підходи до вивчення характеристик систем інтелектуального аналізу даних.

У навчальному курсі обговорюються сутність Data Mining, основні моделі інтелектуальних обчислень, засоби комп'ютерної підтримки, сучасна практична діяльність та перспективні напрями розвитку. Впродовж вивчення курсу описується сфера застосування Data Mining. Докладно розглядаються методи Data Mining: нейронні мережі, дерева рішень, методи обмеженого перебору, генетичні алгоритми, еволюційне програмування, кластерні моделі, комбіновані методи. Знайомство з кожним методом проілюстровано вирішенням практичного завдання за допомогою інструментального засобу, що використовує технологію Data Mining.

Мета дисципліни – засвоєння студентами технології призначених для пошуку у великих обсягах даних неочевидних, об'єктивних і корисних на практиці закономірностей . Враховуючи прикладний характер курсу, особлива увага приділяється застосуванню теоретичних відомостей процесу аналізу даних за допомогою технології Data Mining, вмінню оперувати при цьому комбінацією вивчених методів, а також вибору ефективних методів та підходів до аналізу даних.

Завдання дисципліни: засвоїти основні поняття та теоретичні відомості курсу та простежити відмінності Data Mining від класичних статистичних методів аналізу та OLAP-систем. Розглянути типи закономірностей, що виявляються Data Mining. Ознайомитися з особливостями методів Data Mining (дерева рішень, нейронні мережі, методи цілеспрямованого перебору, кластерні моделі, комбіновані методи та інше). Сформувати навички застосування методів Data Mining для проведення інтелектуального аналізу даних.

У результаті вивчення дисципліни студенти повинні:

Знати:

– основні поняття, що пов'язані з базами даних, концепцією сховищ даних та системами підтримки прийняття рішень;

– основні задачі ІАД (Data Mining);

– класифікацію та особливості вибору та застосування технологічних методів ІАД;

– основні теоретичні відомості та етапи технології Data Mining;

– принципи та особливості роботи методів Data Mining.

Уміти:

– застосувати основні поняття та твердження при виборі відповідного методу аналізу;

– застосовувати при аналізі даних вивчені методи Data Mining;

– визначати типи закономірностей, що виявляються Data Mining;

– визначати типи похибок та точність прогнозу;

– виконувати практичні завдання за допомогою інструментальних засобів, що використовують технологію Data Mining.

Згідно з вимогами освітньої програми студенти повинні досягти таких результатів навчання (компетентностей): знати головні мережеві технології; основи функціонування та основні сервіси глобальної інформаційної мережі Інтернет, сутність інформаційних технологій, їх роль і місце в сучасному суспільстві, програмне забезпечення сучасних інформаційних систем та тенденції його розвитку, сутність е-бізнесу та його організаційно-економічні форми, механізми захисту комерційної інформації в Інтернеті і правове регулювання комерції в мережі, основні типи інформаційних систем в економіці та приклади їх реалізації; ефективно користуватися сервісами Інтернет, проводити закупівлі в електронних крамницях, брати участь в електронних аукціонах, проводити електронні платежі, визначати найбільш ефективні форми використання інформаційних технологій; володіти навичками розробки в стандартному програмному забезпеченні сайту електронної крамниці компанії та роботи в глобальному інформаційному середовищі, включаючи Інтернет-трейдинг у розрізі е-бізнесу.

Міждисциплінарні зв'язки. Викладанню курсу передує вивчення дисциплін «Інформатика», «Економічна теорія». Після вивчення курсу «Економічна теорія» студент повинен володіти системою знань про: економічну систему суспільства, закони її функціонування і розвитку для розуміння чинників зародження, утвердження і напрямів розвитку сучасних соціально-економічних систем, їх спроможності задовольняти потреби людей; про економічні відносини як суспільну форму виробництва, проблеми ефективного використання обмежених виробничих ресурсів і шляхи забезпечення суспільних потреб. Після вивчення курсу «Інформатика» студент повинен володіти системою знань про організацію обчислювальних процесів на персональних комп'ютерах та їх алгоритмізацію, програмне забезпечення персональних комп'ютерів і комп'ютерних мереж, вміти працювати з основними видами функцій, а також з масивами даних у програмному забезпеченні Microsoft Excel.

Навчальне видання із дисципліни «Інтелектуальний аналіз даних» призначене допомогти студентам у вивченні основних понять курсу, логічно і змістовно, відповідає робочій навчальній програмі. Запропоноване автором видання сприятиме оволодінню знаннями та уміннями, які передбачені цією дисципліною.

РОЗДІЛ І. ОСНОВНІ ПОНЯТТЯ ІНТЕЛЕКТУАЛЬНОГО АНАЛІЗУ ДАНИХ

Тема 1. Інтелектуальний аналіз даних (Data Mining). Особливості технології Data Mining та її відмінності від інших методів аналізу даних

План

1. Історія виникнення та причини розвитку.
2. Суть, мета та сфера застосування технології Data Mining.
3. Типи закономірностей.
4. Класи систем Data Mining.

Мета вивчення теми: засвоїти основні концептуальні поняття з курсу «Інтелектуальний аналіз даних»; засвоїти відмінності Data Mining від класичних статистичних методів аналізу й OLAP-систем, вивчити типи закономірностей, що виявляють Data Mining та класи систем інтелектуального аналізу даних.

Перелік ключових слів та понять із теми

Data Mining, асоціація, класифікація, послідовність, кластерізація, прогнозування

Теоретичні відомості з теми

1. Історія виникнення та причини розвитку

Поняття Data Mining, що з'явилося в 1978 р., набуло високої популярності в сучасному трактуванні приблизно з першої половини 90-х років. До цього часу обробка та аналіз даних здійснювалися в рамках прикладної статистики, при цьому в основному вирішувалися завдання обробки невеликих баз даних.

Термін **«Інтелектуальний аналіз даних»** походить від поняття **Data Mining,** котре отримало свою назву з двох понять: пошуку цінної інформації у великій базі **даних** (Data) і **видобутку** (Mining). Обидва процеси вимагають або просіювання величезної кількості сирого матеріалу, або розумного дослідження і пошуку цінностей. Також термін Data Mining часто перекладається як видобуток даних, витягування інформації, розкопування даних, **інтелектуальний аналіз даних**, засоби пошуку закономірностей, вилучення знань, аналіз шаблонів, розкопування знань у базах даних.

Поняття «Виявлення знань в базах даних» (knowledge discovery in databases, KDD) можна вважати синонімом інтелектуального аналізу даних.

Розвиток технології баз даних:

- 1960-і рр. У 1968 році була введена в експлуатацію перша промислова СУБД система IMS фірми IBM.
- 1970-і рр. У 1975 році з'явився перший стандарт асоціації по мовах систем обробки даних – Conference on Data System Languages (CODASYL), який визначив низку фундаментальних понять у теорії систем баз даних, які досі є

основоположними для мережевої моделі даних. У подальший розвиток теорії баз даних великий внесок був зроблений американським математиком Е.Ф. Коддом, який є творцем реляційної моделі даних.

- 1980-і рр. Протягом цього періоду багато дослідників експериментували з новим підходом у напрямках структуризації баз даних і забезпечення до них доступу. Метою цих пошуків було отримання реляційних прототипів для простішого моделювання даних. У результаті, в 1985 році була створена мова, названа SQL. На сьогоднішній день практично всі СУБД забезпечують цей інтерфейс.
- 1990-і рр. З'явилися специфічні типи даних – «графічний образ», «документ», «звук», «карта», типи даних для часу, інтервалів часу, символьних рядків із двобайтовим поданням символів були додані в мову SQL. З'явилися технології **Data Mining**, сховища даних, мультимедійні бази даних і веб-бази даних.

У зв'язку з удосконаленням технологій запису і зберігання даних на людей обвалилися колосальні потоки «інформаційного видобутку» в найрізноманітніших областях. Діяльність будь-якого підприємства (комерційного, виробничого, медичного, наукового і т.д.) тепер супроводжується реєстрацією та записом всіх подробиць його діяльності. Що робити з цією інформацією? Стало зрозумілим, що без продуктивної переробки потоки сирих даних утворюють нікому не потрібне звалище.

Специфіка сучасних вимог до такої переробки така:

• дані мають необмежений обсяг;

• дані є різнорідними (кількісними, якісними, текстовими);

• результати мають бути конкретні і зрозумілі;

• інструменти для обробки сирих даних повинні бути прості у використанні.

Традиційна математична статистика, яка довгий час претендувала на роль основного інструменту аналізу даних не могла більше ефективно вирішувати ці завдання. Головна причина – концепція усереднення за вибіркою, що призводить до операцій над фіктивними величинами (типу середньої температури пацієнтів по лікарні, середньої висоти будинку на вулиці і т.п.). Методи математичної статистики виявилися корисними головним чином для перевірки заздалегідь сформульованих гіпотез (перевірка керованості інтелектуального аналізу даних) і для «грубого» розвідувального аналізу, що становить основу оперативної аналітичної обробки даних (аналітична обробка в реальному часі, online analytical processing, OLAP).

Причини популярності Data Mining:

• стрімке накопичення даних;

• загальна комп'ютеризація бізнес-процесів;

• проникнення Інтернету у всі сфери діяльності;

• прогрес в області інформаційних технологій: вдосконалення СУБД і сховищ даних;

• прогрес в області виробничих технологій: стрімке зростання продуктивності комп'ютерів, об'ємів накопичувачів, впровадження Grid систем.

Про популярність Data Mining говорить і той факт, що результат пошуку терміну «Data Mining» у пошуковій система Google (на вересень 2017 року) – становить більше 18 мільярдів сторінок, на вересень 2013 – 198 мільйонів сторінок.

Data Mining – мультидисциплінарна галузь, що виникла і розвивається на базі таких наук як прикладна статистика, розпізнавання образів, штучний інтелект, теорія баз даних тощо.

2. Суть, мета та сфера застосування технології Data Mining

Суть та мету технології Data Mining можна охарактеризувати так: це технологія, яка призначена для пошуку у великих обсягах даних неочевидних, об'єктивних і корисних на практиці закономірностей.

Неочевидних – означає, що знайдені закономірності не виявляються стандартними методами обробки інформації або експертним шляхом.

Об'єктивних – означає, що виявлені закономірності будуть повністю відповідати дійсності, на відміну від експертної думки, яка завжди є суб'єктивним.

Практично корисних – означає, що висновки мають конкретне значення, котрому можна знайти практичне застосування.

Знання – сукупність відомостей, яка утворює цілісний опис, відповідне деякому рівню обізнаності про описуване питання, предмет, проблему тощо.

Використання знань означає дійсне застосування знайдених знань для досягнення конкретних переваг (наприклад, в конкурентній боротьбі за ринок).

Наведемо ще кілька визначень поняття Data Mining.

Data Mining – це процес виділення з даних неявної і неструктурованою інформації та представлення її у вигляді, придатному для використання.

Data Mining – це процес виділення, дослідження і моделювання великих обсягів даних для виявлення невідомих до цього структур (моделей) з метою досягнення переваг у бізнесі (визначення SAS Institute).

Data Mining – це процес, мета якого – виявити нові значущі кореляції, зразки і тенденції в результаті просіювання великого обсягу збережених даних з використанням методик розпізнавання зразків плюс застосування статистичних і математичних методів (визначення Gartner Group).

Сфера застосування Data Mining нічим не обмежена – вона скрізь, де є будь-які дані. Але в першу чергу методи Data Mining сьогодні, м'яко кажучи, заінтригували комерційні підприємства, що розгортають проекти на основі інформаційних сховищ даних (сховища даних). Досвід багатьох таких підприємств показує, що віддача від використання Data Mining може досягати 1000%. Наприклад, відомі повідомлення про економічний ефект, що в 10-70 разів перевищив початкові витрати від 350 до 750 тис. дол. Є відомості про проект в 20 млн. дол., який окупився всього за 4 місяці. Інший приклад – річна економія 700 тис. дол. за рахунок впровадження Data Mining у мережі універсамів у Великобританії.

Data Mining становлять велику цінність для керівників та аналітиків в їх повсякденній діяльності. Ділові люди усвідомили, що за допомогою методів Data Mining вони можуть отримати відчутні переваги в конкурентній боротьбі. Коротко охарактеризуємо деякі можливі бізнес-додатки інтелектуального аналізу даних.

Роздрібна торгівля. Підприємства роздрібної торгівлі сьогодні збирають докладну інформацію про кожну окрему покупку, використовуючи кредитні картки з маркою магазина і комп'ютеризовані системи контролю. Ось типові завдання, які можна вирішувати за допомогою Data Mining у сфері роздрібної торгівлі:

• *аналіз купівельної корзини* (аналіз подібності) призначений для виявлення товарів, які покупці прагнуть купувати разом. Подібний аналіз потрібен для поліпшення реклами, вироблення стратегії створення запасів товарів і способів їх розкладки в торгових залах;

• *дослідження тимчасових шаблонів* допомагає торговим підприємствам приймати рішення про створення товарних запасів. Воно дає відповіді на питання типу «Якщо сьогодні покупець придбав відеокамеру, то через який час він найімовірніше купить нові батарейки і плівку?»;

• *створення прогнозуючих моделей* дає можливість торговельним підприємствам дізнаватися про характер потреб різних категорій клієнтів із певною поведінкою, наприклад, купують товари відомих дизайнерів або відвідують розпродажі. Ці знання потрібні для розробки точно спрямованих, економічних заходів щодо просування товарів.

Банківська справа. Досягнення технології Data Mining використовуються в банківській справі для вирішення таких поширених завдань:

• *виявлення шахрайства з кредитними картками.* Шляхом аналізу минулих транзакцій, які згодом виявилися шахрайськими, банк виявляє деякі стереотипи такого шахрайства;

• *сегментація клієнтів.* Розбиваючи клієнтів на різні категорії, банки роблять свою маркетингову політику більш цілеспрямованою і результативною, пропонуючи різні види послуг різним групам клієнтів;

• *прогнозування змін клієнтури.* Data Mining допомагає банкам будувати прогнозні моделі цінності своїх клієнтів, і відповідним чином обслуговувати кожну категорію.

Телекомунікації. В області телекомунікацій методи Data Mining допомагають компаніям більш енергійно просувати свої програми маркетингу і ціноутворення, щоб утримувати існуючих клієнтів і залучати нових. Серед типових заходів відзначимо такі:

• *аналіз записів про докладних характеристиках викликів.* Призначення такого аналізу – виявлення категорій клієнтів із схожими стереотипами користування їх послугами та розробка привабливих наборів цін і послуг;

• *виявлення лояльності клієнтів.* Data Mining можна використовувати для визначення характеристик клієнтів, які один раз скориставшись послугами даної компанії, з великою часткою ймовірності залишаться їй вірними. У підсумку

кошти, що виділяються на маркетинг, можна витрачати там, де віддача найбільша.

Страхування. Страхові компанії протягом декількох років накопичують великі обсяги даних. Тут широке поле діяльності для методів Data Mining:

• *виявлення шахрайства.* Страхові компанії можуть знизити рівень шахрайства, відшукуючи певні стереотипи в заявах про виплату страхового відшкодування, що характеризують відносини між юристами, лікарями та заявниками;

• *аналіз ризику*. Шляхом виявлення поєднань факторів, пов'язаних з оплаченими заявами, страховики можуть зменшити свої втрати за зобов'язаннями. Відомий випадок, коли в США велика страхова компанія виявила, що суми, виплачені за заявами одружених людей, вдвічі перевищують суми за заявами самотніх людей. Компанія відреагувала на це нове знання переглядом своєї загальної політики надання знижок сімейним клієнтам.

Інші області в бізнесі. Data Mining може застосовуватися в безлічі інших областей, зокрема таких, як:

• *розвиток автомобільної промисловості.* При виготовленні автомобілів виробники повинні враховувати вимоги кожного окремого клієнта, тому їм потрібні можливість прогнозування популярності певних характеристик і знання того, які характеристики зазвичай замовляються разом;

• *політика гарантій.* Виробникам потрібно передбачати число клієнтів, які подадуть гарантійні заявки, і середню вартість заявок;

• *заохочення часто літаючих клієнтів.* Авіакомпанії можуть виявити групу клієнтів, яких певними заохочувальними заходами можна спонукати літати більше. Наприклад, одна авіакомпанія виявила категорію клієнтів, які здійснювали багато перельотів на короткі відстані, що не накопичували достатню відстань для вступу в їхній дисконтний клуб, тому вона змінила правила прийому до клубу, щоб заохочувати число перельотів так само, як і накопичену відстань.

Медицина. Відомо багато експертних систем для постановки медичних діагнозів. Вони побудовані головним чином на основі правил, що описують поєднання різних симптомів різних захворювань. За допомогою таких правил дізнаються не тільки, на що хворий пацієнт, але й як потрібно його лікувати. Правила допомагають вибирати засоби медикаментозного впливу, визначати показання – протипоказання, орієнтуватися в лікувальних процедурах, створювати умови найбільш ефективного лікування, пророкувати результати призначеного курсу лікування тощо. Технології Data Mining дозволяють виявляти в медичних даних шаблони, що становлять основу зазначених правил.

Молекулярна генетика і генна інженерія. Мабуть, найбільш гостро і водночас чітко завдання виявлення закономірностей в експериментальних даних постає в молекулярній генетиці та генній інженерії. Тут воно формулюється як визначення так званих маркерів, під якими розуміють генетичні коди, контролюючі ті чи інші фенотипічні ознаки живого організму. Такі коди можуть містити сотні, тисячі і більше пов'язаних елементів.

Прикладна хімія. Методи Data Mining знаходять широке застосування в прикладній хімії (органічній та неорганічній). Тут нерідко виникає питання про з'ясування особливостей хімічної будови тих чи інших сполук, що визначають їх властивості. Особливо актуальна така задача при аналізі складних хімічних сполук, опис яких включає сотні і тисячі структурних елементів та їх зв'язків.

Можна навести ще багато прикладів різних областей знання, де методи Data Mining відіграють провідну роль. Особливість цих областей полягає в їх складній системній організації. Вони відносяться головним чином до надкібернетичного рівня організації систем, закономірності якого не можуть бути достатньо точно описані на мові статистичних чи інших аналітичних математичних моделей. Дані в зазначених сферах неоднорідні, гетерогенні, нестаціонарні і часто відрізняються високою розмірністю.

3. Типи закономірностей

Виділяють п'ять стандартних типів закономірностей, які дозволяють виявляти методи Data Mining: ***асоціація, послідовність, класифікація, кластеризація і прогнозування***.

Асоціація має місце в тому випадку, якщо кілька подій зв'язані одна з одною. Наприклад, дослідження, проведене в супермаркеті, може показати, що 65% тих, хто купив кукурудзяні чіпси, беруть також і «Кока-колу», а за наявності знижки за такий комплект «Колу» придбають у 85% випадків. Маючи в своєму розпорядженні відомості про подібну асоціацію, менеджерам легко оцінити, наскільки дієво надається знижка.

Якщо існує ланцюжок пов'язаних у часі подій, то говорять про **послідовність**. Так, наприклад, після покупки будинку в 45% випадків протягом місяця купується і нова кухонна плита, а в межах двох тижнів 60% новоселів вирішують придбати холодильник.

За допомогою **класифікації** виявляються ознаки, що характеризують групу, до якої належить той чи інший об'єкт. Це робиться за допомогою аналізу вже класифікованих об'єктів і формулювання деякого набору правил.

Кластеризація відрізняється від класифікації тим, що самі групи заздалегідь не задані. За допомогою кластеризації засобів Data Mining самостійно виділяють різні однорідні групи даних.

Основою для всіляких систем **прогнозування** служить історична інформація, що зберігається в БД у вигляді часових рядів. Якщо вдається побудувати шаблони, які адекватно відображають динаміку поведінки цільових показників, є ймовірність, що за їх допомогою можна передбачити і поведінку системи в майбутньому.

4. Класи систем Data Mining

Data Mining є мультидисциплінарною галуззю, яка виникла і розвивається на базі досягнень прикладної статистики, розпізнавання образів, методів штучного інтелекту, теорії баз даних тощо (рис. 1.1). Звідси велика кількість **методів і алгоритмів**, реалізованих у різних діючих системах Data Mining.

Багато з таких систем інтегрують у собі відразу кілька підходів. Проте, як правило, в кожній системі є якась ключова компонента (рис. 1.2).

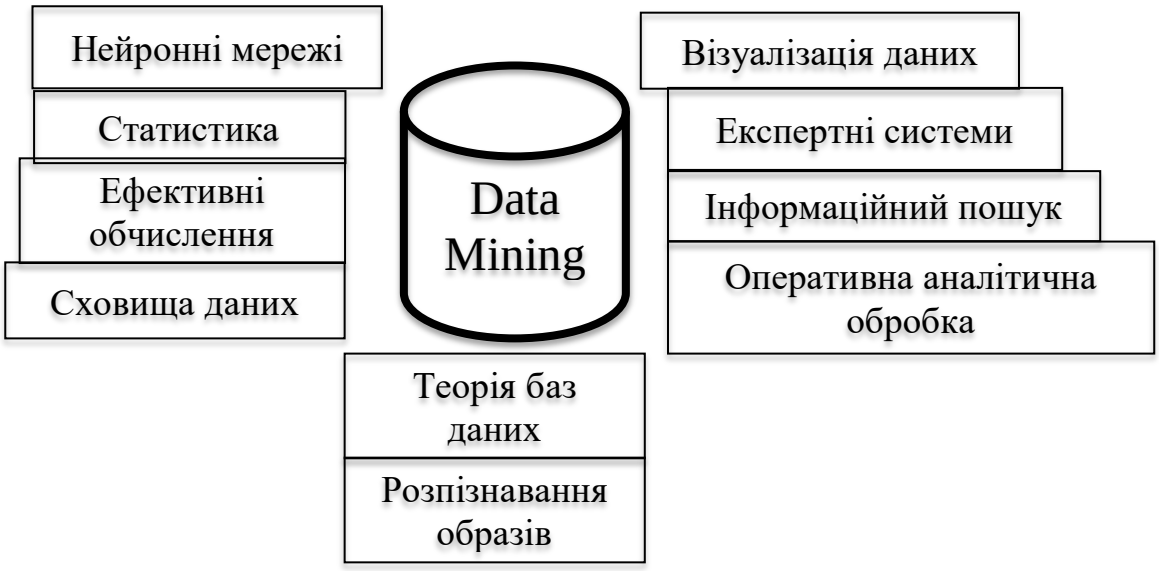


Рисунок 1.1 – Data Mining – мультидисциплінарна галузь

Предметно-орієнтовані аналітичні системи дуже різноманітні. Найбільш широкий підклас таких систем, що одержав поширення в галузі дослідження фінансових ринків, носить назву «Технічний аналіз». Він являє собою сукупність декількох десятків методів прогнозу динаміки цін і вибору оптимальної структури інвестиційного портфеля, заснованих на різних емпіричних моделях динаміки ринку. Ці методи часто використовують нескладний статистичний апарат, але максимально враховують сформовану у своїй сфері специфіку (професійна мова, системи різних індексів). На ринку є безліч програм цього класу. Як правило, вони досить дешеві (зазвичай коштують близько $300-1000).

Статистичні пакети. Останні версії майже всіх відомих статистичних пакетів включають поряд із традиційними статистичними методами також елементи Data Mining. Але основна увага в них приділяється все ж класичним методикам – кореляційному, регресійному, факторному аналізу та ін. Недоліком систем цього класу вважають вимогу до спеціальної підготовки користувача. Також відзначають, що потужні сучасні статистичні пакети є занадто «важкими» для масового застосування у фінансах і бізнесі. До того ж часто ці системи досить дорогі – від $1000 до $15000.

Рисунок 1.2 – Популярні продукти для Data Mining

Є ще більш серйозний принциповий недолік статистичних пакетів, що обмежує їх застосування в Data Mining. Більшість методів, що входять до складу пакетів спираються на статистичну парадигму, в якій головними фігурантами служать усереднені характеристики вибірки. А ці характеристики, як зазначалося вище, при дослідженні реальних складних життєвих феноменів часто є фіктивними величинами.

Як приклади найбільш потужних і поширених статистичних пакетів можна назвати SAS (компанія SAS Institute), SPSS (SPSS), STATGRAPICS (Manugistics), Statistica, STADIA та ін.

Нейронні мережі. Це великий клас систем, архітектура яких має аналогію (як тепер відомо, досить слабку) з побудовою нервової тканини з нейронів. В одній із найбільш поширених архітектурі зі зворотним поширенням помилки імітується робота нейронів у складі ієрархічної мережі, де кожен нейрон більш високого рівня з'єднаний своїми входами з виходами нейронів нижчого шару. На нейрони самого нижнього шару подаються значення вхідних параметрів, на основі яких потрібно приймати якісь рішення, прогнозувати розвиток ситуації тощо. Ці значення розглядаються як сигнали, що передаються в наступний шар, ослабляючись або посилюючись в залежності від числових значень (ваг), приписуваних дугами між нейронними зв'язками. У результаті на виході нейрона верхнього шару виробляється деяке значення, яке розглядається як відповідь – реакція всієї мережі на введені значення вхідних параметрів. Для того щоб мережу можна було застосовувати надалі, її треба «натренувати» на отриманих раніше даних, для яких відомі і значення вхідних параметрів, і правильні відповіді на них. Тренування полягає в підборі ваг, що забезпечують найбільшу близькість відповідей мережі до відомих правильних відповідей.

Основним недоліком нейронно-мережевої парадигми є необхідність мати дуже великий обсяг навчальної вибірки. Інший суттєвий недолік полягає в тому, що навіть натренована нейронна мережа являє собою чорний ящик. Знання, зафіксовані як ваги, абсолютно не піддаються аналізу та інтерпретації людиною (відомі спроби дати інтерпретацію структурі налаштованої нейронної мережі виглядають непереконливими – система «KINOsuite – PR»).

Приклади нейромережевих систем – це системи BrainMaker (CSS), NeuroShell (Ward Systems Group), OWL (HyperLogic) (рис. 1.3). Вартість їх досить значна: $1500-8000.

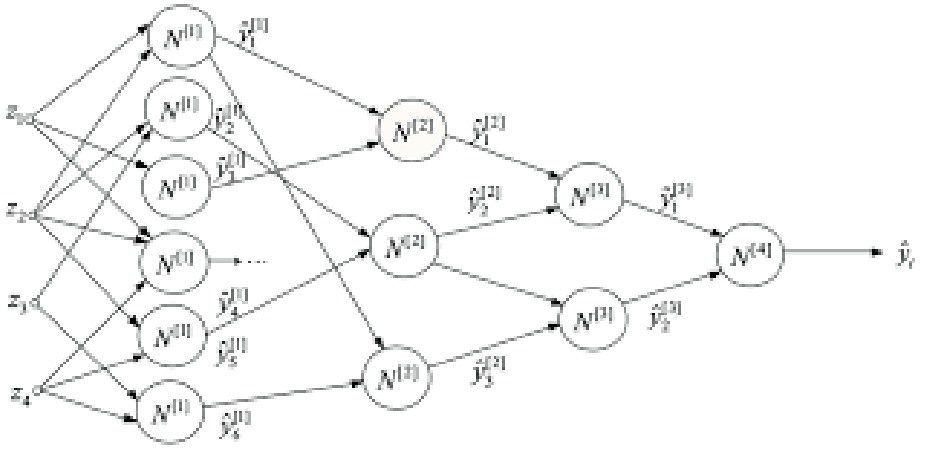


Рисунок 1.3 – Поліноміальна нейронна мережа

Системи міркувань на основі аналогічних випадків. Ідея систем case based reasoning – CBR – на перший погляд вкрай проста. Для того, щоб зробити прогноз на майбутнє чи вибрати правильне рішення ці системи знаходять у минулому близькі аналоги наявної ситуації і вибирають ту ж відповідь, який був для них правильним. Тому цей метод ще називають методом «найближчого сусіда» (nearest neighbour). Останнім часом поширення отримав також термін memory based reasoning, який акцентує увагу, що рішення приймається на підставі всієї інформації, накопиченої в пам'яті.

Системи CBR показують непогані результати в найрізноманітніших задачах. Головним їх мінусом вважають те, що вони взагалі не створюють будь-яких моделей або правил, узагальнюючих попередній досвід у виборі рішення вони ґрунтуються на всьому масиві доступних історичних даних, тому неможливо сказати, на основі яких конкретно факторів CBR системи будують свої відповіді.

Інший мінус полягає в свавіллі, який допускають системи CBR при виборі міри «близькості». Від цієї міри найрішучішим чином залежить обсяг безлічі прецедентів, які потрібно зберігати в пам'яті для досягнення задовільною класифікації або прогнозу.

Приклади систем, що використовують CBR – KATE tools (Acknosoft, Франція), Pattern Recognition Workbench (Unica, США).

Дерева рішень (decision trees) є одним з найбільш популярних підходів до вирішення завдань Data Mining. Вони створюють ієрархічну структуру правил типу «ЯКЩО... ТО...» (if – then), що має вигляд дерева. Для прийняття рішення, до якого класу віднести деякий об'єкт або ситуацію, потрібно відповісти на питання, що стоять у вузлах цього дерева, починаючи з його кореня. Запитання мають вигляд «значення параметра А більше х?». Якщо відповідь позитивна, здійснюється перехід до правого вузла наступного рівня, якщо негативна – то до лівого вузла; потім знову слід питання, пов'язане з відповідним вузлом.

Популярність підходу пов’язана як би з наочністю і зрозумілістю. Але дерева рішень принципово не здатні знаходити «кращі» (найбільш повні і точні) правила в даних. Вони реалізують наївний принцип послідовного перегляду ознак, створюючи лише ілюзію логічного висновку.

Разом із тим, більшість систем використовують саме цей метод. Найвідомішими є See5/C5.0 (RuleQuest, Австралія), Clementine (Integral Solutions, Великобританія), SIPINA (University of Lyon, Франція), IDIS (Information Discovery, США), KnowledgeSeeker (ANGOSS, Канада) (рис. 1.4). Вартість цих систем варіюється від 1 до 10 тис. дол.

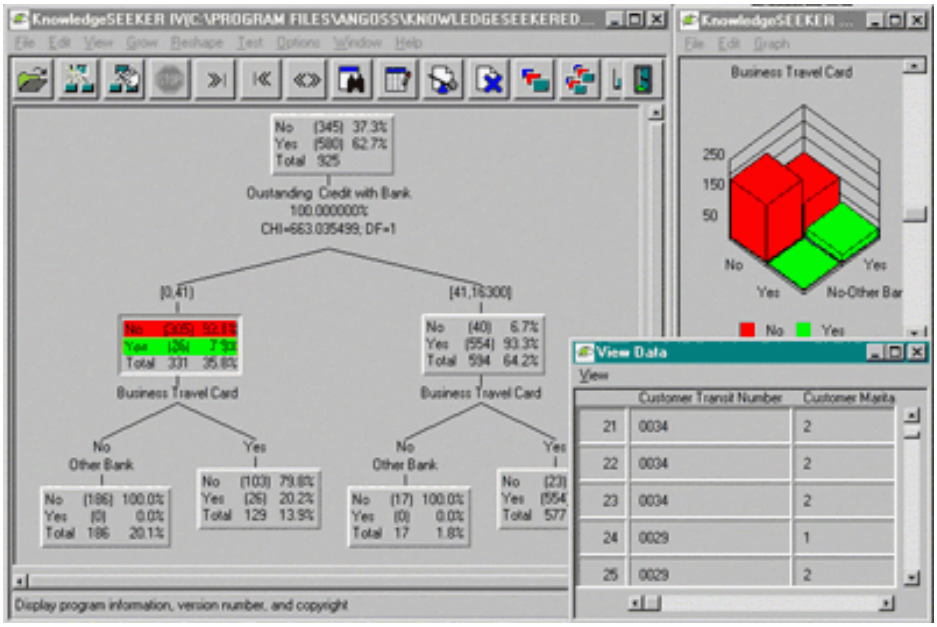


Рисунок 1.4 – Система KnowledgeSeeker обробляє банківську інформацію

Еволюційне програмування. Проілюструємо сучасний стан цього підходу на прикладі системи PolyAnalyst – вітчизняної розробки, що отримала сьогодні загальне визнання на ринку Data Mining. У даній системі гіпотези про вид залежності цільової змінної від інших змінних формулюються у вигляді програм на деякій внутрішній мові програмування. Процес побудови програм виглядає як еволюція у світі програм (цим підхід трохи схожий на генетичні алгоритми). Коли система знаходить програму, яка більш або менш задовільно відображає шукану залежність, вона починає вносити до неї невеликі модифікації та відбирає серед побудованих дочірніх програм ті, які підвищують точність. Таким чином система «вирощує» кілька генетичних ліній програм, які конкурують між собою в точності висловлювання шуканої залежності. Спеціальний модуль системи PolyAnalyst переводить знайдені залежності з внутрішньої мови системи на зрозумілу користувачеві мову (математичні формули, таблиці тощо).

Інший напрям еволюційного програмування пов’язаний з пошуком залежності цільових змінних від інших у формі функцій якогось певного виду. Наприклад, в одному з найбільш вдалих алгоритмів цього типу – метод групового урахування аргументів (МГУА) – залежність шукають у формі

поліномів. У цей час системи МГУА, що реалізовані в системі NeuroShell компанії Ward Systems Group, коштують до $500.

Генетичні алгоритми. Data Mining не основна сфера застосування генетичних алгоритмів. Їх потрібно розглядати скоріше як потужний засіб вирішення різноманітних комбінаторних завдань і завдань оптимізації. Проте генетичні алгоритми увійшли наразі в стандартний інструментарій методів Data Mining, тому вони і включені в цей огляд.

Перший крок при побудові генетичних алгоритмів – це кодування вихідних логічних закономірностей у базі даних, які іменують хромосомами, а весь набір таких закономірностей називають популяцією хромосом. Далі для реалізації концепції відбору вводиться спосіб зіставлення різних хромосом, який здійснюється за допомогою процедур репродукції, мінливості (мутацій), генетичної композиції. Ці процедури імітують біологічні процеси. Найбільш важливі серед них: випадкові мутації даних в індивідуальних хромосомах, переходи (кросинговер) і рекомбінація генетичного матеріалу, що міститься в індивідуальних батьківських, та міграції генів. У ході роботи процедур на кожній стадії еволюції виходять популяції з усе більш досконалими індивідуумами.

Генетичні алгоритми зручні тим, що їх легко розпаралелювати. Наприклад, можна розбити покоління на кілька груп і працювати з кожною з них незалежно, обмінюючись час від часу кількома хромосомами. Існують також і інші методи розпаралелювання генетичних алгоритмів.

Генетичні алгоритми мають ряд недоліків. Критерій відбору хромосом і використовувані процедури є евристичними і далеко не гарантують знаходження «кращого» рішення. Як і в реальному житті, еволюцію може «заклинити» на яку-небудь непродуктивну гілку. І, навпаки, можна навести приклади, як два неперспективних батька, які будуть виключені з еволюції генетичним алгоритмом, виявляються здатними призвести високоефективного нащадка. Це особливо стає помітно при вирішенні високо розмірних завдань зі складними внутрішніми зв’язками.

Прикладом реалізації генетичного алгоритму може служити система GeneHunter фірми Ward Systems Group. Її вартість складає близько $1000.

Алгоритми обмеженого перебору були запропоновані в середині 60-х років М.М. Бонгардом для пошуку логічних закономірностей у даних. Із того часу вони продемонстрували свою ефективність при вирішенні безлічі завдань із різноманітних сфер.

Ці алгоритми обчислюють частоти комбінацій простих логічних подій у підгрупах даних. Наведемо приклад простої логічної події:

$$X = a\ ;\ X < a\ ;\ X\ a\ ;\ a < X < b, \qquad (1.1)$$

де X – параметр,
«a» і «b» – константи.

Обмеженням служить довжина комбінації простих логічних подій (у М. Бонгарда вона дорівнювала 3). На підставі аналізу обчислених частот

робиться висновок про корисність тієї чи іншої комбінації для встановлення асоціації в даних, для класифікації, прогнозування.

Найбільш яскравим сучасним представником цього підходу є система WizWhy підприємства WizSoft (рис. 1.5). Хоча автор системи Абрахам Мейдан не розкриває специфіку алгоритму, покладеного в основу роботи WizWhy, за результатами ретельного тестування системи були зроблені висновки про наявність тут обмеженого перебору (вивчалися результати, залежність часу їх отримання від кількості аналізованих параметрів тощо).

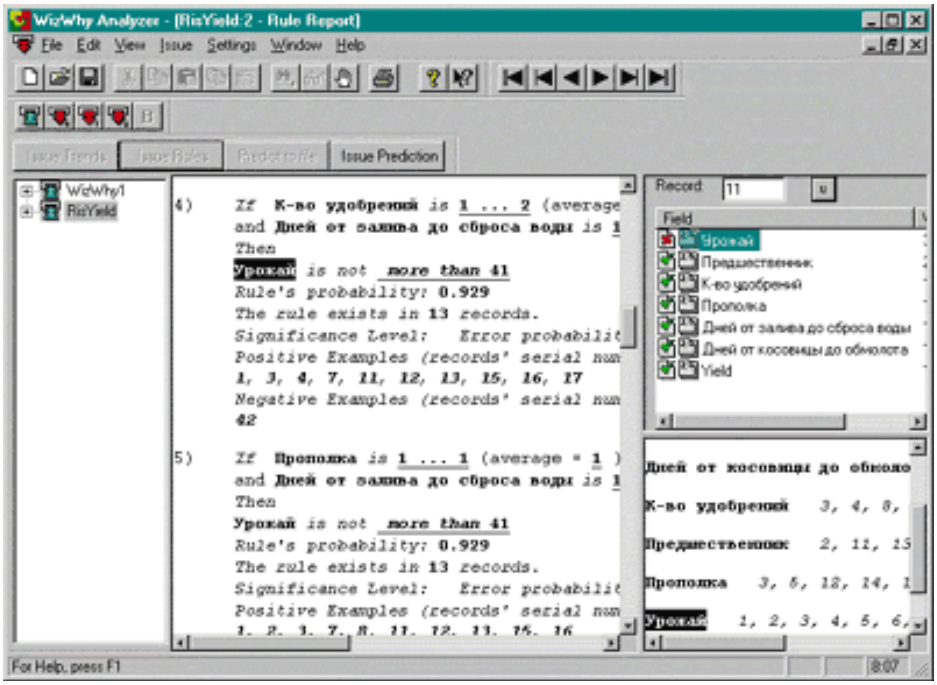


Рисунок 1.5 – Система WizWhy виявила правила, що пояснюють низьку врожайність деяких сільськогосподарських ділянок

Автор WizWhy стверджує, що його система виявляє всі логічні if – then 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 виробляється евристичний пошук простих логічних подій, на яких потім будується весь подальший аналіз. Зрозумівши ці особливості WizWhy, неважко було запропонувати найпростішу тестову задачу, яку система не змогла взагалі розв’язати. Інший недолік – система видає розв’язок за прийнятний час тільки для порівняно невеликої розмірності даних.

Проте, система WizWhy є на сьогодні одним із лідерів на ринку продуктів Data Mining. Це не позбавлене підстав. Система постійно демонструє більш високі показники при вирішенні практичних завдань, ніж всі інші алгоритми. Вартість системи складає близько $4000.

Системи для візуалізації багатовимірних даних. Тою чи іншою мірою засоби для графічного відображення даних підтримуються всіма системами Data Mining. Разом із тим, досить значну частку ринку займають системи, що спеціалізуються виключно на цій функції. Прикладом тут може служити програма DataMiner 3D (рис. 1.6) словацької фірми Dimension5 (5-й вимір).

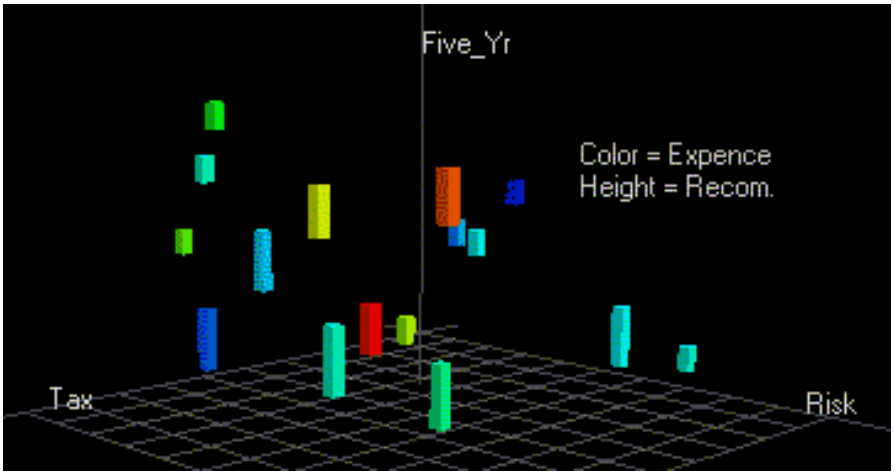


Рисунок 1.6 – Візуалізація даних системою DataMiner 3D

У подібних системах основну увагу сконцентровано на доброзичливості користувацького інтерфейсу, що дозволяє асоціювати з аналізованими показниками різні параметри діаграми розсіювання об'єктів (записів) бази даних. До таких параметрів належать колір, форма, орієнтація щодо власної осі, розміри та інші властивості графічних елементів зображення. Крім того, системи візуалізації даних забезпечені зручними засобами для масштабування і обертання зображень. Вартість систем візуалізації може досягати декількох сотень доларів.

Питання для самоконтролю

1. Навести кілька визначень поняття Data Mining.
2. Описати приклади застосування Data Mining в різних сферах економіки.
3. Які п'ять стандартних типів закономірностей, котрі реалізовані в методах Data Mining?
4. Який принцип дії алгоритма дерева рішень (decision trees).
5. Пояснити принцип дії алгоритму обмеженого перебору.
6. Пояснити різницю між методами еволюційного програмування та генетичних алгоритмів.

Тема 2. Поняття даних. Типи та формати зберігання даних. Бази даних. СУБД

План

1. Дані, набір даних та їх атрибути.
2. Формати зберігання даних.
3. Якісний аналіз даних із використанням Data Mining (DM).
4. Системи управління базами даних.

Мета вивчення теми: засвоїти поняття «дані» та особливості різних типів даних; вивчити етапи якісного процесу аналізу даних; засвоїти сутність систем управління базами даних.

Перелік ключових слів та понять із теми

Data Mining, дані, атрибути даних, змінна, шкала, зберігання даних, системи управління базами даних

Теоретичні відомості з теми

1. Дані, набір даних та їх атрибути

У широкому розумінні **дані – це факти, текст, графіки, картинки, звуки, аналогові або цифрові відео-сегменти**. Дані можуть бути отримані в результаті вимірювань, експериментів, арифметичних і логічних операцій. Дані повинні бути представлені у формі, придатній для зберігання, передачі й обробки. Іншими словами, дані – це необроблений матеріал, що надається постачальниками даних і використовується споживачами для формування інформації на основі даних.

У таблиці 2.1 представлена двовимірна таблиця, що представляє собою набір даних.

Таблиця 2.1 − Двовимірна таблиця «об’єкт-атрибут»

	Атрибути				
Об’єкти	Код клієнта	Вік	Сімейний статус	Прибуток	Клас
	1	19	неодр.	1234	1
	2	23	одр.	1222	1
	3	34	одр.	2700	1
	4	24	неодр.	2343	1
	5	26	одр.	1765	2
	6	32	розл.	2652	1
	7	19	неодр.	1200	2
	8	22	неодр.	1765	2
	9	40	одр.	1998	1
	10	43	розл.	4332	1

По горизонталі таблиці розташовуються атрибути об’єкта або його ознаки. По вертикалі таблиці – об’єкти. Об’єкт описується як набір атрибутів. Об’єкт також відомий як запис, випадок, приклад, рядок таблиці тощо.

Атрибут – властивість, що характеризує об’єкт. Наприклад: колір очей людини, температура води. Атрибут також називають змінною, полем таблиці, виміром, характеристикою.

Змінна (variable) – властивість або характеристика, загальна для всіх досліджуваних об’єктів, прояв якої може змінюватися від об’єкта до об’єкта.

Значення (value) змінної є проявом ознаки.

При аналізі даних, як правило, немає можливості розглянути всю сукупність об’єктів, що нас цікавить. Вивчення дуже великих обсягів даних є дорогим процесом, що вимагає великих затрат часу, а також неминуче призводить до помилок, пов’язаних із людським фактором.

Цілком достатньо розглянути деяку частину всієї сукупності, тобто вибірку, і отримати цікаву для нас інформацію на її підставі.

Однак розмір вибірки повинен залежати від різноманітності об’єктів, представлених у генеральній сукупності. У вибірці повинні бути представлені різні комбінації та елементи генеральної сукупності.

Генеральна сукупність (population) – вся сукупність досліджуваних об’єктів, що цікавить дослідника.

Вибірка (sample) – частина генеральної сукупності, певним способом відібрана з метою дослідження та отримання висновків про властивості та характеристики генеральної сукупності.

Параметри – числові характеристики генеральної сукупності.

Статистики – числові характеристики вибірки. Часто дослідження ґрунтуються на гіпотезах. Гіпотези перевіряються за допомогою даних.

Гіпотеза – припущення щодо параметрів сукупності об’єктів, яке має бути перевірено на її частині. Це частково обґрунтована закономірність знань, що служить або для зв’язку між різними емпіричними фактами, або для пояснення факту групи фактів.

Приклад гіпотези: між показниками тривалості життя та якістю харчування є зв’язок. У цьому випадку метою дослідження може бути пояснення змін конкретної змінної, в даному випадку – тривалості життя. Припустимо, існує гіпотеза, що залежна змінна (тривалість життя) змінюється залежно від деяких причин (якість харчування, спосіб життя, місце проживання тощо), які й є незалежними змінними.

Однак змінна першопочатково не є залежною або незалежною, вона стає такою після формулювання конкретної гіпотези. Залежна змінна в одній гіпотезі може бути незалежною в іншій.

Вимірювання – процес присвоєння чисел характеристикам досліджуваних об’єктів згідно певного правила.

У процесі підготовки даних вимірюється не сам об’єкт, а його характеристики.

Шкала – правило, відповідно до якого об’єктам присвоюються числа.

Багато інструментів Data Mining при імпорті даних з інших джерел пропонують вибрати тип шкали для кожної змінної та/або вибрати тип даних для вхідних і вихідних змінних (символьні, числові, дискретні та безперервні). Користувачеві такого інструменту необхідно володіти цими поняттями.

Змінні можуть бути числовими даними або символьними.

Числові дані, своєю чергою, можуть бути дискретними і неперервними.

Дискретні дані є значеннями ознаки, загальне число яких скінченне або нескінченне, але може бути підраховане за допомогою натуральних чисел від одного до нескінченності.

Прикладом дискретних даних є тривалість маршруту тролейбуса (кількість варіантів тривалості скінченне): 10, 15, 25 хв.

Неперервні дані – дані, значення яких можуть набувати якого завгодно значення в деякому інтервалі. Вимірювання неперервних даних передбачає велику точність.

Приклад неперервних даних: температура, висота, вага, довжина тощо.

Шкали. Існує п’ять типів шкал вимірювань: номінальна, порядкова, інтервальна, відносна і дихотомічна.

Номінальна шкала (nominal scale) – шкала, яка містить тільки категорії; дані в ній не можуть упорядковуватися, з ними не можуть бути зроблені ніякі арифметичні дії.

Номінальна шкала складається з назв, категорій, імен для класифікації і сортування об’єктів або спостережень за деякою ознакою.

Приклад такої шкали: професії, місто проживання, сімейний стан.

Для цієї шкали застосовні тільки такі операції: дорівнює (=), не дорівнює (≠).

Порядкова шкала (ordinal scale) – шкала, в якій числа присвоюють об’єктам для позначення відносної позиції об’єктів, але не величини відмінностей між ними.

Шкала вимірювань дає можливість ранжувати значення змінних. Вимірювання ж у порядковій шкалі містять інформацію лише про порядок проходження величин, але не дозволяють сказати наскільки одна величина більше іншої, або наскільки вона менше іншої.

Приклад такої шкали: місце (1-ше, 2-ге, 3-є), яке команда отримала на змаганнях, номер студента в рейтингу успішності (1-й, 23-й), при цьому невідомо, наскільки один студент успішніше іншого, відомий лише його номер у рейтингу.

Для цієї шкали застосовуються тільки такі операції: дорівнює (=), не дорівнює (≠), більше (>), менше (<).

Інтервальна шкала (interval scale) **–** шкала, різниці між значеннями якої можуть бути обчислені, проте їх відношення не мають сенсу.

Ця шкала дозволяє знаходити різницю між двома величинами, має властивості номінальної та порядкової шкал, а також дозволяє визначити кількісну зміну ознаки.

Приклад такої шкали: температура води в морі вранці – 19 градусів, ввечері – 24, тобто вечірня на 5 градусів вище, але не можна сказати, що вона в 1,26 разів вище.

Номінальна і порядкова шкали є дискретними, а інтервальна шкала – неперервною. Вона дозволяє здійснювати точні вимірювання ознаки і виробляти арифметичні операції додавання, віднімання, множення, ділення.

Для цієї шкали застосовуються тільки такі операції: дорівнює (=), не дорівнює (≠), більше (>), менше (<), операції додавання (+) і віднімання (-).

Відносна шкала (ratio scale) – шкала, в якій є певна точка відліку і можливі відносини між значеннями шкали.

Приклад такої шкали: вага новонародженої дитини (4 кг і 3 кг). Перша в 1,33 рази важче.

Ціна на картоплю в супермаркеті вище в 1,2 рази, ніж ціна на ринку.

Відносні та інтервальні шкали є числовими.

Для цієї шкали можуть бути застосовані тільки такі операції: дорівнює (=), не дорівнює (≠), більше (>), менше (<), операції додавання (+) і віднімання (-), множення (*) і ділення (/).

Дихотомічна шкала (dichotomous scale) – шкала, яка містить тільки дві категорії.

Приклад такої шкали: стать (чоловіча і жіноча).

Приклад використання різних шкал для вимірювань властивостей різних об’єктів, у даному випадку характеристик людей, наведено в таблиці 2.2.

Таблиця 2.2 − Множина вимірювань властивостей різних об’єктів

Номер об’єкту	**Професія (номінальна шкала)**	**Середній бал (інтервальна шкала)**	**Освіта (порядкова шкала)**
1	Слюсар	22	середня
2	Вчений	55	вища
3	Вчитель	47	вища

Приклад використання різних шкал для вимірювань властивостей однієї системи, у даному випадку температурних умов, наведено в таблиці 2.3.

Таблиця 2.3 − Множина вимірювань властивостей однієї системи

Дата змінення	**Хмарність (номінальна шкала)**	**Температура о 7 годині (інтервальна шкала)**	**Сила вітру (порядкова шкала)**
3 жовтня	Хмарно	22ºС	Сильний вітер
4 жовтня	Напівхмарно	17ºС	Слабий вітер
5 жовтня	Ясно	23ºС	Дуже сильний вітер

Типи наборів даних. Найбільш часто зустрічаються дані, що складаються із записів (record data).

Приклади таких наборів даних: табличні дані, матричні дані, документальні дані, транзакційні або операційні.

Табличні дані – дані, що складаються із записів, кожен з яких складається з фіксованого набору атрибутів.

Транзакційні дані представляють собою особливий тип даних, де кожен запис, що є транзакцією, включає набір значень.

Приклад транзакційної бази даних, що містить перелік покупок клієнтів магазину, наведено в таблиці 2.4.

Таблиця 2.4 − Приклад транзакційних даних

TID	Items
1	Bread, Coke, Milk
2	Beer, Bread
3	Beer, Coke, Diaper, Milk
4	Beer, Bread, Diaper, Milk
5	Coke, Diaper, Milk

Графічні дані. Приклади графічних даних: молекулярні структури; графи (рис. 2.1); карти.

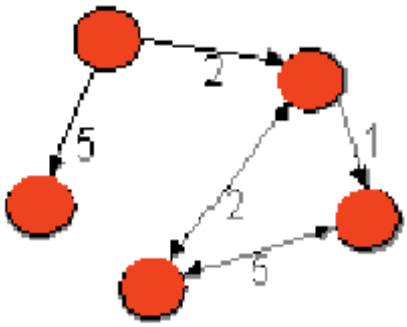


Рисунок 2.1 – Приклад графу

За допомогою карт, наприклад, можна відстежити зміни об’єктів у часі та просторі, визначити характер їх розподілу на площині або в просторі.

Перевагою графічного представлення даних є простота їх сприйняття у порівнянні, наприклад, з табличними даними.

Приклад карти, що є картою Кохонена (моделлю нейронних мереж), представлений на рис. 2.2.

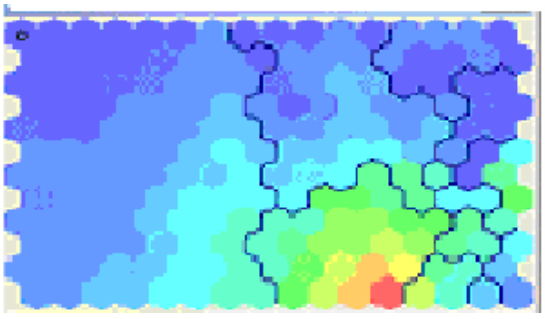


Рисунок 2.2 – Приклад даних типу «Карта Кохонена»

Хімічні дані представляють собою особливий тип даних. Приклад таких даних: молекула бензолу C_6H_6 (рис. 2.3).

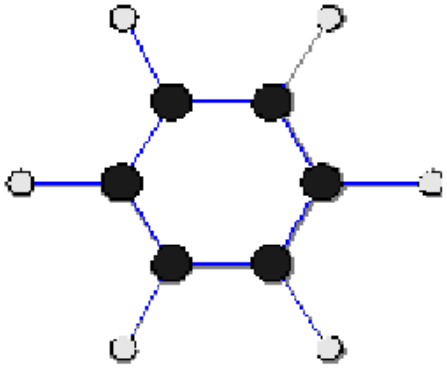


Рисунок 2.3 – Приклад хімічних даних

2. Формати зберігання даних

Одна з основних особливостей даних сучасного світу полягає в тому, що їх стає дуже багато.

Можливі чотири аспекти роботи з даними:

- визначення даних;
- обчислення;
- маніпулювання;
- обробка (збір, передача тощо).

При маніпулюванні даними використовується структура даних типу «файл». Файли можуть мати різні формати.

Більшість інструментів Data Mining дозволяють імпортувати дані з різних джерел, а також експортувати результуючі дані в різні формати.

Дані для експериментів зручно зберігати в якомусь одному форматі.

У деяких інструментах Data Mining ці процедури називаються імпорт / експорт даних, інші дозволяють напряму відкривати різні джерела даних і зберігати результати Data Mining в одному із запропонованих форматів.

Найбільш поширені формати зберігання даних представлені на рис. 2.4.

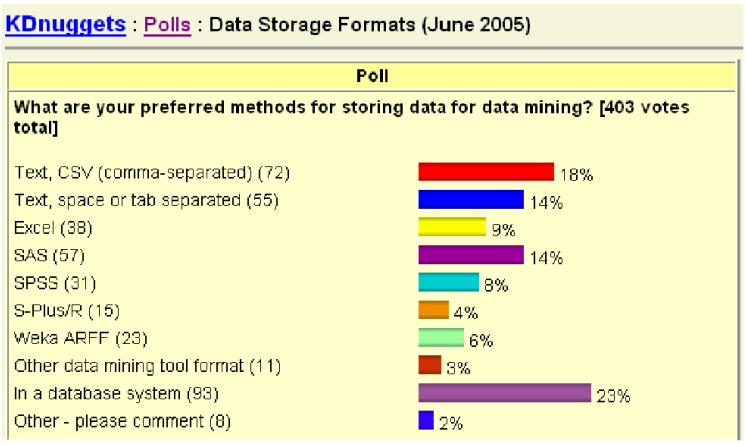


Рисунок 2.4 – Найбільш поширені формати зберігання даних

Найбільше число опитаних (23%) на он-лайн-ресурсі KDnuggets вважають за краще зберігати дані у форматі тієї бази даних, яку вони використовують. У

форматі Text, CSV – 18%, по 14% опитаних зберігають дані у форматі Text, space or tab separated і SAS; у форматі Excel – 9%, SPSS – 8%, S-Plus/R – 4%, Weka ARFF – 6%, в інших форматах інструментів Data Mining – 2%.

Як бачимо з результатів опитування, найбільш поширеним форматом зберігання даних для Data Mining виступають бази даних.

3. Якісний аналіз даних з використанням DM.

Для якісного аналізу будь-яких даних слід дотримуватися загальної схеми використання DM:

1. Висування гіпотез.
2. Збір та систематизація даних.
3. Підбір адекватної моделі.
4. Тестування та інтерпретація отриманих даних.
5. Використання в реальних умовах.

Ця схема не залежить від предметної області та сфери діяльності. Вона є універсальною.

1) Висування гіпотез

Гіпотезою тут будемо вважати припущення про вплив певних факторів на процес, що досліджується.

Автоматизувати процес висування гіпотез є вкрай складно, тому цю задачу мають розв’язувати експерти – фахівці в предметній області.

Слід довіритися їх досвіду та здоровому глузду, максимально використати ці знання про предмет досліджень і зібрати як найбільше гіпотез/припущень.

Зазвичай, добрі результати надають тактики «круглого столу» або «мозкової атаки». На початку слід зібрати та систематизувати всі ідеї, а оцінювати їх пізніше. У результаті повинен бути складений перелік з описів всіх факторів досліджуваного об’єкту.

Наприклад, для задачі прогнозування попиту товару потрібно скласти перелік факторів, що впливатимуть на об’єкт і експертно оцінити суттєвість кожного з них (табл. 2.5).

Таблиця 2.5 − Вплив кожного фактору (у %) на попит на товар

Сезон	100
День тижня	80
Обсяг продажів за попередні тижні	100
Обсяг продажів за аналогічний період минулого року	95
Рекламна компанія	60
Маркетингові заходи	40
Якість продукції	50
Бренд	25
Коливання ціни від середньоринкової	60
Наявність подібного товару в конкурентів	15

Згодом під час аналізу може з'ясуватися, що фактор, який експерти оцінили як важливий, буде мати незначний вплив на процес і навпаки.

2) Збір та систематизація даних

2.1. Збір даних

Для аналізу потрібно як найбільше даних, бо це надає можливість оцінити вплив максимальної кількості показників. Згодом простіше відхилити певну частину даних, аніж розпочинати новий збір.

Методи збору:

1. Отримання даних із внутрішніх джерел.

Це не складно, бо така інформація зазвичай зберігається в облікових системах у табличній формі, де існують різні механізми отримання звітів та експортування даних.

2. Отримання відомостей із непрямих даних.

Наприклад, потрібно оцінити реальний фінансовий стан мешканців певного регіону. Існує кілька категорій товару (зокрема, авто), що різняться за ціною – для незаможних, середнього класу, заможних. Якщо отримати звіт про продажі товару в цьому районі і проаналізувати пропорції, то дійдемо до висновку: чим більшим є відсоток продажів дорогого товару, тим заможнішими є мешканці.

3. Використання відкритих джерел.

До широкого загалу надаються статистичні збірники, звіти корпорацій, результати маркетингових досліджень, соціологічні опитування.

4. Влаштування власних маркетингових досліджень та подібних заходів по збору даних.

Це зазвичай є дорогим заходом, але доволі ефективним.

5. Наповнення даних згідно експертних оцінок співробітниками організації.

Слід оцінити вартість збору даних, що потрібні для аналізу. Одні дані беруться з публічних інформаційних джерел, інші мають бути оплачені, дані про діяльність конкурентів можуть бути доволі дорогими.

Вартість збору інформації різними методами суттєво різниться за ціною та витраченим часом, тому слід зважати на співвідношення теперішніх витрат із майбутніми результатами.

Від даних, які експерти вважають несуттєвими, певна річ можна відмовитися, але не від значущих даних, бо аналіз базуватиметься в цьому випадку на другорядних факторах і, відповідно, отримана модель буде надавати нестабільні та невірні результати.

4. Системи управління базами даних.

Не кожен блок інформації можна вважати базою даних. ***База даних*** *– це сукупність даних, яким властива структурованість і взаємопов'язаність, а також незалежність від прикладних програм.*

Пояснимо, що означають названі властивості бази даних. Щоб користувач легко міг знаходити потрібну інформацію, остання має бути організована певним

чином. Це стосується не лише інформації в комп'ютері, а й будь-якої інформації про об'єкти реального світу. Скажімо, зручно знаходити потрібну книгу в бібліотеці, користуючись каталогом. Легко відшукати в газеті оголошення, що вас цікавлять. Така легкість пошуку можлива завдяки тому, що дані в каталозі або в газеті мають структуру, або, інакше, *структуровані*. Усі книги описані однаково: автор, назва, видавництво, рік видання тощо. Усі оголошення з продажу розміщені по рубриках і також мають визначену структуру: короткий опис товару, ціна, телефон.

Структура бази даних складніша, ніж структура простого каталогу або набору газетних оголошень. Це зумовлено насамперед властивістю *взаємопов'язаності* даних у базі. Пояснимо це на такому прикладі: скажімо, ви хотіли б, крім каталожних карток, що описують кожну книгу, мати картки з інформацією про кожного автора (рік народження, літературний жанр, хобі тощо). Якби такі картки були створені, це був би приклад взаємозалежності даних: відомості про окрему книгу пов'язані з інформацією про автора. Цей зв'язок здійснюється через визначений параметр – прізвище автора.

Нарешті, остання з названих властивостей баз даних – це їхня *незалежність від прикладних програм.* Бази даних складаються таким чином, щоб із ними можна було працювати в різних програмних середовищах і на різних комп'ютерних платформах.

Щоб оперувати даними, які становлять базу, необхідна окрема програма – система управління базами даних. *Керівна програма, призначена для збереження, пошуку й обробки даних у базі, називається* ***системою управління базами даних*** *(скорочено СУБД).*

Система управління базами даних — це прикладна програма, реалізована на електронній обчислювальній машині чи обчислювальному комплексі. За допомогою її можна:

1) створювати структуру бази даних, вводити інформацію та зберігати її на зовнішніх носіях;
2) виконувати певне коло операцій із даними;
3) одержувати результати та зберігати їх на зовнішніх носіях або передавати на віддалені термінали;
4) виводити інформацію на термінал у зручній для користувача формі або на друкувальні пристрої;
5) давати можливість працювати з базами даних багатьом користувачам.

У цьому визначенні відсутній людський фактор – персонал, який відповідає за дані (адміністратор бази даних), але для розуміння роботи СУБД буде достатньо попереднього визначення.

Сучасні СУБД – це програмні додатки, які дозволяють виконувати різноманітні завдання. Усі існуючі системи задовольняють, як правило, таким вимогам:

✓ **можливості маніпулювання даними** (введення, вибір, вставка, відновлення, видалення тощо). Основні операції з даними виконуються під

керуванням СУБД. Важливими показниками є продуктивність СУБД, витрати на збереження і використання даних, простота звернення до бази даних тощо;

✓**можливість пошуку і формування запитів.** За допомогою запитів користувач може оперативно одержувати різну інформацію, що зберігається в базі даних.

✓**забезпечення цілісності (узгодженості) даних.** Під час використання даних багатьма користувачами важливо забезпечити коректність операцій, щоб запобігти порушенню узгодженості даних. Порушення узгодженості даних може призвести до їх невідновної втрати;

✓**забезпечення захисту і таємності.** Крім захисту від некоректних дій користувачів, важливо забезпечити захист даних від несанкціонованого доступу і від апаратних збоїв. Проникнення в базу осіб, які не мають на це права, може спричинити руйнацію даних. Таємність бази даних дозволяє визначити коло осіб, що мають доступ до інформації, і порядок доступу.

Сьогодні існує багато СУБД, що відрізняються архітектурою, внутрішньою мовою програмування, операційною системою, якою вони керуються, а також іншими характеристиками. Найпопулярнішими СУБД, що встановлюються в невеликих організаціях і орієнтовані на роботу з кінцевими користувачами, є Access, FoxPro, Paradox. До складніших систем належать розподілені СУБД, що призначені для роботи з великими базами даних, розподіленими на кількох серверах (сервери можуть міститися в різних регіонах). Потужними СУБД такого типу є Oracle, Sybase, Informix.

Вимоги до СУБД

СУБД разом із БД іноді називають банком даних. У банках даних повинні бути передбачені засоби, що забезпечують захист певних областей даних від несанкціонованого доступу.

Банк даних повинен відповідати таким вимогам:

1. Мати можливість оновлення, поповнення та розширення БД.
2. Забезпечити високу надійність зберігання інформації.
3. Видавати повну та вірогідну інформацію на запити.
4. Мати засоби, що забезпечують захист БД від несанкціонованого доступу.

Основні функції СУБД

До основних функцій СУБД належать такі:

1) опис БД (вказати назви полів, їх довжину, тип та інше);
2) введення в БД підготовлених даних;
3) перевірка правильності введення даних (контроль за типом);
4) редагування даних (вилучення, заміна, коректування, вставка, доповнення);
5) обробка запитів від користувачів (пошук певної інформації);
6) забезпечення одночасної роботи декількох користувачів з однією БД;
7) захист даних.

Питання для самоконтролю

1. Дайте визначення поняттю вибірка (sample).
2. Дайте визначення поняттю гіпотеза.
3. Наведіть приклади застосування гіпотез.
4. Які види шкал ви знаєте? Чим вони відрізняються?
5. Які типи даних ви знаєте?
6. З яких етапів складається загальна схема використання DM?

Тема 3. Метадані. Класифікація метаданих

План

1. Поняття метаданих.
2. Класифікація метаданих.
3. Формат метаданих.

Мета вивчення теми: засвоїти поняття «метадані» та особливості роботи з ними.

Перелік ключових слів та понять із теми

Метадані, метадані за змістом, метадані до ресурсу, формат метаданих

Теоретичні відомості з теми

1. Поняття метаданих

Метадані (у загальному випадку) – це дані, що характеризують або пояснюють інші дані. Наприклад, значення «123456» само по собі недостатньо виразне. А якщо значенню «123456» зіставлено ім'я «поштовий індекс» (що вже є метаданими), то в цьому контексті значення «123456» більш осмислене – можна «витягувати» інформацію про місцеположення адресата, що має даний поштовий індекс.

Оскільки для більшості людей різниця між словами «дані» та «інформація» існує тільки з філософської точки зору і не істотна з практичної точки зору, то мають місце такі визначення:

- метадані – це інформація про дані;
- метадані – це інформація про інформацію.

Для терміну **метадані** немає єдиного формального визначення. Навпаки, існують різні визначення цього терміну. Ось просте і популярне переформулювання:

Метадані – це дані про дані. Цей термін в широкому сенсі слова використовується для будь-яких «даних про дані»: іменах таблиць, колонок в таблиці, програм і тому подібне.

Метадані – це дані з більш загальної формальної системи, що описує задану систему даних.

Існують **вужчі визначення**:

Метадані – це структуровані дані, що представляють собою характеристики описуваних сутностей для цілей їх ідентифікації, пошуку, оцінки, управління ними.

Метадані – це набір допустимих структурованих описів, які доступні в явному вигляді і призначення яких допомогти знайти об'єкт. Це визначення використовується набагато рідше, оскільки воно концентрується на одному з

призначень метаданих – пошук об’єктів, сутностей, ресурсів – та ігнорує інші призначення.

Відмінність між даними і метаданими. Зазвичай неможливо провести однозначне розділення на дані та **метадані** в документі, оскільки:

1. Щось може бути як даними, так і метаданими. Так, заголовок статті можна одночасно віднести як до метаданих (як елемент метаданих – заголовок), так і до власне даних (оскільки заголовок є частиною самого тексту).
2. Дані та метадані можуть мінятися ролями. На вірш, що розглядається як дані, може бути написана музика, в цьому випадку весь вірш може бути «прикріплений» до музичного файлу і в цьому випадку розглядається як метадані. Отже, віднесення до однієї або іншої категорії залежить від точки зору.
3. Можливе створення мета-мета-…-метаданих. Оскільки, відповідно до звичайного визначення, метадані є даними, то можна створити метадані на метадані, метадані на метадані на метадані і т.д. На перший погляд це може здатися безглуздим, але насправді це є дуже істотною і корисною властивістю даних і метаданих.

Ці міркування застосовні незалежно від вибору визначення метаданих (з приведених вище і не тільки).

Метадані використовуються для підвищення якості пошуку. Пошукові запити, використовуючи метадані, можуть врятувати користувача від зайвої ручної роботи з фільтрації. Інформуючи комп’ютер про те, які елементи даних зв’язані і як ці зв’язки враховувати, стає можливим здійснювати достатньо складні операції з фільтрації та пошуку. Наприклад, якщо пошукова система «знає» про те, що «Ван Гог» є «голландським художником», то вона може видати у відповідь на запит про голландських художників веб-сторінку про Ван Гога, навіть якщо слова «голландський художник» не зустрічаються на цій сторінці.

Практично кожний електронний документ має певні метадані. Метадані електронних документів відіграють важливу роль у системах електронного документообігу та автоматизації діловодства, в інформаційно-пошукових системах. Метадані можуть, наприклад, включати дату, коли документ був збережений і відомості про особистість користувача, що зберіг його. Системи електронного документообігу та автоматизації діловодства можуть також здобувати метадані з документу автоматично або підказувати користувачеві додати метадані.

2. Класифікація метаданих

Метадані можна класифікувати за такими ознаками:

- *Змістом*. Метадані можуть або описувати сам ресурс (наприклад, назва і розмір файлу), або вміст ресурсу (наприклад, «у цьому відеофайлі показано, як хлопець грає у футбол»).
- *Відношенням до ресурсу в цілому*. Метадані можуть відноситься до ресурсу в цілому або до його частин. Наприклад, «Title» (назва фільму) відноситься до фільму в цілому, а «Scene description» (опис епізоду фільму) окреме для кожного епізоду фільму.

- *Можливістю логічного виводу*. Метадані можна підрозділити на три шари: нижній шар – це «сирі» дані самі по собі; середній шар – метадані, що описують ці дані; і верхній шар – метадані, які дозволяють робити логічний вивід, використовуючи другий шар.

3. Формат метаданих

Метаданими на практиці зазвичай називають дані, представлені відповідно до одного з форматів метаданих.

Формат метаданих – це стандарт, призначений для формального опису деякої категорії ресурсів (об'єктів, сутностей, документів і т.п.). Такий стандарт зазвичай включає набір полів (атрибутів, властивостей, елементів метаданих), що дозволяють характеризувати даний об'єкт. Наприклад, формат MARC дозволяє описувати книги (і не тільки книги), містить поля для опису назви, автора, тематики і безлічі інших характеристик (формат MARC дозволяє описати сотні характеристик).

Формати метаданих часто розробляються міжнародними організаціями або консорціумами, що включають зацікавлені в появі стандарту державні організації та приватні компанії. Розроблений формат часто закріплюється як стандарт в одній або декількох організаціях, що займаються розробкою й ухваленням стандартів (наприклад W3C, ISO, ANSI тощо).

Питання для самоконтролю

1. Дайте визначення поняттю метадані.
2. Які ви знаєте відмінності між даними і метаданими?
3. Для чого використовуються метадані?
4. Наведіть класифікацію метаданих.
5. Дайте визначення формату метаданих.

Тема 4. Етапи ІАД. Класифікація методів ІАД

План

1. Класифікація стадій Data Mining.
2. Класифікація технологічних методів Data Mining.
3. Властивості методів Data Mining.

Мета вивчення теми: вивчити стадії та методи Data Mining; засвоїти властивості методів інтелектуального аналізу даних.

Перелік ключових слів та понять із теми

Data Mining, метод, штучні нейронні мережі, дерева рішень, символьні правила, методи найближчого сусіда і k-найближчого сусіда, метод опорних векторів, байєсовські мережі, лінійна регресія, кореляційно-регресійний аналіз, ієрархічні методи кластерного аналізу, неієрархічні методи кластерного аналізу

Теоретичні відомості з теми

1. Класифікація стадій Data Mining

Основна особливість Data Mining – це поєднання широкого математичного інструментарію (від класичного статистичного аналізу до нових кібернетичних методів) і останніх досягнень у сфері інформаційних технологій. У технології Data Mining гармонійно об'єдналися строго формалізовані методи і методи неформального аналізу, тобто кількісний та якісний аналіз даних.

До методів і алгоритмів Data Mining належать такі: штучні нейронні мережі, дерева рішень, символьні правила, методи найближчого сусіда і *k*-найближчого сусіда, метод опорних векторів, байєсовські мережі, лінійна регресія, кореляційно-регресійний аналіз; ієрархічні методи кластерного аналізу, неієрархічні методи кластерного аналізу, в тому числі алгоритми *k*-середніх і *k*-медіани; методи пошуку асоціативних правил, у тому числі алгоритм Apriori; метод обмеженого перебору, еволюційне програмування і генетичні алгоритми, різноманітні методи візуалізації даних і безліч інших методів.

Більшість аналітичних методів, що використовуються в технології Data Mining – це відомі математичні алгоритми і методи. Новим в їх застосуванні є можливість їх використання при розв'язуванні тих чи інших конкретних задач, зумовлена можливостями нових технічних і програмних засобів. Слід зазначити, що більшість методів Data Mining були розроблені в рамках теорії штучного інтелекту.

Метод (method) являє собою норму або правило, певний шлях, спосіб, прийом розв'язання задачі теоретичного, практичного, пізнавального, управлінського характеру.

Поняття алгоритму з'явилося задовго до створення електронних обчислювальних машин. Зараз алгоритми є основою для вирішення багатьох

прикладних і теоретичних завдань у різних сферах людської діяльності, у більшості – це завдання, вирішення яких передбачено з використанням комп'ютера.

Алгоритм (algorithm) – точний припис щодо послідовності дій (кроків), що перетворюють вихідні дані в шуканий результат.

Data Mining може складатися з двох або трьох стадій:

Стадія 1. Виявлення закономірностей (**вільний пошук**).

Стадія 2. Використання виявлених закономірностей для передбачення невідомих значень (**прогностичне моделювання**).

На додаток до цих стадій іноді вводять **стадію валідації**, наступну за стадією вільного пошуку. **Мета валідації** – перевірка достовірності знайдених закономірностей. Однак ми будемо вважати валідацію частиною першої стадії, оскільки в реалізації багатьох методів, зокрема, нейронних мереж і дерев рішень, передбачено поділ загальної множини даних на навчальну і перевірочну, і останнє дозволяє перевіряти достовірність отриманих результатів.

Стадія 3. Аналіз винятків – стадія призначена для виявлення і пояснення аномалій, знайдених у закономірностях.

Отже, процес Data Mining може бути представлений низкою таких послідовних стадій:

ВІЛЬНИЙ ПОШУК (у тому числі ВАЛІДАЦІЯ) ->

-> **ПРОГНОСТИЧНЕ МОДЕЛЮВАННЯ** ->

-> **АНАЛІЗ ВИНЯТКІВ**

1. Вільний пошук (Discovery).

На стадії вільного пошуку здійснюється дослідження набору даних із метою пошуку прихованих закономірностей. Попередні гіпотези щодо виду закономірностей тут не визначаються.

Закономірність (law) – істотний і постійно повторюваний взаємозв'язок, що визначає етапи і форми процесу становлення, розвитку різних явищ або процесів. Система Data Mining на цій стадії визначає шаблони, для отримання яких в системах **OLAP**, наприклад, аналітик повинен обдумувати і створювати безліч запитів. Тут же аналітик звільняється від такої роботи – шаблони шукає за нього система. Особливо корисне застосування даного підходу в надвеликих базах даних, де визначити закономірність шляхом створення запитів досить складно, для цього потрібно перепробувати безліч різноманітних варіантів.

Вільний пошук представлений такими **діями**:

• виявлення закономірностей умовної логіки (conditional logic);

• виявлення закономірностей асоціативної логіки (associations and affinities);

• виявлення трендів і коливань (trends and variations).

Припустимо, є база даних кадрового агенства з даними про професії, стаж, вік і бажаний рівень винагороди. У разі самостійного задавання запитів аналітик може отримати приблизно такі результати: середній бажаний рівень винагороди фахівців у віці від 25 до 35 років дорівнює 1200 умовних одиниць. У разі вільного пошуку система сама шукає закономірності, необхідно лише

задати цільову змінну. У результаті пошуку закономірностей система сформує набір логічних правил " якщо..., то...".

Можуть бути знайдені, наприклад, такі закономірності

"Якщо вік <20 років і бажаний рівень винагороди >700 умовних одиниць, то в 75% випадків здобувач шукає роботу програміста"

або

"Якщо вік >35 років і бажаний рівень винагороди >1200 умовних одиниць, то в 90% випадків здобувач шукає роботу керівника". Цільовою змінною в описаних правилах виступає професія.

Задавши іншу цільову змінну, наприклад, вік, отримуємо такі правила: "Якщо здобувач шукає керівну роботу і його стаж >15 років, то вік здобувача >35 років у 65% випадків".

Описані дії, в рамках стадії вільного пошуку, виконуються за допомогою:

• індукції правил умовної логіки (задачі класифікації та кластеризації, опис у компактній формі близьких або схожих груп об'єктів);

• індукції правил асоціативної логіки (задачі асоціації та послідовності й одержувана за їх допомогою інформація);

• визначення трендів і коливань (вихідний етап задачі прогнозування).

На стадії вільного пошуку також повинна здійснюватися валідація закономірностей, тобто перевірка їх достовірності на частині даних, які не брали участь у формуванні закономірностей. Такий прийом розділення даних на навчальну і перевірочну множину часто використовується в методах нейронних мереж і дерев рішень.

2. Прогностичне моделювання (Predictive Modeling).

Друга стадія Data Mining – прогностичне моделювання – використовує результати роботи першої стадії. Тут виявлені закономірності використовуються безпосередньо для прогнозування.

Прогностичне моделювання включає такі дії:

• передбачення невідомих значень (outcome prediction);

• прогнозування розвитку процесів (forecasting).

У процесі прогностичного моделювання розв'язуються задачі класифікації та прогнозування.

При розв'язанні задачі класифікації результати роботи першої стадії (індукції правил) використовуються для віднесення нового об'єкта, з певною впевненістю, до одного з відомих, визначених класів на підставі відомих значень.

При розв'язанні задачі прогнозування результати першої стадії (визначення тренда або коливань) використовуються для передбачення невідомих (пропущених або ж майбутніх) значень цільової змінної (змінних).

Продовжуючи розглянутий приклад першої стадії, можемо зробити такий висновок.

Знаючи, що здобувач шукає керівну роботу і його стаж >15 років, на 65% можна бути впевненим у тому, що вік здобувача >35 років. Або ж, якщо вік здобувача >35 років і бажаний рівень винагороди >1200 умовних одиниць, на 90% можна бути впевненим у тому, що здобувач шукає керівну роботу.

Порівняємо вільний пошук і прогностичне моделювання з точки зору логіки.

Вільний пошук розкриває загальні закономірності. Він за своєю природою індуктивний. Закономірності, отримані на цій стадії, формуються від особистого до загального. У результаті ми отримуємо деяке загальне знання про деякий клас об'єктів на підставі дослідження окремих представників цього класу.

Правило: "Якщо вік здобувача <20 років і бажаний рівень винагороди >700 умовних одиниць, то в 75% випадків здобувач шукає роботу програміста". На підставі особистого, тобто інформації про деякі властивості класу "вік <20 років" і "бажаний рівень винагороди >700 умовних одиниць", робимо висновок про загальне, а саме: шукають роботу – програмісти.

Прогностичне моделювання, навпаки, дедуктивне. Закономірності, отримані на цій стадії, формуються від загального до особистого і одиничного. Тут ми отримуємо нове знання про деякий об'єкт або ж групи об'єктів на підставі:

• знання класу, до якого належать досліджувані об'єкти;

• знання загального правила, діючого в межах даного класу об'єктів.

Знаємо, що претендент шукає керівну роботу і його стаж >15 років, на 65% можна бути впевненим у тому, що вік здобувача >35 років.

На підставі деяких загальних правил, а саме: мета здобувача – керівна робота і його стаж >15 років, ми робимо висновок про одиничне – вік здобувача >35 років.

Слід зазначити, що отримані закономірності, а точніше, їх конструкції, можуть бути прозорими, тобто допускають тлумачення аналітика (розглянуті вище правила), і непрозорими, так званими «чорними ящиками». Типовий приклад останньої конструкції – нейронна мережа.

3. Аналіз винятків (forensic analysis).

На третій стадії Data Mining аналізуються виключення або аномалії, виявлені в знайдених закономірностях.

Дія, що виконується на цій стадії, це виявлення відхилень (deviation detection). Для виявлення відхилень необхідно визначити норму, яка розраховується на стадії вільного пошуку.

Повернемося до одного з прикладів, розглянутих вище.

Знайдено правило "Якщо вік >35 років і бажаний рівень винагороди >1200 умовних одиниць, то в 90% випадків здобувач шукає керівну роботу". Виникає питання – до чого віднести решту 10% випадків? Тут можливі два варіанти. Перший з них – існує деяке логічне пояснення, яке також може бути оформлено у вигляді правила. Другий варіант для решти 10% – це помилки вихідних даних. У цьому випадку стадія аналізу винятків може бути використана для очищення даних.

Далі ми розглянемо кілька відомих класифікацій методів Data Mining за різними ознаками.

2. Класифікація технологічних методів Data Mining

Усі методи Data Mining поділяються на дві великі групи за принципом роботи з вихідними навчальними даними. У цій класифікації верхній рівень визначається на підставі того, зберігаються дані після Data Mining чи вони дистилюються для подальшого використання.

1. Безпосереднє використання даних, або збереження даних.

У цьому випадку вихідні дані зберігаються в явному деталізованому вигляді і безпосередньо використовуються на стадіях прогностичного моделювання та/або аналізу винятків. Проблема цієї групи методів – при їх використанні можуть виникнути складності аналізу надвеликих баз даних.

Методи цієї групи: **кластерний аналіз, метод найближчого сусіда, метод *k*-найближчого сусіда, міркування за аналогією.**

2. Виявлення і використання формалізованих закономірностей, або дистиляція шаблонів.

При технології дистиляції шаблонів один зразок (шаблон) інформації витягується з вихідних даних і перетворюється в якісь формальні конструкції, вигляд яких залежить від використовуваного методу Data Mining. Цей процес виконується на стадії вільного пошуку, у першій же групі методів ця стадія в принципі відсутня. На стадіях прогностичного моделювання та аналізу винятків використовуються результати стадії вільного пошуку, вони значно компактніше самих баз даних. Нагадаємо, що конструкції цих моделей можуть бути трактовані аналітиком або не трактовані («чорні ящики»).

Методи цієї групи: **логічні методи, методи візуалізації; методи крос-табуляції; методи, засновані на рівняннях.**

Логічні методи, або методи логічної індукції, включають:

- нечіткі запити і аналізи;
- символьні правила;
- дерева рішень;
- генетичні алгоритми.

Методи цієї групи є такими, що найкраще інтерпретуються – вони оформляють знайдені закономірності, в більшості випадків у досить прозорому вигляді з точки зору користувача. Отримані правила можуть включати безперервні і дискретні змінні. Слід зауважити, що дерева рішень можуть бути легко перетворені в набори символьних правил шляхом генерації одного правила по шляху від кореня дерева до його термінальної вершини. Дерева рішень і правила фактично є різними способами розв'язання однієї задачі і відрізняються лише за своїми можливостями. Крім того, реалізація правил здійснюється більш повільними алгоритмами, ніж індукція дерев рішень.

Методи крос-табуляції: агенти, байєсовські (довірчі) мережі, крос-таблична візуалізація. Останній метод не зовсім відповідає одній з властивостей Data Mining – самостійного пошуку закономірностей аналітичною системою. Однак надання інформації у вигляді крос-таблиць забезпечує реалізацію основного завдання Data Mining – пошук шаблонів, тому цей метод можна також вважати одним із методів Data Mining.

Методи на основі рівнянь. Методи цієї групи висловлюють виявлені закономірності у вигляді математичних виразів – рівнянь. Отже, вони можуть

працювати лише з чисельними змінними, і змінні інших типів повинні бути закодовані відповідним чином. Це дещо обмежує застосування методів цієї групи, проте вони широко використовуються при вирішенні різних завдань, особливо завдань прогнозування.

Основні методи цієї групи: **статистичні методи і нейронні мережі.**

Статистичні методи найбільш часто застосовуються для розв'язання задач прогнозування. Існує безліч методів статистичного аналізу даних, серед них, наприклад, кореляційно-регресійний аналіз, кореляція рядів динаміки, виявлення тенденцій динамічних рядів, гармонійний аналіз.

Інша класифікація поділяє все різноманіття методів Data Mining на дві групи: **статистичні** та **кібернетичні** методи. Ця схема поділу заснована на різних підходах до навчання математичних моделей.

Слід зазначити, що існує два підходи віднесення статистичних методів до Data Mining. Перший з них протиставляє статистичні методи і Data Mining, його прихильники вважають класичні статистичні методи окремим напрямом аналізу даних. Відповідно до другого підходу, статистичні методи аналізу є частиною математичного інструментарію Data Mining. Більшість авторитетних джерел дотримується другого підходу.

У цій класифікації розрізняють дві групи методів:

• *статистичні методи*, засновані на використанні усередненого накопиченого досвіду, який відображений у ретроспективних даних;

• *кібернетичні методи*, що включають безліч різнорідних математичних підходів.

Недолік такої класифікації: і статистичні, і кібернетичні алгоритми тим чи іншим чином спираються на зіставлення статистичного досвіду з результатами моніторингу поточної ситуації.

Перевагою такої класифікації є її зручність для інтерпретації – вона використовується при описі математичних засобів сучасного підходу до вилучення знань із масивів вихідних спостережень (оперативних і ретроспективних), тобто в задачах Data Mining.

Статистичні методи Data mining. Ці методи являють собою чотири взаємопов'язаних розділи:

• попередній аналіз природи статистичних даних (перевірка гіпотез стаціонарності, нормальності, незалежності, однорідності, оцінка виду функції розподілу, її параметрів тощо);

• виявлення зв'язків і закономірностей (лінійний і нелінійний регресійний аналіз, кореляційний аналіз та ін.);

• багатовимірний статистичний аналіз (лінійний і нелінійний дискримінантний аналіз, кластерний аналіз, компонентний аналіз, факторний аналіз та ін.);

• динамічні моделі і прогноз на основі часових рядів.

Кібернетичні методи Data Mining. Інший напрямок Data Mining – це безліч підходів, об'єднаних ідеєю комп'ютерної математики та використання теорії штучного інтелекту.

До цієї групи відносяться такі методи:

• штучні нейронні мережі (розпізнавання, кластеризація, прогнозування);

• еволюційне програмування (у т.ч. алгоритми методу групового обліку аргументів);

• генетичні алгоритми (оптимізація);

• асоціативна пам'ять (пошук аналогів, прототипів);

• нечітка логіка;

• дерева рішень;

• системи обробки експертних знань.

Методи Data Mining також можна класифікувати за задачами Data Mining. Відповідно до такої класифікації виділяємо дві групи. Перша з них – це поділ методів Data Mining на вирішальні завдання сегментації (тобто задачі класифікації та кластеризації) і завдання прогнозування.

У відповідності до другої класифікації за задачами методи Data Mining можуть бути спрямовані на отримання описових і прогнозуючих результатів.

Описові методи служать для знаходження шаблонів або зразків, що описують дані, які піддаються інтерпретації з точки зору аналітика.

До методів, спрямованих на отримання описових результатів, відносяться ітеративні методи кластерного аналізу, в тому числі: алгоритм *k*-середніх, *k*-медіани, ієрархічні методи кластерного аналізу, карти Кохонена, методи крос-табличної візуалізації, різні методи візуалізації та ін.

Прогнозуючі методи використовують значення одних змінних для передбачення / прогнозування невідомих (пропущених) або майбутніх значень інших (цільових) змінних.

До методів, спрямованих на отримання прогнозуючих результатів, відносяться такі методи: нейронні мережі, дерева рішень, лінійна регресія, метод найближчого сусіда, метод опорних векторів тощо.

3. Властивості методів Data Mining

Різні методи Data Mining характеризуються певними властивостями, які можуть бути визначальними при виборі методу аналізу даних. Методи можна порівнювати між собою, оцінюючи характеристики їх властивостей.

Серед основних властивостей і характеристик методів Data Mining розглянемо такі: точність, масштабованість, інтерпретованість, здатність до перевірки, трудомісткість, гнучкість, швидкість і популярність.

Масштабованість – властивість обчислювальної системи, яка забезпечує передбачуваний зріст системних характеристик, наприклад, швидкості реакції, загальної продуктивності тощо, при додаванні до неї обчислювальних ресурсів.

Більшість інструментів Data Mining, пропонованих зараз на ринку програмного забезпечення, реалізують відразу кілька методів, наприклад, дерева рішень, індукцію правил і візуалізацію, або ж нейронні мережі, карти Кохонена та візуалізацію. В універсальних прикладних статистичних пакетах (наприклад, SPSS, SAS, STATGRAPHICS, Statistica) реалізується широкий спектр найрізноманітніших методів (як статистичних, так і кібернетичних). Слід

враховувати, що для можливості їх використання, а також для інтерпретації результатів роботи статистичних методів (кореляційного, регресійного, факторного, дисперсійного аналізу) потрібні спеціальні знання в галузі статистики.

Універсальність того чи іншого інструмента часто накладає певні обмеження на його можливості. Перевагою використання таких універсальних пакетів є можливість відносно легко порівнювати результати побудованих моделей, отримані різними методами. Така можливість реалізована, наприклад, в пакеті Statistica, де порівняння засноване на так званій «конкурентній оцінці моделей». Ця оцінка полягає в застосуванні різних моделей до одного і того ж набору даних і в наступному порівнянні їх характеристик для вибору найкращої з них.

Основні методи. Кілька основних методів, які використовуються для інтелектуального аналізу даних, описують тип аналізу й операцію з відновлення даних.

Розглянемо деякі ключові методи і приклади того, як використовувати ті чи інші інструменти для інтелектуального аналізу даних.

Асоціація (або відношення), ймовірно, найбільш відомий, знайомий і простий метод інтелектуального аналізу даних. Для виявлення моделей виконується просте зіставлення двох або більше елементів, часто одного і того ж типу. Наприклад, відстежуючи звички покупця, можна помітити, що разом із полуницею зазвичай купують вершки.

Створити інструменти інтелектуального аналізу даних на базі асоціацій або відносин неважко. Наприклад, в InfoSphere Warehouse є майстер, який видає конфігурації інформаційних потоків для створення асоціацій, досліджуючи джерело вхідної інформації, базис прийняття рішень і вихідну інформацію. На рис 4.1 наведено відповідний приклад бази даних.

Класифікацію можна використовувати для отримання уявлення про тип покупців, товарів або об’єктів, описуючи кілька атрибутів для ідентифікації певного класу. Наприклад, автомобілі легко класифікувати за типом (седан, позашляховик, кабріолет), визначивши різні атрибути (кількість місць, форма кузова, ведучі колеса). Вивчаючи новий автомобіль, можна віднести його до певного класу, порівнюючи атрибути з відомим визначенням. Ті ж принципи можна застосувати і до покупців, наприклад, класифікуючи їх за віком та соціальною групою.

Крім того, класифікацію можна використовувати як вхідні дані для інших методів. Наприклад, для визначення класифікації можна застосовувати дерева прийняття рішень. Кластеризація дозволяє використовувати загальні атрибути різних класифікацій з метою виявлення кластерів.

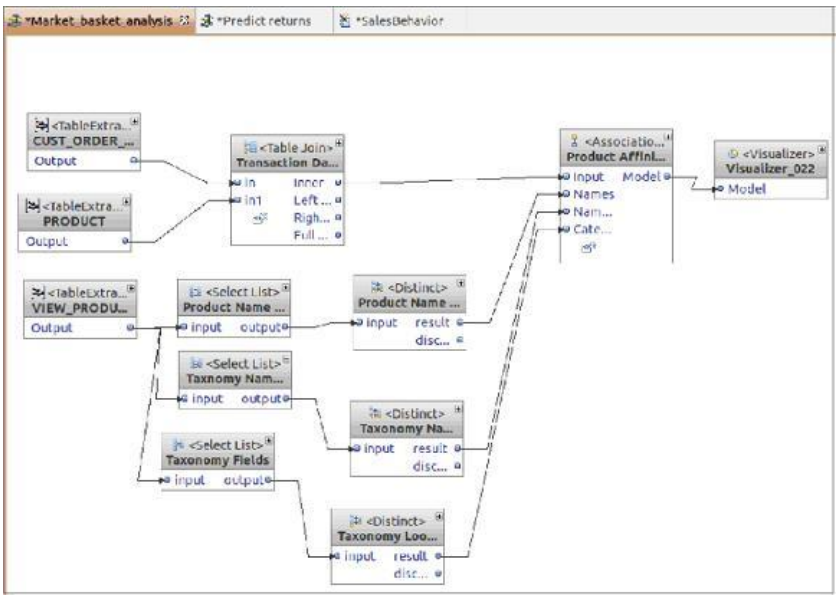


Рисунок 4.1 – Інформаційний потік, який використовується при застосуванні методу асоціації

Досліджуючи один або більше атрибутів або класів, можна згрупувати окремі елементи даних разом, отримуючи структурований висновок. На простому рівні при кластеризації використовується один або кілька атрибутів як основа для визначення кластера подібних результатів. Кластеризація корисна при визначенні різної інформації, тому що вона корелюється з іншими прикладами так, що можна побачити, де подібності і діапазони узгоджуються між собою.

Метод кластеризації працює в обидві сторони. Можна припустити, що в певній точці існує кластер, а потім використовувати свої критерії ідентифікації, щоб перевірити це. Графік, зображений на рис. 4.2, – наочний приклад. Тут вік покупця порівнюється з вартістю покупки. Розумно очікувати, що люди у віці від двадцяти до тридцяти років (до вступу в шлюб і появи дітей), а також в 50-60 років (коли діти покинули будинок) мають більш високий наявний дохід.

У цьому прикладі видно два кластери, один в діапазоні $ 2000/20-30 років та інший в діапазоні $7000-8000/50-65 років. У цьому випадку ми висунули гіпотезу і перевірили її на простому графіку, який можна побудувати за допомогою будь-якого відповідного програмного забезпечення для побудови графіків. Для більш складних комбінацій потрібен повний аналітичний пакет, особливо якщо потрібно автоматично засновувати рішення на інформації про найближчого сусіда.

Така побудова кластерів являє собою спрощений приклад так званого образу найближчого сусіда. Окремих покупців можна розрізняти за їх буквальною близькістю один до одного на графіку. Досить імовірно, що покупці з одного і того ж кластеру поділяють і інші загальні атрибути, і це припущення можна використовувати для пошуку, класифікації та інших видів аналізу членів набору даних.

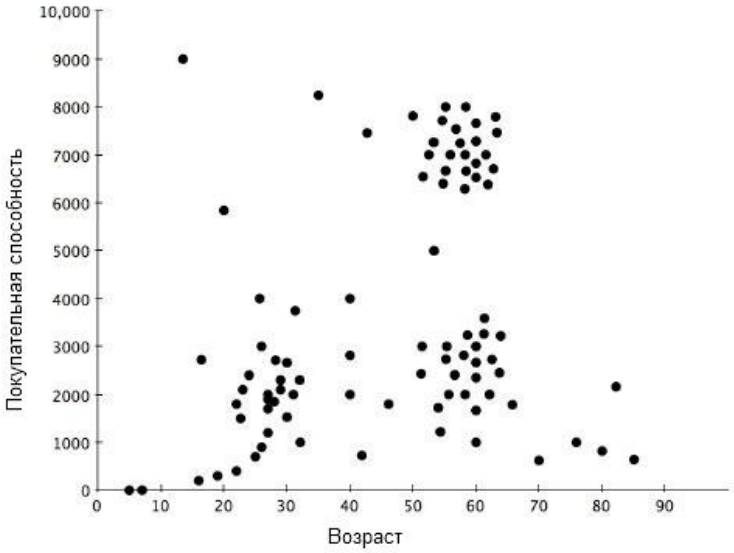


Рисунок 4.2 – Кластеризація

Метод кластеризації можна застосувати і в зворотний бік: враховуючи певні вхідні атрибути, виявляти різні артефакти. Наприклад, недавнє дослідження чотиризначних PIN-кодів виявило кластери чисел у діапазонах 1-12 і 1-31 для першої та другої пар. Зобразивши ці пари на графіку, можна побачити кластери, пов’язані з датами (дні народження, ювілеї).

Прогнозування – це широка тема, яка простягається від передбачення відмов компонентів обладнання до виявлення шахрайства і навіть прогнозування прибутку компанії. У поєднанні з іншими методами інтелектуального аналізу даних прогнозування передбачає аналіз тенденцій, класифікацію, зіставлення з моделлю і відносини. Аналізуючи минулі події або примірники, можна передбачати майбутнє.

Наприклад, використовуючи дані по авторизації кредитних карт, можна об’єднати аналіз дерева рішень минулих транзакцій людини з класифікацією і зіставленням з історичними моделями з метою виявлення шахрайських транзакцій. Якщо, наприклад, купівля авіаквитків збігається з транзакціями, то цілком імовірно, що ці транзакції справжні.

Послідовні моделі, які часто використовуються для аналізу довгострокових даних, – корисний метод виявлення тенденцій або регулярних повторень подібних подій.

Наприклад, за даними про покупців можна визначити, що в різний час року вони купують певні набори продуктів. За цією інформацією додаток прогнозування купівельної корзини, ґрунтуючись на частоті та історії покупок, може автоматично припустити, що в кошик будуть додані ті чи інші продукти.

Дерево рішень, пов’язане з більшістю інших методів (головним чином, класифікації та прогнозування), можна використовувати або в рамках критеріїв відбору, або для підтримки вибору певних даних у рамках загальної структури. Дерево рішень починають із простого питання, яке має дві відповіді (іноді

більше). Кожна відповідь приводить до наступного питання, допомагаючи класифікувати та ідентифікувати дані або робити прогнози.

Дерева рішень часто використовуються із системами класифікації інформації про властивості і з системами прогнозування, де різні прогнози можуть ґрунтуватися на минулому історичному досвіді, який допомагає побудувати структуру дерева рішень і отримати результат.

На практиці дуже рідко використовується тільки один із цих методів. Класифікація і кластеризація – подібні методи. Використовуючи кластеризацію для визначення найближчих сусідів, можна додатково уточнити класифікацію. Дерева рішень часто використовуються для побудови і виявлення класифікацій, які можна простежувати на історичних періодах для визначення послідовностей і моделей.

При всіх основних методах часто має сенс записувати і згодом вивчати отриману інформацію. Для деяких методів це абсолютно очевидно. Наприклад, при побудові послідовних моделей та навчанні з метою прогнозування аналізуються історичні дані з різних джерел і примірників інформації.

В інших випадках цей процес може бути більш яскраво вираженим. Дерева рішень рідко будуються один раз і ніколи не забуваються. При виявленні нової інформації, подій і точок даних може знадобитися побудова додаткових гілок або навіть зовсім нових дерев.

Деякі з цих процесів можна автоматизувати. Наприклад, побудова прогностичної моделі для виявлення шахрайства з кредитними картами зводиться до визначення ймовірностей, які можна використовувати для поточної транзакції, з подальшим оновленням цієї моделі при додаванні нових (підтверджених) транзакцій. Потім ця інформація реєструється, так що наступного разу рішення можна буде прийняти швидше.

Підготовка даних і очищення даних – надзвичайно важливий крок у процесі «видобутку даних». У типових проектах «видобутку даних» великі набори даних, зібрані за допомогою деяких автоматичних методів (наприклад, за допомогою Web), служать вхідними даними аналізу. Часто метод, за допомогою якого були зібрані дані, не був жорстко регульованим, внаслідок чого дані можуть містити значення, що виходять за допустимі межі (наприклад, Дохід: -100), неможливі комбінації даних (наприклад, Пол: Чоловік, Вагітність: Так) тощо.

При видобутку даних вхідні дані часто «зачумлені» – містять багато помилок та, іноді, інформацію в неструктурованою формі. Припустимо, що ви хочете проаналізувати велику базу даних, зібраних за допомогою Web у режимі он-лайн, ґрунтуючись на добровільних відповідях людей, які відвідують ваш Web-сайт (наприклад, потенційних клієнтів Web-продавця, який заповнив запропоновані анкети). У цьому прикладі дуже важливо спочатку перевірити і «очистити» дані на стадії підготовки даних, перед тим як застосовувати аналітичні процедури. Наприклад, деякі індивідууми можуть ввести завідомо помилкову інформацію (наприклад, вік = 300). У таких типах даних помилки не

виявляються до стадії аналізу. Вони можуть сильно зміщувати результат і приводити до невиправданих висновків. зазвичай протягом стадії підготовки даних аналітик застосовує «фільтри» до даних для перевірки правильності їх діапазонів і виключення неможливих значень (наприклад, Вік = 5; Пенсіонер = Так).

Питання для самоконтролю

1. Дайте визначення методу та алгоритму.
2. Які ви знаєте методи і алгоритми Data Mining?
3. Із яких стадій складається процес Data Mining?
4. Які ви знаєте види класифікацій технологічних методів Data Mining?
5. Приведіть приклади застосування інструменту Data Mining асоціації (або відношення)?
6. Для чого застосовуються інструментарій Data Mining дерева рішень (наведіть приклад)?

Тема 5. Задачі Data Mining та їх класифікація. Інформація та знання

План

1. Задачі Data Mining.
2. Класифікація задач інтелектуального аналізу даних.
3. Рівні аналізу.
4. Інформація. Властивості інформації.

Мета вивчення теми: вивчити задачі інтелектуального аналізу даних; засвоїти рівні аналізу Data Mining; засвоїти поняття інформації та вивчити її властивості.

Перелік ключових слів та понять із теми

Data Mining, інформація, класифікація, кластеризація, прогнозування, асоціація, візуалізація, аналіз і виявлення відхилень, оцінювання, аналіз зв'язків, підведення підсумків

Теоретичні відомості з теми

1. Задачі Data Mining

В основу технології Data Mining покладена концепція шаблонів, що представляють собою закономірності. У результаті виявлення цих, прихованих від неозброєного ока закономірностей вирішуються завдання інтелектуального аналізу даних

Задачі (tasks) Data Mining іноді називають закономірностями (regularity) або техніками (techniques).

Єдиної думки щодо того, які задачі слід відносити до Data Mining, немає.

Більшість авторитетних джерел перераховують такі задачі: класифікація, кластеризація, прогнозування, асоціація, візуалізація, аналіз і виявлення відхилень, оцінювання, аналіз зв'язків, підведення підсумків.

Класифікація (Classification). Найбільш проста і поширена задача Data Mining. У результаті розв'язання задачі класифікації виявляються ознаки, які характеризують групи об'єктів досліджуваного набору даних – класи; за цими ознаками новий об'єкт можна віднести до того чи іншого класу.

Методи розв'язання. Для розв'язання задачі класифікації можуть використовуватися методи: найближчого сусіда (Nearest Neighbor), *k*-найближчого сусіда (*k*-Nearest Neighbor); байєсовські мережі (Bayesian Networks); індукція дерев рішень; нейронні мережі (neural networks).

Кластеризація (Clustering) є логічним продовженням ідеї класифікації. Ця задача більш складна. Особливість кластеризації полягає в тому, що класи об'єктів спочатку не визначені. Результатом кластеризації є розбиття об'єктів на групи.

Приклад методу розв'язання задачі кластеризації: навчання «без вчителя» особливого виду нейронних мереж – самоорганізованих карт Кохонена.

Асоціація (Associations). У ході розв'язання задачі пошуку асоціативних правил відшукуються закономірності між пов'язаними подіями в наборі даних.

Відмінність асоціації від двох попередніх задач Data Mining полягає в тому, що пошук закономірностей здійснюється не на основі властивостей аналізованого об'єкта, а між кількома подіями, які відбуваються одночасно.

Найбільш відомий алгоритм розв'язання задачі пошуку асоціативних правил – алгоритм Apriori.

Послідовність (Sequence), або послідовна асоціація (sequential association). Послідовність дозволяє знайти тимчасові закономірності між транзакціями. Задача послідовності подібна асоціації, але її метою є встановлення закономірностей не між подіями, що наступають одночасно, а між подіями, які пов'язаними в часі (тобто відбуваються з деяким певним інтервалом у часі). Іншими словами послідовність визначається високою ймовірністю ланцюжка пов'язаних у часі подій.

Фактично, асоціація є окремим випадком послідовності з тимчасовим лагом, рівним нулю. Цю задачу Data Mining також називають задачею знаходження послідовних шаблонів (sequential pattern).

Правило послідовності: після події X через певний час відбудеться подія Y.

Приклад. Після покупки квартири мешканці в 60% випадків протягом двох тижнів купують холодильник, а протягом двох місяців в 50% випадків купується телевізор. Розв'язок цієї задачі широко застосовується в маркетингу і менеджменті, наприклад, при управлінні циклом роботи з клієнтом (управління життєвим циклом клієнта).

Прогнозування (Forecasting). У результаті розв'язання задачі прогнозування на основі особливостей історичних даних оцінюються пропущені або ж майбутні значення цільових чисельних показників.

Для розв'язання таких задач широко застосовуються методи математичної статистики, нейронні мережі тощо.

Визначення відхилень або викидів (Deviation Detection), аналіз відхилень або викидів. Мета розв'язання цієї задачі – виявлення та аналіз даних, що найбільш відрізняються від загальної множини даних, виявлення так званих нехарактерних шаблонів.

Оцінювання (оцінка). Задача оцінювання зводиться до передбачення неперервних значень ознаки.

Аналіз зв'язків (Link Analysis) – задача знаходження залежностей в наборі даних.

Візуалізація (Visualization, Graph Mining). У результаті візуалізації створюється графічний образ аналізованих даних. Для розв'язання задачі візуалізації використовуються графічні методи, що показують наявність закономірностей у даних.

Приклад методу візуалізації – подання даних у 2-D і 3-D вимірах.

Підведення підсумків (Summarization) – задача, мета якої – опис конкретних груп об'єктів з аналізованого набору даних.

2. Класифікація задач інтелектуального аналізу даних

Згідно класифікації за стратегіями, задачі Data Mining поділяються на такі групи:

• навчання з учителем;

• навчання без вчителя;

• інші.

Категорія «навчання з учителем» представлена такими задачами Data Mining: класифікація, оцінка, прогнозування.

Категорія «навчання без вчителя» представлена задачею кластеризації.

До категорії «інші» належать задачі, не включені в попередні дві стратегії.

Задачі інтелектуального аналізу даних, залежно від використовуваних моделей, можуть бути **дескриптивними** і **прогнозуючими.**

Відповідно до цієї класифікації, **задачі Data Mining** представлені **групами описових і прогнозуючих задач.**

У результаті розв'язання описових (descriptive) задач аналітик отримує шаблони, що описують дані, які піддаються інтерпретації.

Ці задачі описують загальну концепцію аналізованих даних, визначають інформативні, підсумкові, відмінні особливості даних. Концепція описових задач передбачає характеристику і порівняння наборів даних.

Характеристика набору даних забезпечує короткий і стислий опис деякого набору даних.

Порівняння забезпечує порівняльний опис двох або більше наборів даних.

Прогнозуючі (predictive) задачі ґрунтуються на аналізі даних, створенні моделі, передбаченні тенденцій або властивостей нових або невідомих даних.

Досить близьким до вищезгаданої класифікації є розділення задач Data Mining на такі:

a) дослідження та відкриття;
b) прогнозування та класифікація;
c) пояснення й опис.

Автоматичне дослідження і відкриття (вільний пошук). *Приклад задачі: виявлення нових сегментів ринку.*

Для розв'язання цього класу задач використовуються методи кластерного аналізу прогнозування та класифікація.

Приклад задачі: передбачення зростання обсягів продажів на основі поточних значень.

Методи: регресія, нейронні мережі, генетичні алгоритми, дерева рішень.

Задачі класифікації та прогнозування становлять групу так званого індуктивного моделювання, в результаті якого забезпечується вивчення аналізованого об'єкта або системи. У процесі вирішення цих завдань на основі набору даних розробляється загальна модель або гіпотеза.

Пояснення й опис. *Приклад задачі: характеристика клієнтів за демографічними даними і історіями покупок.*

Методи: дерева рішень, системи правил, правила асоціації, аналіз зв'язків.

Якщо дохід клієнта більше, ніж 50 умовних одиниць, і його вік – понад 30 років, тоді клас клієнта – перший.

В інтерпретації узагальненої моделі аналітик отримує нове знання. Групування об’єктів відбувається на основі їх подібності.

Нагадаємо, що головна цінність Data Mining – це практична спрямованість даної технології, шлях від сирих даних до конкретного знання, від постановки завдання до готового додатку, за підтримки якого можна приймати рішення.

Велика кількість понять, які об’єдналися в Data Mining, а також різноманітність методів, що підтримують дану технологію, починаючому аналітику можуть нагадати мозаїку, частини якої мало пов’язані між собою.

Як же ми можемо зв’язати в одне ціле задачі, методи, дії, закономірності, додатки, дані, інформацію, рішення?

Розглянемо два потоки:

1. Дані – інформація – знання і рішення.
2. Завдання – дії і методи розв’язання – програми.

Ці потоки є «двома сторонами однієї медалі», відображенням одного процесу, результатом якого має бути знання і прийняття рішення.

Від даних до рішень. Для початку розглянемо перший потік. На рис. 5.1 показано зв’язок понять «дані», «інформація» і «рішення», яка виникає в процесі прийняття рішень.

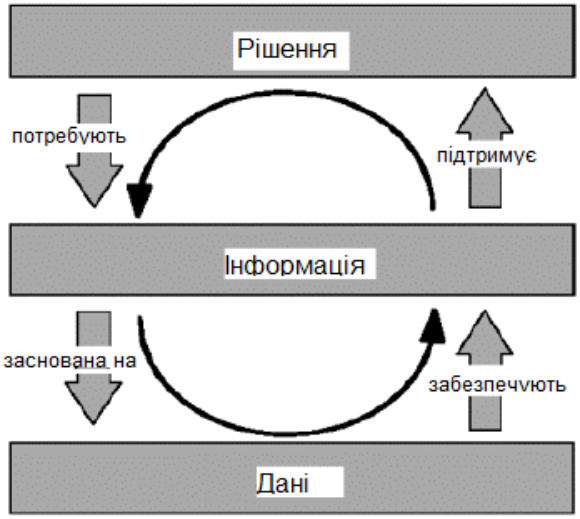


Рисунок 5.1 – Рішення, інформація і дані

Як видно з рисунку, цей процес є циклічним. Прийняття рішень потребує інформації, яка заснована на даних. Дані забезпечують інформацію, яка підтримує рішення і т.д.

Розглянуті поняття є складовою частиною так званої інформаційної піраміди, в основі якої знаходяться дані, наступний рівень – це інформація, потім йде рішення, завершує піраміду рівень знання. У міру просування вгору по інформаційній піраміді обсяги даних переходять у цінність рішень, тобто цінність для бізнесу. А, як відомо, метою Business Intelligence є перетворення обсягів даних у цінність бізнесу.

Тепер підійдемо до цього ж процесу з іншого боку. Розглянемо рис. 5.2.

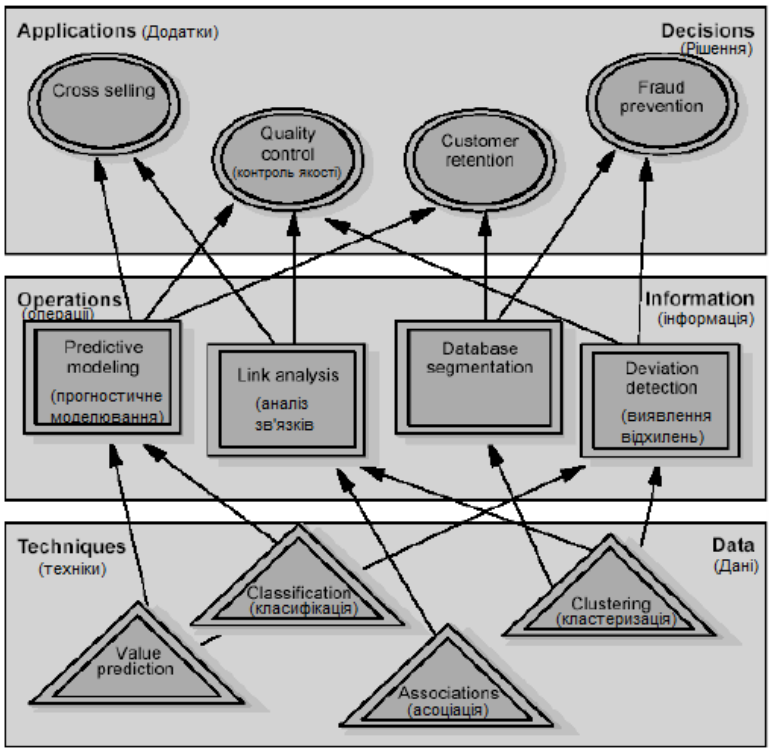


Рисунок 5.2 – Задачі, дії, додатки

Слід зазначити, що рівні аналізу (дані, інформація, знання) практично відповідають етапам еволюції аналізу даних, яка відбувалася протягом останніх років.

3. Рівні аналізу

Верхній – рівень додатків – є рівнем бізнесу (якщо ми маємо справу із завданням бізнесу), на ньому менеджери приймають рішення. Наведені приклади додатків: перехресні продажі, контроль якості, утримування клієнтів.

Середній – рівень дій – за своєю суттю є рівнем інформації, саме на ньому виконуються дії Data Mining; на рисунку наведені такі дії: прогностичне моделювання, аналіз зв'язків, сегментація даних та інші.

Нижній – рівень визначення задачі інтелектуального аналізу даних, яку необхідно розв'язати стосовно даних, що є в наявності, на рисунку наведені завдання передбачення числових значень, класифікація, кластеризація, асоціація.

Розглянемо таблицю, що демонструє зв'язок цих понять.

Таблиця 5.1 − Рівні Data Mining

Рівень 3	Додатки	Утримання клієнтів	Знання DM	Результат
Рівень 2	Дії	Прогностичне моделювання	Інформація	Метод аналізу
Рівень 1	Задачі	Класифікація	Дані	Запити

Нагадаємо, що для розв'язання задачі класифікації результати роботи першої стадії (індукції правил) використовуються для віднесення нового об'єкта

з певною впевненістю до одного з відомих, визначених класів на підставі відомих значень.

Розглянемо задачі утримання клієнтів (визначення надійності клієнтів фірми).

Дані – база даних за клієнтами. Є дані про клієнта (вік, стать, професія, дохід). Певна частина клієнтів, скориставшись продуктом фірми, залишилася їй вірною; інші клієнти більше не купували продукти фірми. На цьому рівні визначаємо тип задачі – це задача класифікації.

На другому рівні визначаємо дію – прогностичне моделювання. За допомогою прогностичного моделювання ми з певною частиною впевненості можемо віднести новий об'єкт, у цьому випадку, нового клієнта, до одного з відомих класів – постійний клієнт, або це, швидше за все, його разова покупка.

На третьому рівні ми можемо скористатися додатком для прийняття рішення. У результаті придбання знань, фірма може істотно знизити витрати, наприклад, на рекламу, знаючи заздалегідь, яким із клієнтів слід активно розсилати рекламні матеріали.

Отже, ми визначилися з поняттями «дані», «завдання», «методи», «дії».

4. Інформація. Властивості інформації

Інформація (лат. informātiō) –

1) будь-які повідомлення про що-небудь;

2) відомості, що є об'єктом зберігання, переробки і передачі (наприклад, генетична інформація);

3) у математиці (кібернетиці) – кількісна міра усунення невизначеності (ентропія), міра організації системи; в теорії інформації – розділ кібернетики, що вивчає кількісні закономірності, які пов'язані зі збором, передачею, перетворенням і обчисленням інформації.

Інформація – будь-які, невідомі раніше відомості про якусь подію, сутності, процеси і т.п., є об'єктом деяких операцій, для яких існує змістовна інтерпретація.

Під операціями тут мається на увазі сприйняття, передача, перетворення, зберігання і використання. Для сприйняття інформації необхідна деяка сприймаюча система, яка може інтерпретувати її, перетворювати, визначати відповідність певним правилам і т.п. Отже, поняття інформації слід розглядати тільки при наявності джерела і одержувача інформації, а також каналу зв'язку між ними.

Властивості інформації:

• **повнота інформації.** Це властивість характеризує якість інформації і визначає достатність даних для прийняття рішень, тобто інформація повинна містити весь необхідний набір даних.

Приклад. «Продажі товару *А* почнуть скорочуватися». Ця інформація неповна, оскільки невідомо, коли саме вони почнуть скорочуватися.

Приклад повної інформації. «Починаючи з першого кварталу, продажі товару *А* почнуть скорочуватися». Цієї інформації достатньо для прийняття рішень;

• **достовірність інформації.** Інформація може бути **достовірною** і **недостовірною**. У недостовірній інформації присутній інформаційний шум, і чим він вищий, тим нижче достовірність інформації;

• **цінність інформації.** Цінність інформації не може бути абстрактною. Інформація повинна бути корисною і цінною для певної категорії користувачів;

• **адекватність інформації.** Ця властивість характеризує ступінь відповідності інформації реальному об'єктивному стану. Адекватна інформація – це повна і достовірна інформація;

• **актуальність інформації.** Інформація повинна бути актуальною, тобто НЕ застарілою. Ця властивість інформації характеризує ступінь відповідність інформації справжньому моменту часу;

• **ясність інформації.** Інформація повинна бути зрозуміла для того кола осіб, для якого вона призначена;

• **доступність інформації.** Доступність характеризує міру можливості отримати певну інформацію. На цю властивість інформації впливають одночасно доступність даних і доступність адекватних методів;

• **суб'єктивність інформації.** Інформація носить суб'єктивний характер, вона визначається ступенем сприйняття суб'єкта (одержувача інформації).

Вимоги, що пред'являються до інформації:

• **динамічний характер інформації.** Інформація існує тільки в момент взаємодії даних і методів, тобто в момент інформаційного процесу. Решту часу вона перебуває в стані даних;

• **адекватність використовуваних методів.** Інформація витягується з даних. Проте в результаті використання одних і тих даних може з'являтися різна інформація. Це залежить від адекватності вибраних методів обробки вихідних даних.

Дані, за своєю суттю, є **об'єктивними**. **Методи** є **суб'єктивними**, в основі методів лежать алгоритми, суб'єктивно складені та підготовлені. Отже, інформація виникає та існує в момент діалектичної взаємодії об'єктивних даних і суб'єктивних методів.

Для бізнесу інформація є вихідною складової прийняття рішень.

Всю інформацію, що виникає в процесі функціонування бізнесу та управління ним, можна класифікувати певним чином. Залежно від **джерела одержання, інформацію** поділяють на **внутрішню** і **зовнішню** (наприклад, інформація, що описує явища, які відбуваються за межами фірми, але мають до неї безпосереднє відношення).

Також **інформація** може бути класифікована на **фактичну** і **прогнозну**. До фактичної інформації про бізнес відноситься інформація, що характеризує доконаний факт, вона є точною. Прогнозна інформація розраховується або передбачається, тому її не можна вважати точною, вона може мати певну похибку.

Знання – сукупність фактів, закономірностей і евристичних правил, за допомогою яких вирішується поставлене завдання.

Отже, формування інформації відбувається в процесі збору та передачі, тобто обробки даних. Яким же чином із інформації отримують знання?

Усе частіше істинні знання утворюються на основі розподілених взаємозв'язків різнорідної інформації. Коли інформація зібрана і передана для отримання явно не визначеного заздалегідь результату, то ви отримуєте знання. Сама по собі інформація в чистому вигляді безглузда. Звідси випливає висновок, що інформація – це чиєсь тактичне знання, передане у вигляді символів і за допомогою будь-яких прикладних засобів.

За визначенням Денхема Грея, «знання – це абсолютне використання інформації і даних, спільно з потенціалом практичного досвіду людей, здібностями, ідеями, інтуїцією, переконаністю і мотиваціями».

Знання мають певні **властивості**, які відрізняють їх від інформації.

1. Структурованість. Знання повинні бути «розкладені по поличках».

2. Зручність доступу і засвоєння. Для людини – це здатність швидко зрозуміти і запам'ятати або, навпаки, згадати, для комп'ютерних знань – засоби доступу до знань.

3. Лаконічність. Лаконічність дозволяє швидко освоювати і переробляти знання і підвищує «коефіцієнт корисного змісту». У цей список лаконічність була додана через усім відому проблему шуму і сміттєвих документів, характерних саме для комп'ютерної інформації – Інтернету та електронного документообігу.

4. Несуперечливість. Знання не повинні суперечити одне одному.

5. Процедури обробки. Знання потрібні для того, щоб їх використовувати. Одна з головних властивостей знань – можливість їх передачі іншим і здатність робити висновки на їх основі. Для цього повинні існувати процедури обробки знань. Здатність робити висновки означає для машини наявність процедур обробки та виведення і підготовленість структур даних для такої обробки, тобто наявність спеціальних форматів знань.

Зіставлення і порівняння понять «інформація», «дані», «знання».

Для того, щоб впевнено оперувати поняттями «інформація», «дані», «знання», необхідно не тільки розуміти суть цих понять, а й відчути відмінності між ними. Однак, однієї інтуїтивної інтерпретації цих понять тут недостатньо. Складність розуміння відмінностей вищезазначених понять – в їх уявній синонімічності. Згадаймо, що поняття Data Mining переводиться на українську мову за допомогою цих же трьох понять: як видобуток даних, вилучення інформації, розкопування знань.

Для початку зробимо спробу розібратися в цих термінах на простих прикладах.

1. Студент, який здає іспит, потребує даних.

2. Студент, який здає іспит, потребує інформації.

3. Студент, який здає іспит, потребує знань.

При розгляді першого варіанту – студент потребує даних – виникає думка, що студенту потрібні дані, наприклад, для обчислень. Інформацією в другому варіанті може виступати конспект або підручник. У результаті їх використання студент отримує лише інформацію, яка в певних випадках може перейти у знання. Третій варіант звучить найбільш логічно.

Інформація, на відміну від даних, має сенс.

Поняття «інформація» і «знання», з філософської точки зору, є поняттями більш високого рівня, ніж «дані», яке виникло відносно недавно.

Поняття «інформації» безпосередньо пов'язано із сутністю процесів усередині інформаційної системи, тоді як поняття «знання» швидше орієнтоване на якість процесів. Поняття «знання» тісно пов'язане з процесом прийняття рішень.

Незважаючи на відмінності, розглянуті поняття, як уже зазначалося раніше, не є розрізненими і непов'язаними. Вони є частиною одного потоку: біля витоку його знаходяться дані, у процесі передачі яких виникає інформація, і в результаті використання інформації, за певних умов, виникають знання.

У процесі руху вгору в інформаційній піраміді обсяги даних переходять у цінність знань. Однак великі обсяги даних зовсім не означають і, тим більше, не гарантують отримання знань. Існує певна залежність цінності отриманих знань від якості та потужності процедур обробки даних. Типовим прикладом інформації, яку не можна перетворити в знання, є текст іноземною мовою. За відсутності словника і перекладача ця інформація взагалі не має цінності, вона не може перейти в знання. За наявності словника процес переходу від інформації до знання можливий, але тривалий і трудомісткий. За наявності перекладача інформація дійсно переходить в знання.

Таким чином, для отримання цінних знань необхідні якісні процедури обробки. Процес переходу від даних до знань займає багато часу і коштує дорого. Тому очевидно, що технологія Data Mining з її потужними і різноманітними алгоритмами є інструментом, за допомогою якого, просуваючись вгору по інформаційній піраміді, ми можемо отримувати дійсно якісні та цінні знання.

Питання для самоконтролю

1. Дайте визначення класифікації (Classification).
2. Дайте визначення кластеризації (Clustering).
3. На які групи поділяють задачі Data Mining?
4. Який зв'язок між поняттями «дані», «інформація» і «рішення»? Намалюйте схему їх зв'язку.
5. Які рівні аналізу ви знаєте?
6. Які властивості мають знання?

Тема 6. Задачі Data Mining. Класифікація та кластеризація

План

1. Задачі та види класифікації.
2. Методи, що застосовуються для розв’язання задач класифікації.
3. Задача кластеризації.
4. Застосування кластерного аналізу.

Мета вивчення теми: вивчити задачі класифікації та кластеризації; засвоїти принцип штучної та природної класифікації.

Перелік ключових слів та понять із теми

Data Mining, класифікація, кластеризація, проста та складна класифікація, багатовимірна класифікація, штучна та природна класифікація

Теоретичні відомості з теми

1. Задачі та види класифікації.

Класифікація є найбільш простою і водночас найбільш часто розв’язуваною задачею Data Mining. Зважаючи на поширеність задач класифікації, необхідно чітке розуміння суті цього поняття.

Наведемо кілька визначень.

Класифікація – системний розподіл досліджуваних предметів, явищ, процесів за родами, видами, типами, з якими-небудь істотними ознаками для зручності їх дослідження; угрупування вихідних понять і розташування їх у певному порядку, що відбиває ступінь цієї схожості.

Класифікація – упорядкована за деяким принципом множина об’єктів, які мають подібні класифікаційні ознаки (одна або декілька властивостей), обраних для визначення схожості або відмінності між цими об’єктами.

Класифікація вимагає дотримання таких правил:

1) у кожному акті ділення необхідно застосовувати тільки одну основу;
2) ділення повинне бути пропорційним, тобто загальний обсяг видових понять повинен дорівнювати обсягу діленого родового поняття;
3) члени ділення повинні взаємно виключати один одного, їх обсяги не повинні перехрещуватися;
4) ділення повинне бути послідовним.

Розрізняють:

1) **допоміжну (штучну) класифікацію**, яка виробляється за зовнішньою ознакою і служить для надання множині предметів (процесів, явищ) потрібного порядку;

2) **природну класифікацію**, яка виробляється за істотними ознаками, що характеризують внутрішню спільність предметів і явищ. Вона є результатом і

важливим засобом наукового дослідження, тому що передбачає і закріплює результати вивчення закономірностей об'єктів, що класифікуються.

Допоміжна класифікація створюється з метою найбільш швидкого відшукування якогось індивідуального предмету серед предметів, що класифікуються. Мета цієї класифікації визначає принцип її побудови. В основу допоміжної класифікації лягає яка-небудь зовнішня несуттєва ознака, яка, однак, виявляється корисною в процесі пошуку.

Прикладами допоміжної класифікації можуть бути розподіл студентів курсу в списку в алфавітному порядку або такий же розподіл бібліотечних карток в алфавітному каталозі тощо. Знаючи порядок букв в алфавіті, ми можемо легко і швидко відшукати потрібне нам прізвище у списку або дані, що цікавлять нас у книзі, в каталозі.

Але знання того, яке місце в допоміжній класифікаційній системі займає той чи інший предмет, не дає можливості щось стверджувати про його властивості. Так, наприклад, те, що студент Архипов записаний у списку першим, а студент Яковлєв – останнім, нічого не говорить про їх здібності і риси характеру. Тому допоміжна класифікація не є науковою.

На відміну від допоміжної **природна класифікація** являє собою розподіл предметів за класами на підставі їх найбільш суттєвих ознак. Найбільш істотними є такі ознаки предмета, які зумовлюють інші його ознаки. Наприклад, найбільш суттєвою ознакою людини є її здатність до праці. Ця ознака зумовлює наявність у людини таких ознак, як прямоходіння, здатність до спілкування (праця передбачає колектив), здатність до мислення та ін.

Залежно від обраних ознак, їх поєднання і процедури розподілу понять, **класифікація може бути**:

- **простою** – розподіл родового поняття тільки за ознакою і тільки один раз до розкриття всіх видів. Прикладом такої класифікації є дихотомія, при якій членами поділу бувають тільки два поняття, кожне з яких суперечить іншому (тобто дотримується принцип: «А» і «не А»);
- **складною** – застосовується для поділу одного поняття за різними основами і синтезу таких простих ділень в єдине ціле.

Прикладом такої класифікації є періодична система хімічних елементів.

Під **класифікацією** будемо розуміти віднесення об'єктів (спостережень, подій) до одного із заздалегідь відомих класів.

Класифікація – це закономірність, що дозволяє робити висновок щодо визначення характеристик конкретної групи. Отже, для проведення класифікації повинні бути присутні ознаки, що характеризують групу, до якої належить та чи інша подія або об'єкт (зазвичай при цьому на підставі аналізу вже класифікованих подій формулюються якісь правила).

Класифікація відноситься до стратегії навчання з вчителем (supervised learning), яку також іменують контрольованим або керованим навчанням.

Маши́нне навчання — узагальнена назва штучної генерації знань із досвіду. Штучна система навчається на прикладах і після закінчення фази

навчання може узагальнювати. Тобто система не просто вивчає наведені приклади, а розпізнає певні закономірності в даних для навчання.

Серед багатьох програмних продуктів варто згадати системи автоматичного діагностування, розпізнавання шахрайства з кредитними картками, аналіз ринку цінних паперів, класифікація ланцюжків ДНК, розпізнавання мовлення та тексту, автономні системи.

Практичне використання відбувається, переважно, за допомогою алгоритмів. Різноманітні алгоритми машинного навчання можна грубо поділити за такою схемою:

1. **Навчання з вчителем** (англ. Supervised learning): алгоритм вивчає функцію на основі наданих пар вхідних та вихідних даних. При цьому, в процесі навчання, «вчитель» вказує вірні вихідні дані для кожного значення вхідних даних. Одним із розділів навчання з вчителем є машинна класифікація. Такі алгоритми застосовуються для розпізнавання текстів.
2. **Навчання без вчител**я (англ. Unsupervised learning).
3. **Навчання із закріпленням** (англ. Reinforcement Learning): алгоритм навчається за допомогою тактики нагороди та покарання для максимізації вигоди для агентів (систем, до яких належить компонента, що навчається).

Задачею класифікації часто називають передбачення категоріальної залежної змінної (тобто залежної змінної, що є категорією) на основі вибірки безперервних і/або 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 т.д. Цей тип завдань належить до завдань **бінарної класифікації**, в них залежна змінна може приймати тільки два значення (наприклад, так чи ні, 0 або 1).

Інший варіант класифікації виникає, якщо залежна **змінна** може приймати значення з деякої множини визначених 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** (за двома і більше ознаками).

Багатовимірна класифікація була розроблена біологами при вирішенні проблем дискримінації для класифікування організмів. Однією з перших робіт, присвячених цьому напрямку, вважають роботу Р. Фішера (1930 р.), в якій організми поділялися на підвиди залежно від результатів вимірювань їх фізичних параметрів. Біологія була і залишається найбільш затребуваним і зручним середовищем для розробки багатовимірних методів класифікації.

Розглянемо задачу класифікації на простому прикладі. Припустимо, є база даних про клієнтів туристичного агентства з інформацією про вік і доходи за місяць. Є рекламний матеріал двох видів: більш дорогий і комфортний відпочинок та дешевший, молодіжний відпочинок. Відповідно, визначені два класи клієнтів: клас 1 і клас 2. База даних наведена в таблиці 6.1.

Завдання. Визначити, до якого класу належить новий клієнт і який з двох видів рекламних матеріалів йому варто відсилати.

Таблиця 6.1 - База даних клієнтів туристичного агентства

Код клієнта	Вік	Дохід	Клас
1	18	25	1
2	22	100	1
3	30	70	1
4	32	120	1
5	24	15	2
6	25	22	1
7	32	50	2
8	19	45	2
9	22	75	1
10	40	90	2

Для наочності представимо нашу базу даних у двомірному просторі (вік і дохід), у вигляді множини об’єктів, що належать класам 1 (помаранчева мітка) і 2 (сіра мітка). На рис. 6.1 наведені об’єкти з двох класів.

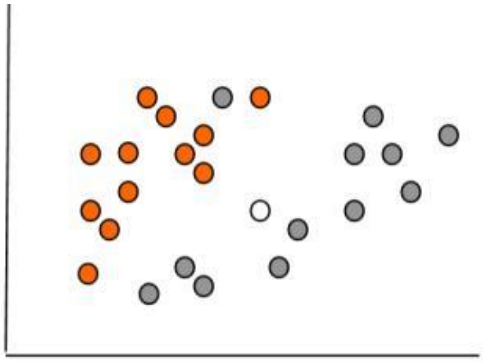


Рисунок 6.1 – Множина об’єктів бази даних у двомірному вимірі

Розв’язок нашої задачі буде полягати в тому, щоб визначити, до якого класу належить новий клієнт, на рисунку позначений білою міткою.

Мета процесу класифікації полягає в тому, щоб побудувати модель, яка використовує прогнозуючі атрибути як вхідні параметри і отримує значення залежного атрибута. **Процес класифікації** полягає в розбитті множини об’єктів на класи за певним критерієм.

Класифікатором називається якась сутність, що визначає, якому з визначених класів належить об’єкт за вектором ознак.

Для проведення класифікації за допомогою математичних методів необхідно мати формальний опис об’єкта, яким можна оперувати, використовуючи математичний апарат класифікації. Таким описом у нашому випадку виступає база даних. Кожен об’єкт (запис бази даних) несе інформацію про деякі властивості об’єкта.

Набір вихідних даних (або вибірку 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 та вихідні (цільові) значення прикладів. Вихідні значення призначені для навчання моделі.

Тестова (test set) множина також містить вхідні та вихідні значення прикладів. Тут вихідні значення використовуються для перевірки працездатності моделі.

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

1. Конструювання моделі: опис множини визначених класів.

Кожен приклад набору даних відноситься до одного визначеного класу.

На цьому етапі використовується навчальна множина, на ньому відбувається конструювання моделі.

Отримана модель **представлена класифікаційними правилами, деревом рішень або математичною формулою.**

2. Використання моделі: класифікація нових або невідомих значень.

Оцінка правильності (точності) моделі.

Відомі значення з тестового прикладу порівнюються з результатами використання отриманої моделі.

Рівень точності – відсоток правильно класифікованих прикладів у тестовій множині.

Тестова множина, тобто множина, на якій тестується побудована модель, не повинна залежати від навчальної множини.

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

2. Методи, що застосовуються для розв'язання задач класифікації

Для класифікації використовуються різні методи. Основні з них:

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

Схематичне розв'язок задачі класифікації деякими методами (за допомогою лінійної регресії, дерев рішень і нейронних мереж) наведені на рис. 6.2- 6.4.

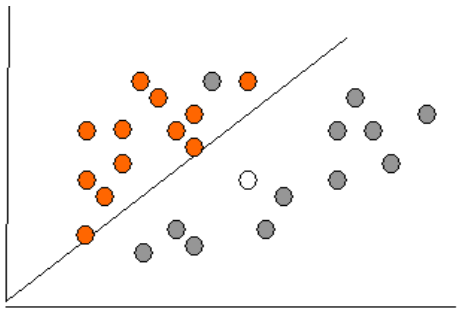


Рисунок 6.2 – Розв’язок задачі класифікації методом лінійної регресії

```
if X> 5 then grey
  else if Y> 3 then orange
          else if X> 2 then grey
                  else orange
```

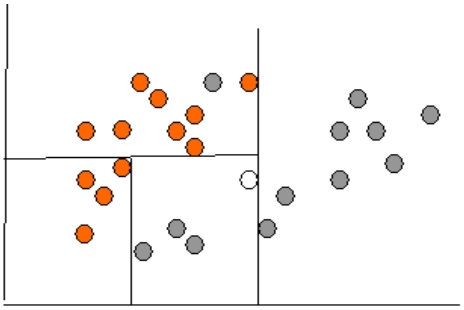


Рисунок 6.3 – Розв’язок задачі класифікації методом дерев рішень

Рисунок 6.4 – Розв’язок задачі класифікації методом нейронних мереж

Точність класифікації: оцінка рівня помилок. Оцінка точності класифікації може проводитися за допомогою крос-перевірки. Крос-перевірка (Cross-validation) – це процедура оцінки точності класифікації на даних із тестової множини, яку також називають крос-перевірочною множиною. Точність класифікації тестової множини порівнюється з точністю класифікації навчаючої множини. Якщо класифікація тестової множини дає приблизно такі ж результати за точністю, як і класифікація навчальної множини, вважається, що дана модель пройшла крос-перевірку.

Поділ на навчальну і тестову множину здійснюється шляхом ділення вибірки в певній пропорції, наприклад навчальна множина – дві третини даних і тестова – одна третина даних. Цей спосіб слід використовувати для вибірок із

великою кількістю прикладів. Якщо ж вибірка має малі обсяги, рекомендується застосовувати спеціальні методи, при використанні яких навчальна і тестова вибірки можуть частково перетинатися.

Оцінювання класифікаційних методів. Оцінювання методів слід проводити, виходячи з таких характеристик: **швидкість, робастність, інтерпретованість, надійність.**

Швидкість характеризує час, який потрібен на створення моделі та її використання.

Робастність, тобто стійкість до будь-яких порушень вихідних передумов, означає можливість роботи з зашумленими даними і пропущеними значеннями в даних.

Інтерпретованість забезпечує можливість розуміння моделі аналітиком.

Властивості класифікаційних правил:

- розмір дерева рішень;
- компактність класифікаційних правил.

Надійність методів класифікації передбачає можливість роботи цих методів при наявності в наборі даних шумів і викидів.

3. Задача кластеризації

Введемо поняття кластеризації, кластера, коротко розглянемо класи методів, за допомогою яких вирішується задача кластеризації, деякі моменти процесу кластеризації, а також розберемо приклади застосування кластерного аналізу.

Задача кластеризації схожа із задачею класифікації, є її логічним продовженням, але відмінність її в тому, що класи досліджуваного набору даних заздалегідь не зумовлені.

Синонімами терміну «кластеризацію» є «автоматична класифікація», «навчання без вчителя» і «таксономія».

Кластеризація призначена для розбиття сукупності об’єктів на однорідні групи (кластери або класи). Якщо дані вибірки представити як точки в просторі ознак, то задача кластеризації зводиться до визначення «згущувань точок».

Мета кластеризації – пошук існуючих структур. Кластеризація є описовою процедурою, вона не робить ніяких статистичних висновків, але дає можливість провести розвідувальний аналіз і вивчити «структуру даних».

Саме поняття «**кластер**» визначено неоднозначно. Перекладається поняття кластер (cluster) як «скупчення», «гроно».

Кластер можна охарактеризувати як групу об’єктів, що мають загальні властивості.

Характеристиками кластера можна назвати дві ознаки:

- внутрішня однорідність;
- зовнішня ізольованість.

Питання, що ставиться аналітиками при вирішенні багатьох завдань, полягає в тому, як організувати дані в наочні структури, тобто розгорнути таксономії.

Найбільше застосування кластеризація спочатку отримала в таких науках як біологія, антропологія, психологія. Для вирішення економічних завдань

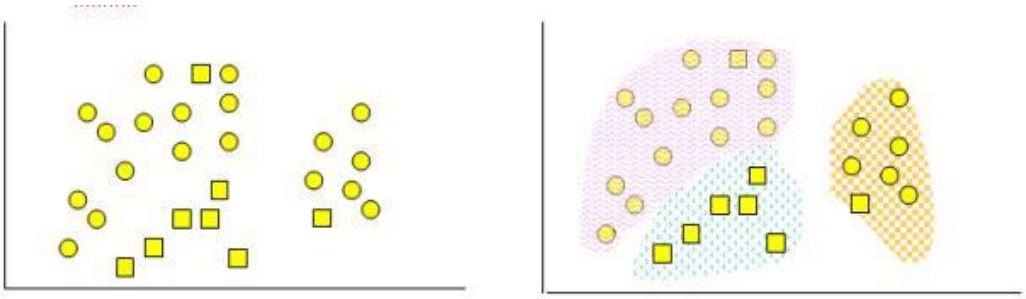
кластеризація тривалий час мало використовувалася через специфіку економічних даних і явищ.

У таблиці 6.2 наведено порівняння деяких параметрів задач класифікації та кластеризації.

Таблиця 6.2 - Порівняння класифікації та кластерізації

Характеристика	**Класифікація**	**Кластеризація**
Контрольованість навчання	Контрольоване навчання	Неконтрольоване навчання
Стратегія	Навчання з вчителем	Навчання без вчителя
Наявність позначки класу	Навчальна множина супроводжується міткою, що вказує клас, до якого належить спостереження	Мітки класу навчальної множини невідомі
Підстава для класифікації	Нові дані класифікуються на підставі навчальної множини	Дано множину даних з метою встановлення існування класів або кластерів даних

На рисунку 6.5 схематично представлені задачі класифікації і кластеризації.



а) класифікація б) кластерізація

Рисунок 6.5 – Порівняння задач класифікації та кластеризації

Кластери можуть бути **такими, що не перетинаються**, або **ексклюзивними** (non-overlapping, exclusive), і такими, що **перетинаються** (overlapping). Схематичне зображення таких кластерів дано на рисунку 6.6.

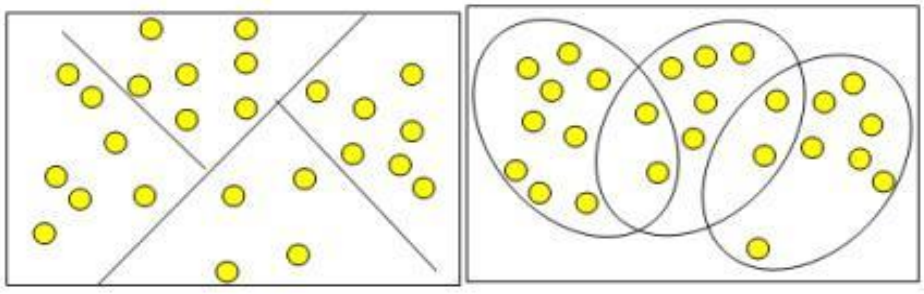


Рисунок 6.6 – Кластери, що не перетинаються і перетинаються

Слід зазначити, що в результаті застосування різних методів кластерного аналізу можуть бути отримані кластери різної форми. Наприклад, можливі кластери «ланцюжкового» типу, коли кластери представлені довгими «ланцюжками», кластери подовженої форми тощо, а деякі методи можуть створювати кластери довільної форми.

Різні методи можуть прагнути створювати кластери певних розмірів (наприклад, малих або великих) або припускати в наборі даних наявність кластерів різного розміру.

Деякі методи кластерного аналізу особливо чутливі до шумів або викидів, інші – менш.

У результаті застосування різних методів кластеризації можуть бути отримані неоднакові результати, це нормально і є особливістю роботи того чи іншого алгоритму.

Ці особливості слід враховувати при виборі методу кластеризації. На цей час розроблено більше сотні різних алгоритмів кластеризації.

Наведемо коротку характеристику підходів до кластеризації.

– *Алгоритми, засновані на поділі даних (Partitioning algorithms), у тому числі ітеративні:*

- поділ об’єктів на *k* кластерів;
- ітеративний перерозподіл об’єктів для поліпшення кластеризації.

– *Ієрархічні алгоритми (Hierarchy Algorithms):*

- агломерація: кожен об’єкт спочатку є кластером, кластери, з’єднуючись один з одним, формують більший кластер і т.д.

– *Методи, засновані на концентрації об’єктів (Density-based methods):*

- засновані на можливості з’єднання об’єктів;
- ігнорують шуми, знаходження кластерів довільної форми.

– *Грід-методи (Grid-based methods):*

- квантування об’єктів в грід-структури.

– *Модельні методи (Model-based):*

- використання моделі для знаходження кластерів, найбільш відповідних даним.

Оцінка якості кластеризації може бути проведена на основі таких процедур:

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

Процес кластеризації залежить від обраного методу і майже завжди є ітеративним. Він може стати захоплюючим процесом і включати безліч експериментів з вибору різноманітних параметрів, наприклад, міри відстані,

типу стандартизації змінних, кількості кластерів і т.д. Однак експерименти не повинні бути самоціллю – адже кінцевою метою кластеризації є отримання змістовних відомостей про структуру досліджуваних даних. Отримані результати вимагають подальшої інтерпретації, дослідження і вивчення властивостей і характеристик об'єктів для можливості точного опису сформованих кластерів.

4. Застосування кластерного аналізу

Кластерний аналіз застосовується в різних областях. Він корисний, коли потрібно класифікувати велику кількість інформації. Огляд багатьох опублікованих досліджень, що проводяться за допомогою кластерного аналізу, дав Хартіган (Hartigan, 1975).

Так, у медицині використовується кластеризація захворювань, лікування захворювань або їх симптомів, а також таксономія пацієнтів, препаратів і т.д. В археології встановлюються таксономії кам'яних споруд і стародавніх об'єктів і т.д. У маркетингу це може бути задача сегментації конкурентів і споживачів. У менеджменті прикладом задачі кластеризації буде розбиття персоналу на різні групи, класифікація споживачів і постачальників, виявлення схожих виробничих ситуацій, при яких виникає брак. У медицині – класифікація симптомів. У соціології задача кластеризації – розбиття респондентів на однорідні групи.

Питання для самоконтролю

1. Назвіть задачі класифікації.
2. Яких правил потрібно дотримуватися при класифікації?
3. Які види класифікацій ви знаєте?
4. Які існують алгоритми машинного навчання? У чому їх відмінність?
5. Проведіть порівняння класифікації та кластерізації.
6. Як проходить оцінка якості кластеризації?

КОНТРОЛЬНІ ПИТАННЯ ДО РОЗДІЛУ І

1. Дайте визначення інтелектуального аналізу даних.
2. Що таке розвідувальний аналіз?
3. Розкрийте поняття даних. Надайте означення понять об’єкт і атрибут, вибірка, залежна і незалежна змінна.
4. Які існують типи змінних?
5. Які типи шкал ви знаєте?
6. У чому полягає задача класифікації? Наведіть практичний приклад.
7. Що таке «навчання з учителем» і «без учителя»? До якого типу відноситься задача класифікації?
8. Задача класифікації є описовою або прогнозуючою, і чому?
9. Навіщо потрібна навчальна і тестова вибірки для розв’язання задачі класифікації?
10. Які існують підходи для поділу вихідної вибірки на навчальну і тестову?
11. Опишіть метод наївної Байєсової класифікації.
12. Розкрийте сутність методу побудови дерева рішень.
13. Опишіть метод опорних векторів.
14. Укажіть особливості методу *k*-найближчих сусідів.
15. Як оцінити якість побудованої моделі класифікації?
16. У чому полягає задача кластеризації? Наведіть практичний приклад.
17. Що таке «навчання з учителем» і «без учителя»? До якого типу відноситься задача кластеризації?
18. Задача кластеризації є описовою або прогнозуючою, і чому?
19. Чим визначається «схожість» об’єктів при розв’язанні задачі кластеризації?
20. Що таке однорівнева і ієрархічна кластеризація?

РОЗДІЛ ІІ. ЗАСТОСУВАННЯ МЕТОДІВ ІНТЕЛЕКТУАЛЬНОГО АНАЛІЗУ ДАНИХ

Тема 7. Задачі Data Mining. Прогнозування та візуалізація

План

1. Задачі прогнозування.
2. Прогнозування і часові ряди.
3. Тренд, сезонність і цикл.
4. Види помилок та прогнозів.
5. Візуалізація інструментів Data Mining.
6. Методи візуалізації.
7. Принципи компонування візуальних засобів.
8. Основні тенденції в області візуалізації.

Мета вивчення теми: вивчити задачі прогнозування; засвоїти особливості візуалізації даних.

Перелік ключових слів та понять із теми

Прогнозування, часовий ряд, ряд динаміки, прогноз, прогностика, похибки прогнозу, тренд, сезонність, цикл, візуалізація

Теоретичні відомості з теми

1.Задачі прогнозування

Задачі прогнозування розв'язуються в найрізноманітніших сферах людської діяльності, таких як наука, економіка, виробництво й безліч інших сфер. Прогнозування є важливим елементом організації управління як окремими господарюючими суб'єктами, так і економікою в цілому.

Розвиток методів прогнозування безпосередньо пов'язаний із розвитком інформаційних технологій, зокрема, із зростанням обсягів збережених даних і ускладненням методів і алгоритмів прогнозування, реалізованих в інструментах Data Mining.

Задача прогнозування, мабуть, може вважатися однією з найбільш складних задач Data Mining, вона вимагає ретельного дослідження вихідного набору даних і методів, що задовольняють аналізу.

Прогнозування (від грецького Prognosis), у широкому розумінні цього слова, визначається як випереджаюче відображення майбутнього**.**

Метою прогнозування є передбачення майбутніх подій.

Прогнозування (forecasting) є однією з задач Data Mining і одночасно одним із ключових моментів при прийнятті рішень.

Прогностика (prognostics) – теорія й практика прогнозування.

Прогнозування спрямоване на визначення тенденцій динаміки конкретного об'єкта або події на основі ретроспективних даних, тобто аналізу

його стану колись і тепер. Отже, розв'язок задачі прогнозування вимагає деякої навчальної вибірки даних.

Прогнозування – установлення функціональної залежності між залежними й незалежними змінними.

Прогнозування є розповсюдженим і затребуваним завданням у багатьох сферах людської діяльності. У результаті прогнозування зменшується ризик прийняття невірних, необґрунтованих або суб'єктивних рішень.

Приклади задач прогнозування: прогноз руху грошових коштів, прогнозування врожайності агрокультури, прогнозування фінансової стабільності підприємства.

Крім економічної й фінансової сфери, задачі прогнозування постають в медицині, фармакології; популярним зараз стає політичне прогнозування.

Загалом розв'язок задачі прогнозування зводиться до розв'язку таких підзадач:

- вибір моделі прогнозування;
- аналіз адекватності й точності побудованого прогнозу.

Порівняння задач прогнозування і класифікації.

Прогнозування подібне із задачею класифікації.

Багато методів Data Mining використовуються для розв'язку задач класифікації і прогнозування. Це, наприклад, лінійна регресія, нейронні мережі, дерева рішень (які, іноді, так і називають – дерева прогнозування й класифікації).

Задачі класифікації й прогнозування мають подібності й відмінності.

Так у чому ж подібність задач прогнозування й класифікації? При розв'язку обох задач використовується двоетапний процес побудови моделі на основі навчального набору та її використання для прогнозування невідомих значень залежної змінної.

Відмінність задач класифікації й прогнозування полягає в тому, що в першій задачі передбачається клас залежної змінної, а в другій – числові значення залежної змінної, пропущені або невідомі (які відносяться до майбутнього).

Повертаючись до прикладу про туристичне агентство, розглянутого у попередній лекції, ми можемо сказати, що визначення класу клієнта є розв'язком задачі класифікації, а прогнозування доходу, який принесе цей клієнт наступного року, буде розв'язком задачі прогнозування.

2. Прогнозування і часові ряди

Прогнозування і часові ряди. Основою для прогнозування служить історична інформація, що зберігається в базі даних у вигляді **часових рядів**.

Існує поняття Data Mining **часових рядів** (Time-Series Data Mining).

На основі ретроспективної інформації у вигляді часових рядів можливий розв'язок різних задач Data Mining.

На рис. 7.1 представлені результати опитування відносно Data Mining часових рядів. Як бачимо, найбільший відсоток (23%) серед розв'язуваних задач займає прогнозування. Далі йдуть класифікація і кластеризація (по 14%),

сегментація й виявлення аномалій (по 9%), виявлення правил (8%). На інші задачі доводиться менш, ніж по 6%.

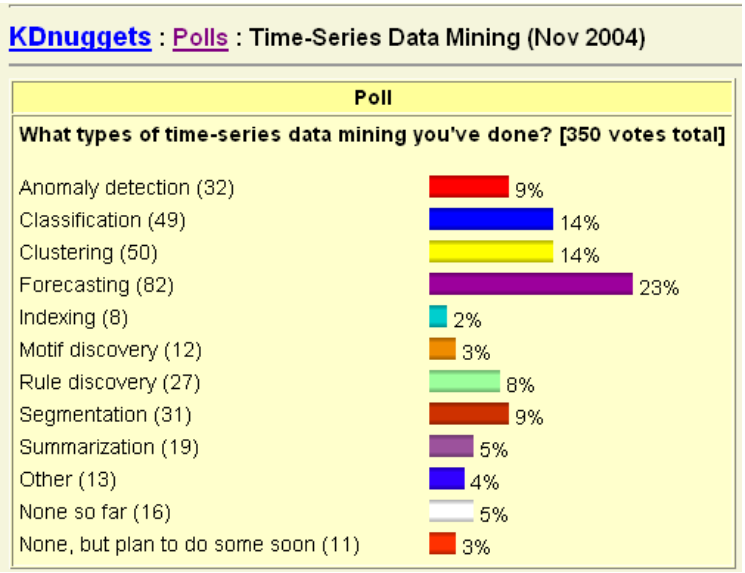


Рисунок 7.1 – Data Mining часових рядів

Однак, щоб зосередитися на понятті прогнозування, ми будемо розглядати часові ряди лише в рамках розв'язку задачі прогнозування.

Приведемо **дві принципові відмінності часового ряду від простої послідовності спостережень:**

- члени часового ряду, на відміну від елементів випадкової вибірки, не є статистично незалежними.
- члени часового ряду не є однаково розподіленими.

Часовий ряд – послідовність спостережуваних значень будь-якої ознаки, упорядкованих у невипадкові моменти часу.

Відмінністю аналізу часових рядів від аналізу випадкових вибірок є припущення про рівні проміжки часу між спостереженнями та їх хронологічний порядок. Прив'язка спостережень до часу відіграє тут ключову роль, тоді як при аналізі випадкової вибірки вона не має ніякого значення.

Типовий приклад часового ряду – дані біржових торгів.

Інформація, накопичена в різноманітних базах даних підприємства, є часовими рядами, якщо вона розташована в хронологічному порядку і зроблена в послідовні моменти часу.

Аналіз часового ряду здійснюється з метою:

- визначення природи ряду;
- прогнозування майбутніх значень ряду.

У процесі визначення структури й закономірностей часового ряду передбачається виявлення: шумів і викидів, тренду, сезонного компонента, циклічного компонента. Визначення природи часового ряду може бути використане як своєрідна «розвідка» даних. Знання аналітика про наявність сезонного компонента необхідне, наприклад, для визначення кількості записів вибірки, яка повинна брати участь у побудові прогнозу.

Аналіз часового ряду ускладнюють **шуми й викиди** (будуть докладно розглянуті в наступних темах курсу). Існують різні методи визначення й фільтрації викидів, що дають можливість виключити їх з метою більш якісного Data Mining.

3.Тренд, сезонність і цикл

Основними складовими часового ряду є тренд і сезонний компонент.

Тренд є систематичним компонентом часового ряду, який може змінюватися в часі.

Трендом називають невипадкову функцію, яка формується під дією загальних або довгочасних тенденцій, що впливають на часовий ряд.

Прикладом тенденції може виступати, наприклад, фактор зростання досліджуваного ринку.

Автоматичного способу виявлення трендів у часових рядах не існує. Але якщо часовий ряд включає монотонний тренд (тобто відзначене його стійке зростання або стійке спадання), аналізувати часовий ряд у більшості випадків неважко.

Існує велика різноманітність постановок задач прогнозування, які можна поділити на дві групи: прогнозування односерійних рядів і прогнозування мультисерійних, або взаємовпливаючих, рядів.

Група прогнозування односерійних рядів включає задачу побудови прогнозу однієї змінної за ретроспективними даними тільки цієї змінної, без врахування впливу інших змінних і факторів.

Група прогнозування мультисерійних, або взаємовпливаючих, рядів включає задачу аналізу, де необхідно враховувати взаємовпливаючі фактори на одну або декілька змінних.

Крім розподілу на класи по односерійності й багатосерійності, ряди також бувають сезонними й несезонними.

Останній розподіл має на увазі наявність або відсутність у часового ряду такої складової як сезонність, тобто включення **сезонного компонента**.

Сезонна складова часового ряду є періодично повторюваним компонентом часового ряду.

Властивість сезонності означає, що через приблизно рівні проміжки часу форма кривої, яка описує поведінку залежної змінної, повторює свої характерні обриси.

Властивість сезонності важлива при визначенні кількості ретроспективних даних, які будуть використовуватися для прогнозування.

Розглянемо простий приклад. На рис. 7.2 наведено фрагмент ряду, який ілюструє поведінку змінної «обсяги продажу товару X» за період, що становить один місяць. При вивченні кривої, наведеної на рисунку, аналітик не може зробити припущень щодо повторюваності форми кривої через рівні проміжки часу.

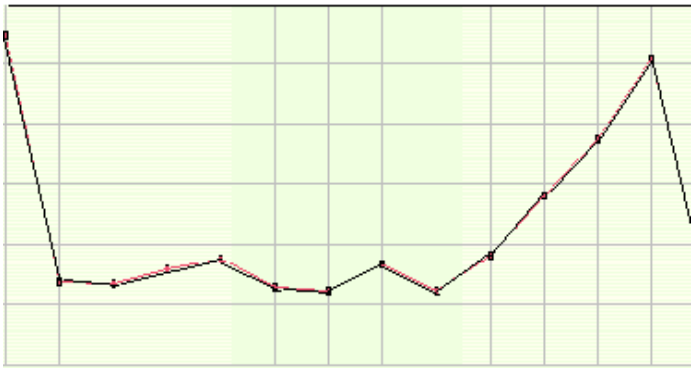


Рисунок 7.2 – Фрагмент часового ряду за сезонний період

Однак при розгляді більш тривалого ряду (за 12 місяців), зображеного на рис. 7.3, можна побачити явну наявність сезонного компонента. Отже, про сезонність продажів можна говорити тільки, коли розглядаються дані за кілька місяців.

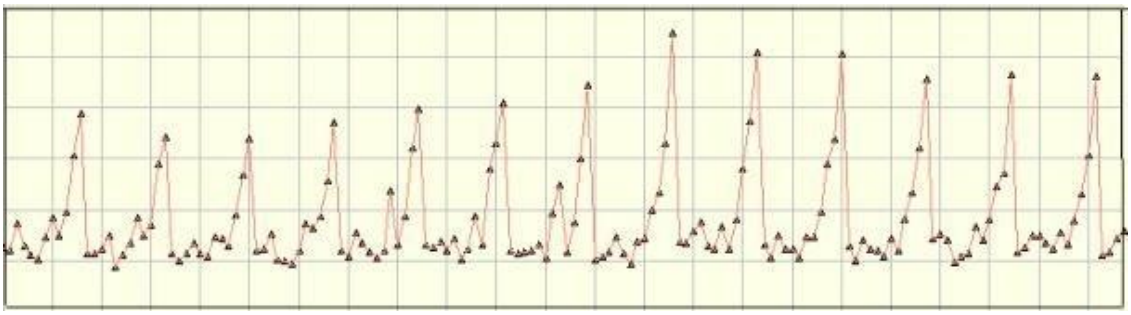


Рисунок 7.3 – Фрагмент часового ряду з 12-ти сезонних періодів

Отже, у процесі підготовки даних для прогнозування аналітикові слід визначити, чи має ряд, який він аналізує, властивість сезонності.

Визначення наявності компоненти сезонності необхідне для того, щоб вхідна інформація мала властивість репрезентативності.

Ряд можна вважати несезонним, якщо при розгляді його зовнішнього вигляду не можна зробити припущень про повторюваність форми кривої через рівні проміжки часу.

Іноді по зовнішньому вигляду кривої ряду не можна визначити, є він сезонним чи ні.

Існує поняття сезонного мультиряду. У ньому кожний ряд описує поведінку факторів, які впливають на залежну (цільову) змінну.

Приклад такого ряду – ряди продажів декількох товарів, що піддаються сезонним коливанням.

При зборі даних і виборі факторів для розв’язку задачі прогнозування в таких випадках слід ураховувати, що вплив обсягів продажів товарів один на одного тут набагато менше, ніж вплив фактору сезонності.

Важливо не плутати поняття сезонного компонента ряду й сезонів природи. Незважаючи на близькість їх звучання, ці поняття відрізняються. Так, наприклад, обсяги продажів морозива влітку набагато більше, ніж в інші сезони, однак це є тенденцією попиту на даний товар.

Дуже часто тренд і сезонність присутні в часовому ряді одночасно.

Приклад. Прибуток фірми зростає протягом декількох років (тобто в часовому ряді присутній тренд); ряд також містить сезонний компонент.

Відмінності циклічного компонента від сезонного:

1. Тривалість циклу, як правило, більше, ніж один сезонний період.
2. Цикли, на відміну від сезонних періодів, не мають певної тривалості.

При виконанні яких-небудь перетворень зрозуміти природу часового ряду значно простіше, такими перетвореннями можуть бути, наприклад, видалення тренда й згладжування ряду.

Перед початком прогнозування необхідно відповісти на такі питання:

1. Що потрібно прогнозувати?
2. У яких часових елементах (параметрах)?
3. З якою точністю прогнозу?

При відповіді на перше питання, ми визначаємо змінні, які будуть прогнозуватися. Це може бути, наприклад, рівень проведення конкретного виду продукції в наступному кварталі, прогноз суми продажу цієї продукції і т.д.

При виборі змінних слід ураховувати доступність ретроспективних даних, переваги осіб, що ухвалюють рішення, остаточну вартість Data Mining.

Часто при розв'язку задач прогнозування виникає необхідність прогнозування не самої змінної, а зміни її значень.

Друге питання при розв'язку задачі прогнозування – визначення таких параметрів:

- періоду прогнозування;
- горизонту прогнозування;
- інтервалу прогнозування.

Період прогнозування – основна одиниця часу, на яку робиться прогноз.

Наприклад, ми прагнемо довідатися дохід компанії через місяць. Період прогнозування для цієї задачі – місяць.

Горизонт прогнозування – це число періодів у майбутньому, які покриває прогноз.

Якщо ми прагнемо дізнатися прогноз на 12 місяців уперед, із даними по кожному місяцю, то період прогнозування в цьому завданні – місяць, горизонт прогнозування – 12 місяців.

Інтервал прогнозування – частота, з якою робиться новий прогноз. Інтервал прогнозування може збігатися з періодом прогнозування.

Рекомендації з вибору параметрів прогнозування. При виборі параметрів необхідно враховувати, що горизонт прогнозування повинен бути не менше, ніж час, який необхідний для реалізації розв'язку, прийнятого на основі цього прогнозу. Тільки в цьому випадку прогнозування буде мати сенс.

Зі збільшенням горизонту прогнозування точність прогнозу, як правило, знижується, а зі зменшенням горизонту – підвищується.

Ми можемо поліпшити якість прогнозування, зменшуючи час, необхідний на реалізацію розв'язку, для якого реалізується прогноз, і, отже, зменшити при цьому горизонт і помилку прогнозування.

При виборі інтервалу прогнозування слід вибирати між двома ризиками: вчасно не визначити зміни в аналізованому процесі й високою вартістю прогнозу. При тривалому інтервалі прогнозування виникає ризик не ідентифікувати зміни, які відбуваються в процесі, при короткому – зростають витрати на прогнозування.

При виборі інтервалу необхідно також ураховувати стабільність аналізованого процесу й вартість проведення прогнозу.

Точність прогнозу, що необхідна для розв'язку конкретної задачі, дуже впливає на прогнозуючу систему. Помилка прогнозу залежить від використовуваної системи прогнозу.

Чим більше ресурсів має така система, тим більше шансів одержати більш точний прогноз. Однак прогнозування не може повністю усунути ризики при прийнятті рішень. Тому завжди враховується можлива помилка прогнозування.

4. Види помилок та прогнозів

Точність прогнозу характеризується помилкою прогнозу.

Найпоширеніші види помилок:

- **Середня помилка (СП).** Вона обчислюється простим усередненням помилок на кожному кроці. Недолік цього виду помилки – позитивні й негативні помилки анулюють одна одну.
- **Середня абсолютна помилка (САП)**. Вона розраховується як середнє абсолютних помилок. Якщо вона дорівнює нулю, то ми маємо досконалий прогноз. У порівнянні із середньою квадратичною помилкою, цей захід «не надає занадто великого значення» викидам.
- **Сума квадратів помилок (SSE)**, середньоквадратична помилка. Вона обчислюється як сума (або середнє) квадратів помилок. Це найбільше часто використовувана оцінка точності прогнозу.
- **Відносна помилка (ВП).** Попередні міри використовували дійсні значення помилок. Відносна помилка виражає якість припасування в термінах відносних помилок.

Види прогнозів. Прогноз може бути короткостроковим, середньостроковим і довгостроковим.

Короткостроковий прогноз являє собою прогноз на кілька кроків уперед, тобто здійснюється побудова прогнозу не більше ніж на 3% від обсягу спостережень або на 1-3 кроку вперед.

Середньостроковий прогноз – це прогноз на 3-5% від обсягу спостережень, але не більш 7-12 кроків уперед; також під цим типом прогнозу розуміють прогноз на один або половину сезонного циклу. Для побудови короткострокових і середньострокових прогнозів цілком підходять статистичні методи.

Довгостроковий прогноз – це прогноз більш ніж на 5% від обсягу спостережень.

При побудові даного типу прогнозів статистичні методи практично не використовуються, крім випадків дуже «гарних» рядів, для яких прогноз можна просто «намалювати».

Дотепер ми розглядали аспекти прогнозування, так чи інакше пов'язані із процесом ухвалення рішення. Існують і інші фактори, які необхідно враховувати при прогнозуванні.

Задача 1. Відомо, що аналізований процес відносно стабільний у часі, зміни відбуваються повільно, процес не залежить від зовнішніх факторів.

Задача 2. Аналізований процес нестабільний і дуже сильно залежить від зовнішніх факторів.

Розв'язок першої задачі повинен бути зосереджений на використанні великої кількості ретроспективних даних. **При розв'язку другої задачі** особливу увагу слід звернути на оцінки фахівця в предметній області, експерта, щоб мати можливість відобразити в прогнозуючій моделі всі необхідні зовнішні фактори, а також приділити час для збору даних по цих факторах (збір зовнішніх даних часто набагато складніший збору внутрішніх даних інформаційної системи). Доступність даних, на основі яких буде здійснюватися прогнозування, – важливий фактор побудови прогнозної моделі. Для можливості виконання якісного прогнозу дані повинні бути представницькими, точними й достовірними.

Методи прогнозування. Серед розповсюджених методів Data Mining, використовуваних для прогнозування, відзначимо **нейронні мережі** й **лінійну регресію**.

Вибір методу прогнозування залежить від багатьох факторів, у тому числі від параметрів прогнозування. Вибір методу слід провадити з обліком усіх специфічних особливостей набору ретроспективних даних і цілей, заради яких він будується.

Програмне забезпечення Data Mining, використовуване для прогнозування, повинно забезпечувати користувача точним і достовірним прогнозом. Однак одержання такого прогнозу залежить не тільки від програмного забезпечення й методів, закладених у його основу, але також і від інших факторів, серед яких повнота й вірогідність вихідних даних, своєчасність і оперативність їх поповнення, кваліфікація користувача.

5. Візуалізація інструментів Data Mining

Візуалізація – це спосіб, який дозволяє побачити кінцевий результат обчислень, організувати керування обчислювальним процесом і навіть повернутися назад до вихідних даних, щоб визначити найбільш раціональний напрямок подальшого руху.

У результаті використання візуалізації створюється графічний образ даних. Застосування візуалізації допомагає в процесі аналізу даних побачити аномалії, структури, тренди. При розгляді задачі прогнозування ми використовували графічне представлення часового ряду й побачили, що в ньому присутній сезонний компонент. У попередній лекції розглянуто задачі **класифікації й кластеризації**, і для ілюстрації розподілу об'єктів у двомірному просторі також використана візуалізацію.

Можна говорити про те, що застосування візуалізації є більш економічним: лінія тренду або скупчення точок на діаграмі розсіювання дозволяє аналітикові набагато швидше визначити закономірності й прийти до потрібного розв'язку. Отже, тут йдеться про використання в Data Mining не символів, а образів.

Головна перевага візуалізації – практично повна відсутність необхідності в спеціальній підготовці користувача. За допомогою візуалізації ознайомитися з інформацією дуже легко, досить лише на неї подивитися.

Хоча найпростіші види візуалізації з'явилися досить давно, її використання зараз тільки набирає популярність. Візуалізація не спрямована винятково на вдосконалення техніки аналізу – за словами Скотта Лейбса, у деяких випадках візуалізація може навіть замінити її.

Візуалізація даних може бути представлена у вигляді: графіків, схем, гістограм, діаграм тощо.

Коротко роль візуалізації можна описати такими її можливостями:

- підтримка інтерактивного й погодженого дослідження;
- допомога в показі результатів;
- використання очей (зору), щоб створювати зорові образи й осмислювати їх.

Погана візуалізація. Результати візуалізації іноді можуть вводити користувача в оману. Приведемо простий приклад поганої візуалізації. Допустимо, ми маємо базу «Прибуток компанії А» за період з 2000 по 2017 рік, вона представлена в таблиці 7.1.

Таблиця 7.1 − Прибуток компанії А

Рік	Прибуток
2000	1100
2001	1101
2002	1104
2003	1105
2004	1106
2017	1007

Побудуємо гістограмму в Excel за цими даними. Гістограмма являє собою візуальне зображення розподілу даних.

Ця інформація відображається за допомогою серії прямокутників або смуг однакової ширини, висота яких указує кількість даних у кожному класі.

Використовуючи всі значення побудови графіка, прийняті за замовчуванням, одержуємо гістограмму, наведену на рис. 7.4.

Рисунок 7.4 – Гістограма прибуткукомпанії
(мінімальне значення вісі *Y* дорівнює 940)

Цей рисунок демонструє значне зростання прибутку компанії А за період з 2000 по 2017 року. Однак, якщо ми звернули увагу на вісь *Y*, що показує величину прибутку, то побачимо, що ця вісь перетинає вісь *X* у значенні, рівному 1096. Фактично, вісь *Y* зі значеннями від 1096 до 1108 вводить користувача в оману. Змінивши значення параметрів, відповідальних за формат вісі *Y*, одержуємо графік, наведений на рис. 7.5.

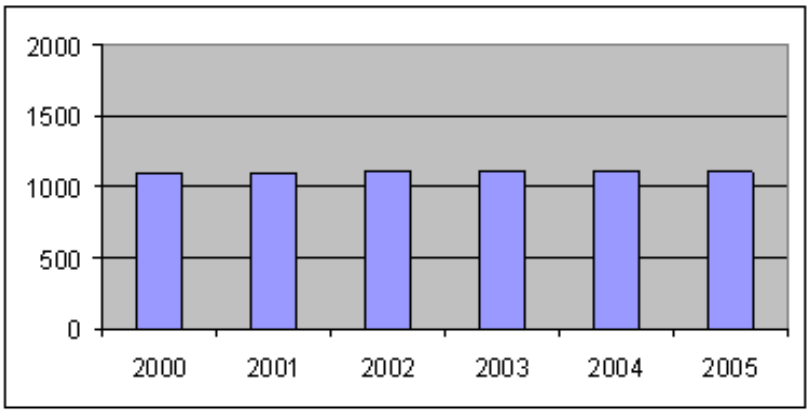


Рисунок 7.5 – Гістограма прибутку компанії
(мінімальне значення вісі Y дорівнює 0)

Вісь *Y* зі значеннями від 0 до 2000 дає користувачеві правильну інформацію про незначну зміну прибутків компанії.

Якщо мова йде про велику розмірність і складності вихідних даних, кошти візуалізації забезпечують їхнє різке зменшення, конденсуючи, можливо, мільйони записів даних у прості, легкі для розуміння й маніпулювання показники. Такі показники називають візуальним або графічним способом показу інформації. Візуалізацію можна вважати ключовим фактором у дослідженні даних, отриманих за допомогою інструментів Data Mining. У таких випадках говорять про візуальний Data Mining.

Зі зростанням кількості даних, що накопичуються, навіть при використанні як завгодно потужних і різносторонніх алгоритмів Data Mining, стає усе складніше «переварювати» і інтерпретувати отримані результати. А, як відомо, одне з положень Data Mining – пошук практично корисних закономірностей. Закономірність може стати практично корисною, тільки якщо її можна осмислити й зрозуміти.

У 1987 році з ініціативи ACM SIGGRAPH IEEE Computer Society Technical Committee of Computer Graphics, у зв'язку з необхідністю використання нових методів, засобів і технологій даних, були сформульовані відповідні завдання напрямку візуалізації.

До способів візуального або графічного представлення даних відносять графіки, діаграми, таблиці, звіти, списки, структурні схеми, карти тощо.

Візуалізація традиційно розглядалася як допоміжний засіб при аналізі даних, однак зараз усе більше досліджень говорить про її самостійну роль.

Традиційні методи візуалізації можуть знаходити таке застосування:

- представляти користувачеві інформацію в наочному вигляді;
- компактно описувати закономірності, властиві вихідному набору даних;
- знижувати розмірність або стискати інформацію;
- відновлювати пробіли в наборі даних;
- знаходити шуми й викиди в наборі даних.

Візуалізація інструментів Data Mining. Кожний з алгоритмів Data Mining використовує певний підхід до візуалізації. У попередніх лекціях ми розглянули ряд методів Data Mining. У ході використання кожного з методів, а точніше, його програмної реалізації, ми одержували якісь візуалізатори, за допомогою яких нам вдавалося інтерпретувати результати, отримані в результаті роботи відповідних методів і алгоритмів.

Для дерев рішень це візуалізатор дерева рішень, список правил, таблиця спряженості.

Для нейронних мереж залежно від інструмента це може бути топологія мережі, графік зміни величини помилки, що демонструє процес навчання.

Для карт Кохонена: карти входів, виходів, інші специфічні карти.

Для лінійної регресії як візуалізатор виступає лінія регресії.

Для кластеризації: дендрограми, діаграми розсіювання.

Діаграми й графіки розсіювання часто використовуються для оцінки якості роботи того або іншого методу.

Усі ці способи візуального представлення або відображення даних можуть виконувати одну з функцій:

- є ілюстрацією побудови моделі (наприклад, представлення структури (графа) нейронної мережі);
- допомагають інтерпретувати отриманий результат;
- є засобом оцінки якості побудованої моделі;
- поєднують перераховані вище функції (дерево розв'язків, дендрограма).

Існує багато різних способів представлення моделей, але графічне їх представлення дає користувачеві максимальну «цінність».

Користувач, у більшості випадків, не є фахівцем у моделюванні, найчастіше він експерт у своїй предметній області. Тому модель Data Mining повинна бути представлена на найбільш природній для нього мові або, хоча б, містити мінімальну кількість різних математичних і технічних елементів.

Отже, доступність є однією з основних характеристик моделі Data Mining. Незважаючи на це, існує й такий розповсюджений і найбільш простий спосіб показу моделі, як «чорний ящик». У цьому випадку користувач не розуміє поведінки тієї моделі, якою користується. Однак, незважаючи на нерозуміння, він одержує результат – виявлені закономірності. Класичним прикладом такої моделі є модель нейронної мережі.

Інший спосіб представлення моделі – представлення її в інтуїтивному, зрозумілому виді. У цьому випадку користувач дійсно може розуміти те, що відбувається «усередині» моделі. Таким чином можна забезпечити його особисту участь у процесі.

Такі моделі забезпечують користувачеві можливість обговорювати її логіку з колегами, клієнтами й іншими користувачами, або пояснювати її.

Розуміння моделі веде до розуміння її змісту. У результаті розуміння зростає довіра до моделі. Класичним прикладом є дерево рішень. Побудоване дерево рішень дійсно поліпшує розуміння моделі, тобто використовуваного інструмента Data Mining.

Крім розуміння, такі моделі забезпечують користувача можливістю взаємодіяти з моделлю, задавати їй питання й одержувати відповіді. Прикладом такої взаємодії є засіб «що, якщо». За допомогою діалогу «система-користувач» користувач може одержати розуміння моделі.

Тепер перейдемо до функцій, які допомагають інтерпретувати й оцінити результати побудови Data Mining моделей. Це всілякі графіки, діаграми, таблиці, списки тощо.

Прикладами засобів візуалізації, за допомогою яких можна оцінити якість моделі, є діаграма розсіювання, таблиця спряженості, графік зміни величини помилки.

Діаграма розсіювання являє собою графік відхилення значень, прогнозованих за допомогою моделі, від реальних. Ці діаграми використовують для безперервних величин. Візуальна оцінка якості побудованої моделі можлива тільки по закінченню процесу побудови моделі.

Таблиця спряженості використовується для оцінки результатів класифікації. Такі таблиці застосовуються для різних методів класифікації. Оцінка якості побудованої моделі тут також можлива тільки по закінченню процесу побудови моделі.

Графік зміни величини помилки. Графік демонструє зміну величини помилки в процесі роботи моделі. Наприклад, у процесі роботи нейронних мереж користувач може спостерігати за зміною помилки на навчальній й тестовій множинах і зупинити навчання для недопущення «перенавчання» мережі. Тут оцінка якості моделі і його зміни може оцінюватися безпосередньо в процесі побудови моделі.

Прикладами засобів візуалізації, які допомагають інтерпретувати результат, є: лінія тренду в лінійній регресії, карти Кохонена, діаграма розсіювання в кластерному аналізі.

6. Методи візуалізації

Методи візуалізації, залежно від кількості використовуваних вимірів, прийнято класифікувати на дві групи:

- представлення даних в одному, двох і трьох вимірах;
- представлення даних у чотирьох і більше вимірах.

Представлення даних в одному, двох і трьох вимірах. До цієї групи методів ставляться добре відомі способи відображення інформації, які доступні для сприйняття людською уявою. Практично будь-який сучасний інструмент Data Mining включає способи візуального представлення із цієї групи.

Відповідно до кількості вимірів представлення це можуть бути такі способи:

- одномірне (univariate) або 1-D;
- двовимірне (bivariate) або 2-D;
- тривимірне, проекційне (projection) або 3-D.

Слід відзначити, що найбільше природно людське око сприймає двомірні представлення інформації.

При використанні двох- і тривимірного представлення інформації користувач має можливість побачити закономірності набору даних:

- його кластерну структуру й розподіл об’єктів на класи (наприклад, на діаграмі розсіювання);
- топологічні особливості;
- наявність трендів;
- інформацію про взаємне розташування даних;
- існування інших залежностей, властивих досліджуваному набору даних.

Якщо набір даних має більше трьох вимірів, то можливі такі варіанти:

- використання багатомірних методів представлення інформації (вони розглянуті нижче);
- зниження розмірності до одно-, двох- або тривимірного представлення. Існують різні способи зниження розмірності, один з них – факторний аналіз – був розглянутий в одній з попередніх лекцій. Для зниження розмірністі й одночасного візуального представлення інформації на двовимірних картах використовуються карти, що самоорганізуються.

Представлення даних в чотирьох і більше вимірах. Представлення інформації в чотиривимірному й більш вимірах недоступні для людського сприйняття. Однак розроблені спеціальні методи для можливості відображення й сприйняття людиною такої інформації.

Найбільш відомі способи багатомірного представлення інформації:

- паралельні координати;
- «особи Чернова»;
- пелюсткові діаграми.

Паралельні координати. У паралельних координатах змінні кодуються по горизонталі, вертикальна лінія визначає значення змінної. Приклад набору даних, представленого в декартових координатах і паралельних координатах, подано на рис. 7.6. Цей метод представлення багатомірних даних був винайдений Альфредом Інселбергом (Alfred Inselberg) в 1985 році.

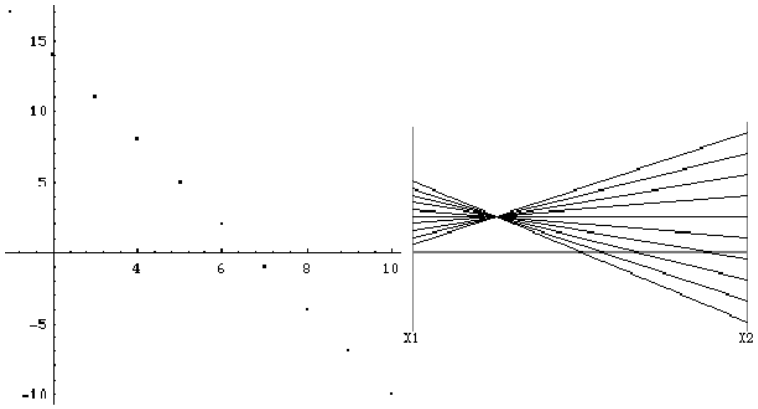


Рисунок 7.6 – Набір даних у декартових координатах
і в паралельних координатах

«Особи Чернова». Основна ідея представлення інформації в «особах Чернова» полягає в кодуванні значень різних змінних у характеристиках або рисах людської особи. Приклад такої «особи» наведено на рис 7.7.

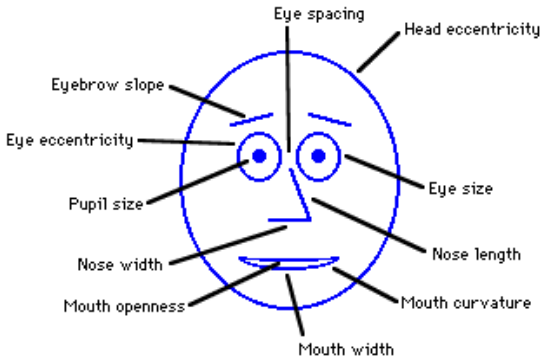


Рисунок 7.7 – «Особа Чернова»

Для кожного спостереження зображується окрема «особа». На кожній «особі» відносні значення змінних представлені як форми й розміри окремих рис особи (наприклад, довжина й ширина носа, розмір очей, розмір зіниці, кут між бровами).

Аналіз інформації за допомогою такого способу відображення заснований на здатності людини інтуїтивно знаходити подібності й відмінності в рисах особи.

На рис. 7.8 представлений набір даних, кожний запис якого виражений у вигляді «Особи Чернова».

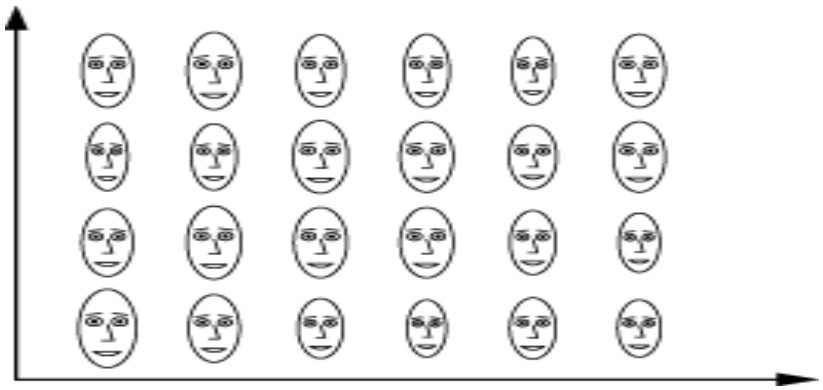


Рисунок 7.8 – Приклад багатомірного зображення даних
за допомогою «осіб Чернова»

Перед використанням методів візуалізації необхідно:

- проаналізувати, чи варто зображувати всі дані або ж лише якусь їхню частину;
- вибрати розміри, пропорції й масштаб зображення;
- вибрати метод, який може найбільше яскаво відобразити закономірності, властиві набору даних.

Багато сучасних засобів аналізу даних дозволяють будувати сотні типів різних графіків і діаграм. Тому вибір методу візуалізації, якщо він самостійно здійснюється користувачем, не такий простий і легкий, як може здатися на перший погляд. Наявність великої кількості засобів візуалізації, представлених в інструменті, який застосовує користувач, може навіть викликати розгубленість.

Ту саму інформацію можна представити за допомогою різних засобів. Для того, щоб засіб візуалізації міг виконувати своє основне призначення – представляти інформацію в простому й доступному для людського сприйняття вигляді – необхідно дотримуватися законів відповідності обраного розв'язку змісту відображуваної інформації і її функціональному призначенню. Іншими словами, потрібно зробити так, щоб при погляді на візуальне представлення інформації можна було відразу виявити закономірності у вихідних даних і приймати на їхній основі рішення.

Серед двомірних і тривимірних засобів найбільше широко відомі лінійні графіки, лінійні, стовпчикові, кругові секторні й векторні діаграми.

Приведемо рекомендації з використання цих найбільш простих і популярних засобів візуалізації.

За допомогою лінійного графіка можна відобразити тенденцію, передати зміни якої-небудь ознаки в часі. Для порівняння декількох рядів чисел такі графіки наносяться на ті самі осі координат.

Гістограму застосовують для порівняння значень протягом деякого періоду або ж співвідношення величин.

Кругові діаграми використовують, якщо необхідно відобразити співвідношення частин і цілого, тобто для аналізу складу або структури явищ. Складові частини цілого зображуються секторами круга. Сектори рекомендують розміщати за їхньою величиною: угорі – найбільший, інші – по рухові годинної стрілки в порядку зменшення їх величини. Кругові діаграми також застосовують

для відображення результатів факторного аналізу, якщо дії всіх факторів є односпрямованими. При цьому кожний фактор відображається у вигляді одного із секторів кола.

Вибір того або іншого засобу візуалізації залежить від поставленого завдання (наприклад, потрібно визначити структуру даних або ж динаміку процесу) і від характеру набору даних.

Якість візуалізації. Сучасні аналітичні засоби, у тому числі й Data Mining, немислимі без якісної візуалізації. У результаті використання засобів візуалізації повинні бути отримані наочні й виразні, ясні й прості зображення, за рахунок використання різноманітних засобів: кольору, контрасту, границь, пропорцій, масштабу тощо.

У зв'язку із зростанням вимог до засобів візуалізації, а також необхідності порівняння їх між собою, в останні роки був сформований ряд принципів якісного візуального представлення інформації.

Принципи Тафта (Tufte's Principles) про графічне представлення даних високої якості говорить:

- надавайте користувачеві найбільшу кількість ідей, у найкоротший час, з найменшою кількістю чорнила на найменшому просторі;
- говоріть правду про дані.

7. Принципи компонування візуальних засобів

Основні принципи компонування візуальних засобів представлення інформації:

1. Принцип лаконічності.
2. Принцип узагальнення й уніфікації.
3. Принцип акценту на основних значимих елементах.
4. Принцип автономності.
5. Принцип структурності.
6. Принцип стадійності.
7. Принцип використання звичних асоціацій і стереотипів.

Принцип лаконічності говорить про те, що засіб візуалізації повинен містити лише ті елементи, які необхідні для повідомлення користувачеві істотної інформації, точного розуміння її значення або прийняття (з імовірністю не нижче допустимої величини) відповідного оптимального розв'язку.

Крім позначених вище принципів, засіб візуалізації повинний мати високу надійність і швидкість, яка влаштує користувача, що приймає на основі цієї інформації рішення.

Представлення просторових характеристик. Окремим напрямком візуалізації є наочне представлення просторових характеристик об'єктів. У більшості випадків такі засоби виділяють на карті окремі регіони й позначають їхніми різними кольорами залежно від значення аналізованого показника.

На рис. 7.9 наведений приклад такої візуалізації в середовищі Mineset, що є, у цьому випадку, інструментом візуального Data Mining. Карта представлена у вигляді графічного інтерфейсу, що відображає дані у вигляді тривимірного

ландшафту довільно визначених і позиціонованих форм (стовпчастих діаграм, кожна з індивідуальними висотою й кольором). Такий спосіб дозволяє наочно показувати кількісні й реляційні характеристики просторово-орієнтованих даних і швидко ідентифікувати в них тренди.

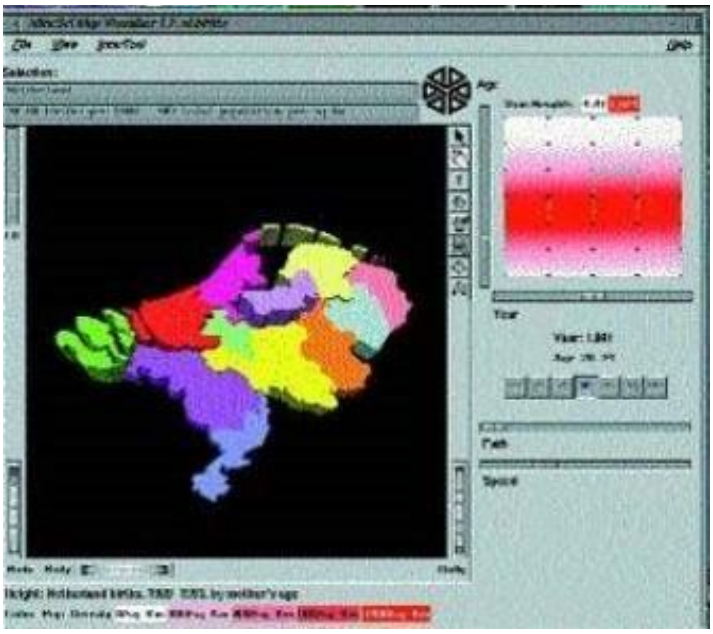


Рисунок 7.9 – Mineset. Ландшафтний візуалізатор

8. Основні тенденції в області візуалізації

Як ми вже відзначали, за допомогою засобів візуалізації підтримуються важливі завдання бізнесу, серед яких – процес прийняття рішень. У зв'язку із цим виникає необхідність переходу засобів візуалізації на більш якісний рівень, який характеризується появою абсолютно нових засобів візуалізації й поглядів на їх функції, а також розвитком ряду тенденцій у цій області.

Серед основних тенденцій в області візуалізації Філіп Рассом (Philip Russom) виділяє:

1. Розробка складних видів діаграм.
2. Підвищення рівня взаємодії з візуалізацією користувача.
3. Збільшення розмірів і складності структур даних, що представляються візуалізацією.

Розробка складних видів діаграм. Більшість візуалізацій даних побудовані на основі діаграм стандартного типу (секторні діаграми, графіки розсіювання тощо). Ці способи є одночасно найбільш елементарними й розповсюдженими. В останні роки перелік видів діаграм, підтримуваних інструментальними засобами візуалізації, суттєво розширився. Оскільки потреби користувачів досить різноманітні, інструменти візуалізації підтримують всілякі типи діаграм. Наприклад, відомо, що бізнес-користувачі віддають перевагу секторним діаграмам і гістограмам, тоді як вчених більше влаштовують візуалізації у вигляді графіків розсіювання й діаграм констеляції. Користувачі, що працюють із геопросторовими даними, більш зацікавлені в картах та інших тривимірних представленнях даних.

Електронні інструментальні панелі, своєю чергою, більш популярні серед керівників, що використовують бізнес-аналітичні технології для контролю над показниками роботи компанії. Такі користувачі потребують наочної візуалізації у вигляді «спідометрів», «термометрів» і «світлофорів».

Засоби створення діаграм і презентаційної графіки призначені головним чином для візуалізації даних. Однак можливості такої візуалізації звичайно вбудовані й у безліч різних інших програм і систем – в інструменти репортинга й OLAP, кошту для Text Mining і Data Mining, а також в Crm-Додатки й додатка для керування бізнесом. Для створення вбудованої візуалізації багато постачальників реалізують візуалізаційну функціональність у вигляді компонентів, що вбудовуються в різні інструменти, додатки, програми й web-сторінки (у тому числі інструментальні панелі й персоналізовані сторінки порталів).

Підвищення рівня взаємодії користувача з візуалізацією. Ще зовсім недавно більша частина коштів візуалізації являла собою статичні діаграми, призначені винятково для перегляду. Зараз широко використовуються динамічні діаграми, уже самі по собі, що є користувацьким інтерфейсом, у якому користувач може прямо й інтерактивно маніпулювати візуалізацією, підбираючи нову виставу інформації.

Наприклад, базова взаємодія дозволяє користувачеві обертати діаграму або змінювати її тип у пошуках найбільш повної представлення даних. Крім того, користувач може міняти візуальні властивості – приміром, шрифти, кольори й рамки. У візуалізаціях складного типу (графіках розсіювання або діаграмах констеляції) користувач може вибирати інформаційні крапки за допомогою миші й переміщати їх, полегшуючи тим самим розуміння представлення даних.

Більш досконалі методи візуалізації даних часто містять у собі діаграму або будь-яку іншу візуалізацію як складений рівень. Користувач може глибшатися (drill down) у візуалізацію, досліджуючи подробиці узагальнених нею даних, або глибшатися в OLAP, Data Mining або інші складні технології.

Складна взаємодія дозволяє користувачеві змінювати візуалізацію для знаходження альтернативних інтерпретацій даних. Взаємодія з візуалізацією має на увазі мінімальний за своєю складністю користувацький інтерфейс, у якому користувач може управляти виставою даних, просто «кликая» на елементи візуалізації, перетаскуючи й поміщаючи представлення об’єктів даних або вибираючи пункти меню. Інструменти OLAP або Data Mining перетворюють безпосередню взаємодія з візуалізацією в один з етапів ітераційного аналізу даних. Кошту Text Mining або керування документами надають такій безпосередній взаємодії характер навігаційного механізму, що допомагає користувачеві досліджувати бібліотеки документів.

Візуальний запит є найбільш сучасною формою складної взаємодії користувача з даними. У ньому користувач може, наприклад, бачити крайні інформаційні крапки графіка розсіювання, вибирати їхньою мишкою й одержувати нові візуалізації, що представляють саме ці крапки. Додаток візуалізації даних генерує відповідна мова запиту, управляє прийняттям запиту

базою даних і візуально представляє результуючу безліч. Користувач може сфокусуватися на аналізі, не відволікаючись на складання запиту.

Збільшення розмірів і складності структур даних, що представляються візуалізацією. Елементарна секторна діаграма або гістограма візуалізує прості послідовності числових інформаційних крапок. Однак нові вдосконалені типи діаграм здатні візуалізувати тисячі таких крапок і навіть складні структури даних – наприклад, нейронні мережі.

Скажемо, кошту OLAP (а також інструменти генерації запитів і випуску звітів) уже давно підтримують діаграми для своїх он-лайнових звітів. Нові візуалізаційні програми обновляють контент за рахунок періодично повторюваного зчитування даних. Візуалізаційні програми відслідковування лінійних процесів (коливання фондового ринку, показники роботи комп'ютерних систем, сейсмограми, сітки корисності) потребують завантаження даних у режимі реального часу або близькому до нього режимі для зручності користувачів.

Користувачі інструментів Data Mining звичайно аналізують дуже великі набори чисельних даних. Традиційні типи діаграм для бізнесу (секторні діаграми й гістограми) погано справляються з показом тисяч інформаційних точок. Тому інструменти Data Mining майже завжди підтримують якусь форму візуалізації даних, здатну відображати структури й закономірності досліджуваних наборів даних, відповідно до тих аналітичних підходів, які використовується в інструменті.

Крім того, що візуалізація підтримує обробку структурованих даних, вона також є ключовим засобом представлення схем так званих неструктурованих даних, наприклад текстових документів.

Text Mining. Зокрема, засоби Text Mining можуть здійснювати парсинг більших пакетів документів і формувати предметні покажчики понять і тем, освітлених у цих документах. Коли предметні покажчики створені за допомогою нейронної мережевої технології, користувачеві непросто продемонструвати їх без деякої форми візуалізації даних.

Питання для самоконтролю

1. Назвіть мету та визначення поняттю прогнозування.
2. Дайте визначення поняттю часовий ряд.
3. Які існують помилки прогнозу?
4. Які існують види прогнозів? Чим вони відрізняються?
5. Дайте визначення поняттю візуалізації. Для чого вона використовується?
6. Які існують способи багатомірного представлення інформації?

Тема 8. Основи аналізу даних

План

1. Підготовчі етапи процесу Data Mining.
2. Дублювання даних.
3. Очищення даних.
4. Етапи очищення даних.

Мета вивчення теми: вивчити основи інтелектуального аналізу даних; засвоїти, в чому полягає процес очищення даних.

Перелік ключових слів та понять із теми

Інтелектуальний аналіз даних, дослідження, предметна область, завдання, дані, набір даних, очищення даних, дублювання даних, викид

Теоретичні відомості з теми

1. Підготовчі етапи процесу Data Mining.

Процес Data Mining є свого роду дослідженням. Як будь-яке дослідження, цей процес складається з певних етапів, що включають елементи порівняння, типізації, класифікації, узагальнення, абстрагування, повторення.

Процес Data Mining нерозривно пов'язаний із процесом прийняття рішень.

Процес Data Mining будує модель, а в процесі прийняття рішень ця модель експлуатується.

Розглянемо традиційний процес Data Mining. Він включає такі етапи:

- аналіз предметної області;
- постановка задачі;
- підготовка даних;
- побудова моделей;
- перевірка й оцінка моделей;
- вибір моделі;
- застосування моделі;
- корекція й відновлення моделі.

Етап 1. Аналіз предметної області.

Дослідження – це процес пізнання певної предметної області, об'єкта або явища з певною метою.

Процес дослідження полягає в спостереженні властивостей об'єктів з метою виявлення й оцінки важливих, з погляду суб'єкта-дослідника, закономірних відносин між показниками даних властивостей.

Вирішення будь-якого завдання у сфері розробки програмного забезпечення повинне починатися з вивчення предметної області.

Предметна область – це подумки обмежена область реальної дійсності, що підлягає опису або моделюванню й дослідженню.

Предметна область складається з об'єктів, що різняться за властивостями й перебувають у певних відносинах між собою або взаємодіють яким-небудь

чином. Дослідникові необхідно вміти виділити їхню частину, необхідну для використання. Наприклад, при розв'язанні задачі «Чи видавати кредит?» важливими є всі дані про приватне життя клієнта, аж до того, чи має роботу подружжя, чи є в клієнта неповнолітні діти, який його рівень освіти тощо. Для розв'язку іншого завдання банківської діяльності ці дані будуть абсолютно неважливі. Отже, важливість даних залежить від вибору предметної області.

У процесі вивчення предметної області повинна бути створена її модель. Знання з різних джерел повинні бути формалізовані за допомогою яких-небудь засобів.

Це можуть бути текстові описи предметної області або спеціалізовані графічні нотації. Існує велика кількість методик опису предметної області, *наприклад,* методика структурного аналізу SADT і заснована на ньому IDEF0, діаграми потоків даних Гейна-Сарсона, методика об'єктно-орієнтованого аналізу UML та ін. Модель предметної області описує процеси, що відбуваються в предметній області, і дані, які в цих процесах використовуються.

Це перший етап процесу Data Mining. Але від того, наскільки вірно змодельована предметна область, залежить успіх подальшої розробки додатка Data Mining.

Етап 2. Постановка задачі.

Постановка задачі Data Mining включає такі кроки:

- формулювання задачі;
- формалізація задачі.

Постановка задачі включає також опис статичної та динамічної поведінки досліджуваних об'єктів.

Приклад. При просуванні нового товару на ринок необхідно визначити, яка група клієнтів фірми буде найбільш зацікавлена в цьому товарі.

Опис статики – це опис об'єктів та їх властивостей.

Приклад. Клієнт є об'єктом. Властивості об'єкта «клієнт»: родинний стан, дохід за попередній рік, місце проживання.

При описі динаміки описується поведінка об'єктів і причини, що впливають на їхню поведінку.

Приклад. Клієнт купує товар А. З появою нового товару В клієнт уже не купує товар А, а купує тільки товар В. Поява товару В змінила поведінку клієнта. Динаміка поведінки об'єктів часто описується разом зі статикою.

Технологія Data Mining не може замінити аналітика й відповісти на питання, що не були задані. Тому постановка задачі є необхідним етапом процесу Data Mining, оскільки саме на цьому етапі визначається, яку ж задачу необхідно розв'язати. Іноді етапи аналізу предметної області й постановки задачі поєднують в один етап.

Етап 3. Підготовка даних.

Ціль етапу: розробка бази даних для Data Mining (поняття «даних» було розглянуто в темі 2).

Підготовка даних є найважливішим етапом, від якості виконання якого залежить можливість одержання якісних результатів усього процесу Data Mining.

Крім того, слід пам’ятати, що на етап підготовки даних, за деякими оцінками, може бути витрачене до 80% усього часу, відведеного на проект.

Розглянемо докладно цей етап.

1. *Визначення й аналіз вимог до даних.* На цьому кроці здійснюється так зване моделювання даних, тобто визначення й аналіз вимог до даних, які необхідні для здійснення Data Mining. При цьому вивчаються питання розподілу користувачів (географічне, організаційне, функціональне); питання доступу до даних, які потрібні для аналізу, необхідність у зовнішніх або внутрішніх джерелах даних; а також аналітичні характеристики системи (виміру даних, основні види вихідних документів, послідовність перетворення інформації тощо).

2. *Збір даних.* Наявність в організації сховища даних робить аналіз простіше й ефективніше, його використання, з погляду вкладень, обходиться дешевше, ніж використання окремих баз даних або вітрин даних. Однак далеко не всі підприємства оснащені сховищами даних. У цьому випадку джерелом для вхідних даних є оперативні, довідкові й архівні БД, тобто дані з існуючих інформаційних систем.

Також для Data Mining може знадобитися інформація з інформаційних систем керівників, зовнішніх джерел, паперових носіїв, а також знання експертів або результати опитувань.

Слід пам’ятати, що в процесі підготовки даних аналітики й розроблювачі не повинні прив’язуватися до показників, які є в наявності, й описати максимальну кількість факторів і ознак, що впливають на процес, що аналізується.

На цьому етапі здійснюється кодування деяких даних. Допустимо, одним з атрибутів клієнта є рівень доходу, який повинен бути представленим у системі одним із значень: дуже низьким, низьким, середнім, високим, дуже високим.

Необхідно визначити градації рівня доходу, у цьому процесі буде потрібно співробітництво аналітика з експертом у предметній області. Можливо, для таких перетворень даних буде потрібно написання спеціальних процедур.

Визначення необхідної кількості даних. При визначенні необхідної кількості даних слід ураховувати, чи є дані впорядкованими чи ні.

Якщо дані впорядковані й ми маємо справу з тимчасовими рядами, бажано знати, чи включає такий набір даних сезонну/циклічну компоненту. У випадку присутності в **наборі** даних сезонної/циклічної компоненти, необхідно мати дані як мінімум за один сезон/цикл.

Якщо дані не впорядковані, тобто події з набору даних не зв’язані за часом, у ході збору даних слід дотримуватися таких правил.

Кількість записів у наборі. Недостатня кількість записів у наборі даних може стати причиною побудови некоректної моделі. З погляду статистики, точність моделі збільшується зі збільшенням кількості досліджуваних даних. Можливо, деякі дані є застарілими або описують якусь нетипову ситуацію, їх потрібно виключити з бази даних. Алгоритми, що використовуються для побудови моделей на надвеликих базах даних, повинні бути масштабованими.

Співвідношення кількості записів у наборі й кількості вхідних змінних. При використанні багатьох алгоритмів необхідно певне (бажане) співвідношення вхідних змінних і кількості спостережень. Кількість записів (прикладів) у наборі даних повинна бути значно більшою кількості факторів (змінних).

Набір даних повинен бути репрезентативним і представляти якнайбільше можливих ситуацій. Пропорції подання різних прикладів у наборі даних повинні відповідати реальній ситуації.

3. *Попередня обробка даних.* Аналізувати можна і якісні, і неякісні дані. Результат буде досягнутий і в тому, і в іншому випадку. Для забезпечення якісного аналізу необхідне проведення попередньої обробки даних, яка є необхідним етапом процесу Data Mining.

Оцінювання якості даних. Дані, отримані в результаті збору, повинні відповідати певним критеріям якості. Таким чином, можна виділити важливий підетап процесу Data Mining – оцінювання якості даних.

Якість даних (Data quality) – це критерій, що визначає повноту, точність, своєчасність і можливість інтерпретації даних.

Дані можуть бути високої якості й низької якості, останні – це так звані брудні або «погані» дані.

Дані високої якості – це повні, точні, своєчасні дані, які піддаються інтерпретації.

Такі дані забезпечують одержання якісного результату: знань, які зможуть підтримувати процес прийняття рішень.

Про важливість обговорюваної проблеми говорить той факт, що «серйозне відношення до якості даних» посідає перше місце серед десяти основних тенденцій, що прогнозуються на початку 2017 року в області Business Intelligence і Сховищ даних компанією Knightsbridge Solutions. Цей прогноз був зроблений в січні 2017 року, а в червні 2017 року Даффі Брансон (Duffie Brunson), один із керівників компанії Knightsbridge Solutions, проаналізував якість даних раніше прогнозів.

Прогноз. Багато компаній стали звертати більше уваги на якість даних, оскільки низька якість коштує грошей у тому розумінні, що веде до зниження продуктивності, прийняттю неправильних бізнес-рішень і неможливості одержати бажаний результат, а також ускладнює виконання вимог законодавства. Тому компанії дійсно мають намір вживати конкретні дії для вирішення проблем якості даних.

Реальність. Дана тенденція зберігається, особливо в індустрії фінансових послуг. У першу чергу це стосується фірм, що намагаються виконувати угоду Basel II. Неякісні дані не можуть використовуватися в системах оцінки ризиків, які застосовуються для установки цін на кредити й обчислення потреб організації в капіталі. Цікаво відзначити, що суттєво змінилися погляди на способи вирішення проблеми якості даних. Спочатку менеджери звертали основну увагу на інструменти оцінки якості, вважаючи, що «власник» даних повинен вирішувати проблему на рівні джерела, наприклад, очищаючи дані й перенавчаючи співробітників. Але зараз їх погляди суттєво змінилися. Поняття якості даних набагато ширше, чим просто їх акуратне введення в систему на

першому етапі. Сьогодні вже є розуміння, що якість даних повинна забезпечуватися процесами витягу, перетворення й завантаження (Extraction, Transformation, Loading – ETL), а також одержання даних із джерел, які готують дані для аналізу.

Розглянемо поняття якості даних більш детально. Дані низької якості, або брудні дані – це відсутні, неточні дані з погляду практичного застосування (наприклад, представлені в невірному форматі, не відповідному до стандарту). Брудні дані з'явилися не сьогодні, вони виникли одночасно із системами введення даних.

Брудні дані можуть з'явитися з різних причин, таким як помилка при введенні даних, використання інших форматів подання або одиниць виміру, невідповідність стандартам, відсутність своєчасного відновлення, невдале відновлення всіх копій даних, невдале видалення записів-дублікатів і т.д.

Необхідно оцінити вартість наявності брудних даних; інакше кажучи, наявність брудних даних може дійсно призвести до фінансових втрат і юридичної відповідальності, якщо їх наявність не запобігається або вони не виявляються й не очищаються.

Описані різні типи брудних даних, серед них виділені такі групи:

- брудні дані, які можуть бути автоматично виявлені й очищені;
- дані, поява яких може бути відвернена;
- дані, які непридатні для автоматичного виявлення й очищення;
- дані, появу яких неможливо запобігти.

Тому важливо розуміти, що спеціальні кошти очищення можуть упоратися не з усіма видами брудних даних.

Найпоширеніші види брудних даних такі:

- пропущені значення;
- дублікати даних;
- шуми й викиди.

Пропущені значення (Missing Values).

Деякі значення даних можуть бути пропущені у зв'язку з тим, що:

- дані взагалі не були зібрані (наприклад, при анкетуванні схований вік);
- деякі атрибути можуть бути незастосовні для деяких об'єктів (наприклад, атрибут «річний дохід» не застосуємо до дитини).

Що можна зробити із пропущеними даними:

- виключити об'єкти із пропущеними значеннями з обробки;
- розрахувати нові значення для пропущених даних;
- ігнорувати пропущені значення в процесі аналізу;
- замінити пропущені значення на можливі значення.

2. Дублювання даних

Набір даних може включати продубльовані дані, тобто **дублікати –** записи з однаковими значеннями всіх атрибутів.

Наявність дублікатів у наборі даних може бути способом підвищення значимості деяких записів. Така необхідність іноді виникає для особливого виділення певних записів з набору даних. Однак у більшості випадків, продубльовані дані є результатом помилок при підготовці даних.

Існує два варіанти обробки дублікатів. При першому варіанті віддаляється вся група записів, що містить дублікати. Цей варіант використовується в тому випадку, якщо наявність дублікатів викликає недовіру до інформації, повністю її знецінює. Другий варіант полягає в заміні групи дублікатів на один унікальний запис.

Шуми й викиди.

Викиди – різко одмінні об’єкти або спостереження в наборі даних.

Шуми й викиди є досить загальною проблемою в аналізі даних. Викиди можуть являти собою окремі спостереження або бути об’єднаними в якісь групи. Завдання аналітика не тільки їх виявити, але й оцінити ступінь їх впливу на результати подальшого аналізу. Якщо викиди є інформативною частиною аналізованого набору даних, використовують робастні методи й процедури.

Досить поширена практика проведення двоетапного аналізу – з викидами та з їхньою відсутністю – і порівняння отриманих результатів.

Різні методи Data Mining мають різну чутливість до викидів, цей факт необхідно враховувати при виборі методу аналізу даних. Також у деякі інструменти Data Mining мають бути вбудовані процедури очищення від шумів і викидів. Візуалізація даних дозволяє представити дані, у тому числі й викиди, у графічному виді. Приклад наявності викидів зображений на діаграмі розсіювання на рис. 8.1, де видні кілька спостережень, що різко відрізняються від інших (спостережень, що перебувають на великій відстані від більшості).

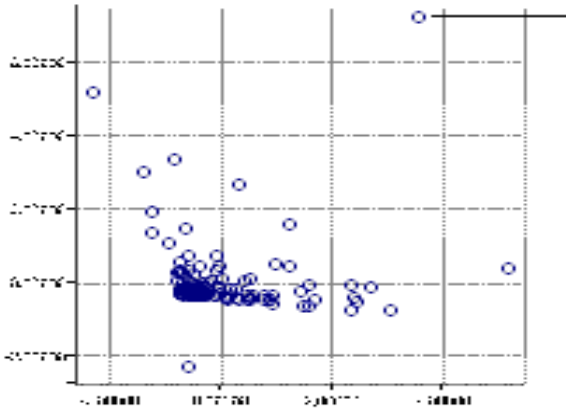


Рисунок 8.1 – Приклад набору даних з викидами

Очевидно, що результати Data Mining на основі брудних даних не можуть вважатися надійними й корисними. Однак наявність таких даних не обов’язково означає необхідність їх очищення або ж запобігання появи. Завжди повинен бути розумний вибір між наявністю брудних даних і вартістю й/або часом, необхідним для їхнього очищення.

3. Очищення даних

Очищення даних (data cleaning, data cleansing або scrubbing) займається виявленням та видаленням помилок і невідповідностей у даних з метою поліпшення якості даних.

Проблеми з якістю зустрічаються в окремих наборах даних – таких як файли й бази даних. Коли інтеграції підлягає безліч джерел даних (наприклад, у Сховищах, інтегрованих системах баз даних або глобальних інформаційних Інтернет-системах), необхідність в очищенні даних суттєво зростає. Це відбувається тому, що джерела часто містять розрізнені дані в різному представленні. Для забезпечення доступу до точних і погоджених даних необхідна консолідація різних представлень даних і виключення інформації, що дублюється. Спеціальні кошти очищення звичайно мають справу з конкретними областями – в основному це імена й адреси – або ж з виключенням дублікатів.

Перетворення забезпечуються або у формі бібліотеки правил, або користувачем в інтерактивному режимі. Перетворення даних можуть бути автоматично отримані за допомогою коштів узгодження схеми.

Метод очищення даних повинен задовольняти таким критеріям:

1. Повинен виявляти й видаляти всі основні помилки й невідповідності і в окремих джерелах даних, і при інтеграції декількох джерел.

2. Метод повинен підтримуватися певними інструментами, щоб скоротити обсяги ручної перевірки й програмування, і бути гнучким у плані роботи з додатковими джерелами.

3. Очищення даних не повинне проводитися у відриві від зв’язаних зі схемою перетворення даних, виконуваних на основі складних метаданих.

4. Функції маппірования для очищення й інших перетворень даних повинні бути визначені декларативним чином та підходити для використання в інших джерелах даних і в обробці запитів.

5. Інфраструктура технологічного процесу повинна особливо інтенсивно підтримуватися для Сховищ даних, забезпечуючи ефективне й надійне виконання всіх етапів перетворення для безлічі джерел і більших наборів даних.

Сьогодні інтерес до очищення даних зростає. Ціла низка дослідницьких груп займається загальними проблемами, пов’язаними з очищенням даних, у тому числі, зі специфічними підходами до Data Mining і перетворенням даних на підставі зіставлення схеми. Останнім часом деякі дослідження торкнулися єдиного, більш складного підходу до очищення рядів даних, який включає аспекти перетворення даних, специфічні оператори та їх реалізації.

4. Етапи очищення даних

Очищення даних включає такі етапи:

1. Аналіз даних.
2. Визначення порядку й правил перетворення даних.
3. Підтвердження.
4. Перетворення.
5. Протитечія очищених даних.

Етап 1. Аналіз даних. Докладний аналіз даних необхідний для виявлення підлягаючих видаленню видів помилок і невідповідностей. Тут можна

використовувати і ручну перевірку даних або їх шаблонів, і спеціальні програми для одержання метаданих про властивості даних та визначення проблем якості.

Етап 2. Визначення порядку й правил перетворення даних. Залежно від числа джерел даних, ступені їх неоднорідності й забруднення, дані можуть вимагати досить великого перетворення й очищення. Іноді для відображення джерел загальної моделі даних використовується трансляція схеми; для сховищ даних звичайно використовується реляційне представлення. Перші кроки з очищення можуть уточнити або змінити опис проблем окремих джерел даних, а також підготувати дані для інтеграції. Подальші кроки повинні бути спрямовані на інтеграцію схеми/даних і усунення проблем численних елементів, наприклад, дублікатів. Для сховищ у процесі роботи з визначення ETL повинні бути визначені методи контролю й потік даних, що підлягає перетворенню й очищенню.

Перетворення даних, що зв'язані зі схемою, так само як і етапи очищення, повинні, наскільки можливо, визначатися за допомогою декларативного запиту й мови маппірования, забезпечуючи, таким чином, автоматичну генерацію коду перетворення. До того ж, у процесі перетворення повинна існувати можливість запуску написаного користувачем коду очищення й спеціальних коштів. Етапи перетворення можуть вимагати зворотного зв'язку з користувачем за тими елементами даних, для яких відсутня вбудована логіка очищення.

Етап 3. Підтвердження. На цьому етапі визначається правильність і ефективність процесу перетворення. Це здійснюється шляхом тестування й оцінювання на прикладі або на копії даних джерела, – щоб з'ясувати, чи необхідно якось поліпшити ці визначення. При аналізі, проектуванні й підтвердженні може знадобитися безліч ітерацій, наприклад, у зв'язку з тим, що деякі помилки стають помітні тільки після проведення певних перетворень.

Етап 4. Перетворення. На цьому етапі здійснюється виконання перетворень або в процесі ETL для завантаження й відновлення Сховища даних, або при відповіді на запити по безлічі джерел.

Етап 5. Протитечія очищених даних. Після того, як помилки окремого джерела вилучені, забруднені дані у вихідних джерелах повинні замінитися на очищені, для того, щоб поліпшені дані потрапили також в успадковані додатки й надалі при витягу не вимагали додаткового очищення. Для Сховищ очищені дані перебувають в області зберігання даних.

Такий процес перетворення вимагає більших обсягів метаданих (схем, характеристик даних рівня схеми, визначень технологічного процесу та ін.). Для погодженості, гнучкості й спрощення використання в інших випадках, ці метадані повинні зберігатися в депозиторії на основі СУБД. Для підтримки якості даних докладна інформація про процес перетворення повинна записуватися як у депозитарій, так і в трансформовані елементи даних (інформація про повноту й актуальність вихідних даних, походження інформації про першоджерело трансформованих об'єктів, зроблені з ними зміни). Наприклад, на мал. 3 похідна таблиця Споживачі містить атрибути Ідентифікатор і Номер, дозволяючи простежити шлях вихідних записів.

Далі докладно описуються можливі методи аналізу даних (виявлення конфліктів), визначення перетворень і розв'язання конфліктів. Конфлікти найменувань звичайно дозволяються шляхом перейменування; структурні конфлікти вимагають часткового перебудування й уніфікації вихідних схем.

Сьогодні ринок програмного забезпечення пропонує великий вибір засобів, метою яких є перетворення й очищення даних.

Розглянемо дві класифікації таких засобів.

Эрхард Рам (Erhard Ram) і Хонг Ганьби До (Hong Hai Do) визначають таку класифікацію засобів очищення й відповідних їм інструментів.

1. Засоби аналізу й модернізації даних.
2. Спеціальні засоби очищення:
   - очищення специфічної області;
   - виключення дублікатів.
3. Інструменти ETL.

1. Засоби аналізу й модернізації даних.

Засоби аналізу й модернізації, що обробляють дані з метою виявлення помилок, невідповідностей і визначення необхідних, що очищають перетворення, згідно із цією класифікацією, можуть бути розділені на засоби профайлинга даних і засоби Data Mining.

Профайлинг даних. MIGRATIONARCHITECT (Evoke Software) є одним із деяких комерційних інструментів цієї категорії. Для кожного атрибута він визначає такі метадані: тип даних, довжину, безліч елементів, дискретні значення та їх процентне відношення, мінімальні й максимальні значення, втрачені значення й унікальність. MIGRATIONARCHITECT також може допомогти в розробці цільової схеми для міграції даних.

Засоби Data Mining. Наприклад, WIZRULE (Wizsoft) і DATAMININGSUITE (Information Discovery) виводять відносини між атрибутами та їх значеннями, обчислюють рівень вірогідності, що відображає число кваліфікуючих рядів.

WIZRULE може відображати три види правил: 1) математичну формулу, 2) правило if-then («якщо-то») і 3) правило правопису. Ці правила відсівають невірно написані імена, наприклад, «значення Edinburgh 52 рази зустрічається в полі Споживач; у 2-му рядку містять однакові значення». WIZRULE також автоматично вказує на відхилення від набору виявлених правил як на можливі помилки.

Засоби модернізації даних, наприклад, INTEGRITY (Vality), використовують виявлені шаблони й правила для визначення й виконання перетворень, що очищають, тобто модернізують успадковані дані. В INTEGRITY елементи даних зазнають низку обробок – типізація, аналіз шаблонів і частот та ін.

Результатом цих дій є табличне представлення вмісту полів, їхніх шаблонів і частот, залежно від того, які шаблони можна вибрати для стандартизації даних. Для визначення перетворень, що очищають, INTEGRITY пропонує мову з набором операторів для перетворень стовпців (наприклад, переміщення,

розщеплення, видалення) і рядків. INTEGRITY ідентифікує й консолідує запис за допомогою методу статистичної відповідності. При обчисленні оцінок для упорядкування відповідностей, за якими користувач відбирає справжні дублікати, використовуються зважені коефіцієнти.

2. Спеціальні засоби очищення.

Спеціальні засоби очищення звичайно мають справу з конкретними областями – в основному це імена й адреси – або ж із виключенням дублікатів. Перетворення або забезпечуються заздалегідь, у формі бібліотеки правил, або в інтерактивному режимі, користувачем. Перетворення даних можуть бути автоматично отримані й за допомогою засобів узгодження схеми.

Низка засобів орієнтована на специфічну область – наприклад, на очищення даних по іменах і адресах або на специфічні фази очищення – наприклад, аналіз даних або виключення дублікатів. Завдяки своїй обмеженості застосування, спеціалізовані засоби звичайно дуже ефективні, однак для роботи із широким спектром проблем перетворення й очищення вони потребують доповнення іншими інструментами.

2.1. Очищення специфічної області.

Імена й адреси записані в різних джерелах і звичайно мають безліч елементів, тому пошук відповідностей їх конкретному споживачеві має велике значення для керування відносинами із клієнтами. Ряд комерційних інструментів, наприклад IDCENTRIC (First Logic), PUREINTEGRATE (Oracle), QUICKADDRESS (QAS Systems),

REUNION (Pitney Bowes) і TRILLIUM (Trillium Software), призначені для очищення саме таких даних. Вони містять відповідні методи: наприклад, метод витягу й перетворення імен і адрес в окремі стандартні елементи, перевірку допустимості назв вулиць, міст і індексів, разом із можливостями зіставлення на основі очищених даних. Вони включають величезну бібліотеку визначених правил щодо проблем, що часто зустрічаються в даних такого роду. Приміром, модуль витяг TRILLIUM і модуль зіставлення містять понад 200000 бізнес-правил. Ці інструменти забезпечують і можливості настроювання або розширення бібліотеки правил за рахунок правил, визначених користувачем для власних специфічних випадків.

2.2. Виключення дублікатів.

Прикладами засобів для виявлення й видалення дублікатів є DATACLEANSER (EDD), MERGE/PURGELIBRARY (Sagent/Qmsoftware), MATCHIT (Helpitsystems) і MASTERMERGE (Pitney Bowes). Звичайно вони вимагають, щоб джерело даних уже було очищене й підготовлене для узгодження. Ними підтримується кілька підходів до узгодження значень атрибутів; а такі засоби як DATACLEANSER і MERGE/PURGE LIBRARY дозволяють також інтегрувати правила узгодження, визначені користувачем.

3. Інструменти ETL.

Засоби ETL забезпечують можливість складних перетворень і більшої частини технологічного процесу перетворення й очищення даних. Загальною

проблемою засобів ETL є обмежені за рахунок власних API і форматів метаданих можливості взаємодії, що ускладнюють спільне використання різних засобів.

Багато комерційних інструментів підтримують процес ETL для Сховищ даних на комплексному рівні, наприклад, COPYMANAGER (Information Builders), DATASTAGE (Informix/Ardent), EXTRACT (ETI), POWERMART (Informatica), DECISIONBASE (CA/Platinum), DATATRANSFORMATIONSERVICE (Microsoft), METASUITE (Minerva/Carleton), SAGENTSOLUTIONPLATFORM (Sagent) і AREHOUSEADMINISTRATOR (SAS). Для однакового керування всіма метаданими по джерелах даних, цільових схемах, маппіруваннях, скриптах і т.д. вони використовують репозиторій на основі СУБД. Схеми й дані вилучаються з оперативних джерел даних як через «рідний» файл і шлюзи СУБД DBMS, так і через стандартні інтерфейси – наприклад, ODBC і EDA. Перетворення даних визначаються через простий графічний інтерфейс. Для визначення індивідуальних кроків маппірування звичайно існує власна мова правил і комплексна бібліотека визначених функцій перетворення. Ці засоби підтримують і повторне використання існуючих перетворених розв'язків, наприклад, зовнішніх процедур C/C++ за допомогою наявного в них інтерфейсу для їхньої інтеграції у внутрішню бібліотеку перетворень. Процес перетворення виконується або системою, що інтерпретує специфічні перетворення в процесі роботи, або відкомпільованим кодом. Усі засоби на базі системи (наприклад, COPYMANAGER, DECISIONBASE, POWERMART, DATASTAGE, WAREHOUSEADMINISTRATOR), мають планувальник і підтримують технологічні процеси зі складними залежностями виконання між етапами перетворення. Технологічний процес може також допомагати роботі зовнішніх засобів (скажімо, у специфічних завданнях очищення це будуть очищення імен/адрес або виключення дублікатів).

Засоби ETL звичайно містять мало вбудованих можливостей очищення, але дозволяють користувачеві визначати функціональність очищення через власний API. Як правило, аналіз даних для автоматичного виявлення помилок і невідповідностей у даних не підтримується. Проте користувачі можуть реалізовувати таку логіку при роботі з метаданими й шляхом визначення характеристик умісту за допомогою функцій агрегації (sum, count, min, max, median, variance, deviation).

Мови правил звичайно охоплюють конструкції if-then і case, що сприяють обробці виключень у значеннях даних, – невірних написань, абревіатур, втрачених або зашифрованих значень і значень поза припустимим діапазоном. Ці проблеми можуть також вирішуватися за допомогою функціональних можливостей по вибірці даних із таблиць. Підтримка узгодження елементів даних звичайно обмежена використанням можливостей об'єднання й декількох простих строкових функцій відповідності, наприклад точної або групової відповідності або soundex. Проте визначені користувачем функції відповідності полів, так само як і функції кореляції подібності полів, можуть програмуватися й додаватися у внутрішню бібліотеку перетворень.

Інша класифікація засобів очищення даних, запропонована Джулі Борт, підрозділяє інструменти очищення даних на дві умовні категорії:

- універсальні системи, призначені для обслуговування всієї бази даних цілком;
- верифікатори імені/адреси для очищення тільки даних про клієнтів.

Універсальні системи. До цієї категорії належить більша частина продуктів, наявних на ринку. Це: Enterprise Integrator компанії Apertus; Integrity Data Reengineering Tool проведення Validy Technology; Data Quality Administrator від Gladstone Computer Services; Inforefiner фірми Platinium Technology; QDB Analyze (проведення QDB Solutions) Trillium Software System компанії Hart-Hanks Data Technologies.

Ці системи слід вибирати тоді, коли йдеться про створення банків даних усього підприємства й, відповідно, про суцільне очищення даних. Кожна система використовує власну технологію й має власну сферу додатків. Деякі з них працюють у пакетному режимі, наприклад, Trillium, яка переглядає дані в пошуках певних образів і навчається на основі знайденої інформації. Образи, що підлягають розпізнаванню (скажімо, назви фірм або міські адреси), задаються на етапі попереднього програмування. Інші продукти, як то системи компаній Apertus і Validy, являють собою засоби розробки. У першій застосовуються правила, написані мовою Object Query Language. З нею досить легко працювати, але для написання правил потрібна справжня майстерність.

Система компанії Validy при відборі записів використовує алгоритми нечіткої логіки й робить це дуже ефективно, вивуджуючи таке, що людині просто в голову не прийшло б перевіряти. Але цю систему сутужніше освоїти.

Верифікатори імені/адреси. У простих системах, на зразок систем аналізу ринку, цілком можна обійтися очищенням імен і адрес. Приклади продуктів цієї категорії: Nadis компанії Group 1 Software і пакет компанії Postalsoft. Останній містить три бібліотеки: виправлення й кодування адрес, оформлення правильних імен і злиття/очищення. Перша бібліотека коректує адреси, друга пропонує спосіб їх стандартизації, третя виконує консолідуючі функції.

Ці продукти простіше використовувати, і, оскільки область застосування їх не така широка, роботу з очищення вони виконають значно швидше. Як додаткову функцію це програмне забезпечення надає адресам вид, що відповідає вимогам пошти. Приміром, Nadis автоматично перетворить ім’я й адреса в стандарт Universal Name and Address data standard.

Додатковий продукт компанії Group 1, Code-1 Plus, перевіряє список адрес на відповідність вимогам. Сертифікація гарантує коректність Zip-Коду й використовується при більших обсягах вихідної пошти. Ті, хто застосовував ці засоби, говорять, що автоматизація роботи із забезпечення відповідності адрес різним правилам, установленим поштовим відомством, коштує витрачених зусиль і засобів, навіть якщо доводиться доповнювати названі пакети іншими коштами очищення.

Отже, шляхом використання спеціальних засобів очищення й редагування даних вирішується проблема неякісних або брудних даних. Однак

автоматизований процес очищення даних іноді може призводити до помилок у даних, яких раніше в них не було.

Річ Олшефскі (Rich Olshefski) пропонує класифікацію помилок у даних, які виникають у результаті використання засобів очищення. Ці помилки є двома крайностями очищення даних. Якісні, правильно очищені дані перебувають десь на «золотій середині» між цими крайностями по очищенню й редагуванню даних.

Помилка типу 1 виникає, коли інструмент очищення намагається вирішити проблему, якої насправді не існує, тобто починає виправляти невідповідності в даних там, де їх немає.

Помилка типу 2 виникає, коли інструменти очищення повністю упускають існуючу проблему, тобто трапляється при недогляді програмою невірних даних. Такі дані безперешкодно проходять перевірку, будучи при цьому помилковими. Цю помилку ще називають «втраченою помилкою». Програма очищення даних пропускає дані, які насправді повинна була виправити.

Проблема. Саме складне завдання, що постає перед програмою очищення даних, полягає в мінімізації помилок типу 1 і 2. Для усунення помилок Типу 1 програма повинна намагатися не виправляти те, що й так вірно. Це відразу ж закономірним чином підвищує ймовірність виникнення помилки типу 2. Помилок типу 2 можна уникнути шляхом скрупульозної роботи з даними, що, звичайно ж, негайно приводить до зайвого очищення й, відповідно, – до допущення помилки типу 1.

Деякі програми очищення намагаються так чи інакше підтримувати баланс між зайвою старанністю й зайвою довірою, створюючи великі за обсягом звіти про «підозрілі» записи. Ці програми збирають усе підозріле в одну велику купу, яка і є таким звітом. Така методика суттєво збільшує витрати на уточнення даних, оскільки вимагає участі дорогих людських ресурсів.

Іншим шляхом надмірної компенсації помилок типу 1 є внесення занадто малого числа виправлень. А самі примітивні – і тому найнебезпечніші – програми очищення даних намагаються компенсувати й помилки Типу 2, видаючи на виході щось набагато більш кепське, ніж те, що було до «очищення».

Визначення якісної програми очищення даних, за словами Річа Олшефскі, складається із чотирьох елементів. *Програма повинна:*

- не торкатися правильних даних;
- виправляти невірні;
- створювати невеликий за обсягом звіт про підозрілі записи;
- вимагати мінімальних витрат на установку, обслуговування й ручні перевірки.

Питання для самоконтролю

1. Які етапи включає традиційний процес Data Mining?
2. Які кроки та як відбувається постановка задачі Data Mining? Наведіть приклад.
3. Які цілі підготовки даних?

4. Що можна зробити із пропущеними даними?
5. Для чого відбувається дублювання даних?
6. Дайте визначення поняттю «очищення даних» (data cleaning, data cleansing або scrubbing). Для чого робиться очищення даних?

Тема 9. Методи дерев рішень, класифікації та прогнозування

План

1. Метод дерев рішень.
2. Переваги дерев рішень.
3. Алгоритми.
4. Метод опорних векторів.
5. Лінійний SVM.
6. Метод «найближчого сусіда».
7. Байєсовська класифікація.

Мета вивчення теми: вивчити методи прогнозування та класифікації; засвоїти поняття дерева рішень; вивчити метод опорних векторів; засвоїти метод найближчого сусіда; засвоїти поняття байєсовської класифікації.

Перелік ключових слів та понять із теми

Прогнозування, класифікація, метод дерев рішень, метод опорних векторів, метод найближчого сусіда, байєсовська класифікація, алгоритм CART, крос-перевірка

Теоретичні відомості з теми

1. Метод дерев рішень

Метод дерев рішень (decision trees) є одним із найбільш популярних методів розв'язання задач класифікації й прогнозування. Іноді цей метод Data Mining також називають деревами вирішальних правил, деревами класифікації і регресії.

Як видно з останньої назви, за допомогою даного методу розв'язуються задачі класифікації й прогнозування.

Якщо залежна, тобто цільова змінна приймає дискретні значення, за допомогою методу дерева рішень розв'язується задача класифікації.

Якщо ж залежна змінна приймає безперервні значення, то дерево рішень установлює залежність цієї змінної від незалежних змінних, тобто розв'язує задачу чисельного прогнозування.

У найбільш простому вигляді дерево рішень – це спосіб показу правил в ієрархічній, послідовній структурі. Основа такої структури – відповіді «Так» або «Ні» на низку питань.

Наведемо приклад дерева рішень, задача якого – відповістити на запитання: «Чи грати в гольф?» Щоб розв'язати задачу, тобто прийняти рішення, чи грати в гольф, слід віднести поточну ситуацію до одного з відомих класів (у цьому випадку – «грати» або «не грати»). Для цього потрібно відповісти на низку питань, які є у вузлах цього дерева, починаючи з його кореня.

Перший вузол нашого дерева «Сонячно?» є вузлом перевірки, тобто умовою. При позитивній відповіді на запитання здійснюється перехід до лівої частини дерева, що називається лівою гілкою, при негативному – до правої

частини дерева. Отже, внутрішній вузол дерева є вузлом перевірки певної умови. Далі йде наступне питання і т.д., поки не буде досягнутий кінцевий вузол дерева, що є вузлом розв'язку. Для нашого дерева існує два типи кінцевого вузла: «грати» і «не грати» у гольф.

У результаті проходження від кореня дерева (іноді називається кореневою вершиною) до його вершини розв'язується задача класифікації, тобто вибирається один із класів – «грати» чи «не грати» у гольф**.**

Метою побудови дерева рішень в нашому випадку є визначення значення категоріальної залежної змінної.

Отже, для нашої задачі основними елементами дерева рішень є:

- Корінь дерева: «Сонячно?»
- Внутрішній вузол дерева або вузол перевірки: «Температура повітря висока?», «Чи йде дощ?»
- Листок, кінцевий вузол дерева, вузол розв'язку або вершина: «Грати», «Не грати». Гілки дерева (варіанти відповіді): «Так», «Ні».

У розглянутому прикладі розв'язується задача бінарної класифікації, тобто створюється дихотомічна класифікаційна модель. Приклад демонструє роботу так званих бінарних дерев.

У вузлах бінарних дерев розгалуження може відбуватися тільки у двох напрямках, тобто існує можливість тільки двох відповідей на поставлене питання («так» і «ні»).

Бінарні дерева є найпростішим, частковим випадком дерев рішень. В інших випадках, відповідей і, відповідно, гілок дерева, що виходять із його внутрішнього вузла, може бути більше двох.

Розглянемо більш складний приклад.

База даних, на основі якої повинне здійснюватися прогнозування, містить такі ретроспективні дані про клієнтів банку, що є її атрибутами:

- вік,
- наявність нерухомості,
- освіта,
- середньомісячний дохід,
- чи повернув клієнт вчасно кредит.

Задача полягає в тому, щоб на підставі перерахованих вище даних (крім останнього атрибута) визначити, чи варто видавати кредит новому клієнтові.

Як ми вже розглядали в лекції, присвяченій задачі класифікації, така задача розв'язується у два етапи:

- побудова класифікаційної моделі
- її використання.

На етапі побудови моделі, власне, і будується дерево класифікації або створюється набір якихось правил.

На етапі використання моделі побудоване дерево, або шлях від його кореня до однієї з вершин, що є набором правил для конкретного клієнта, використовується для відповіді на поставлене питання «Чи видавати кредит?»

Правилом є логічна конструкція, представлена у вигляді «якщо : то :».

Наведемо приклад дерева класифікації, за допомогою якого розв'язується задача «Чи видавати кредит клієнтові?». Вона є типовою задачею класифікації, і за допомогою дерев рішень одержують досить хороші варіанти її розв'язку.

Як ми бачимо, внутрішні вузли дерева (вік, наявність нерухомості, дохід і освіта) є атрибутами описаної вище бази даних.

Ці атрибути називають прогнозуючими, або атрибутами розщеплення (splitting attribute). Кінцеві вузли дерева, або листки, іменуються мітками класу, що є значеннями залежної категоріальної змінної «видавати» або «не видавати» кредит.

Кожна гілка дерева, що йде від внутрішнього вузла, відзначена предикатом розщеплення. Останній може відноситися лише до одного атрибуту розщеплення даного вузла.

Характерна риса предикатів розщеплення: кожний запис використовує унікальний шлях від кореня дерева тільки до одного вузла-розв'язку. Об'єднана інформація про атрибути розщеплення й предикати розщеплення у вузлі називається критерієм розщеплення (splitting criterion).

Наприклад, критерій розщеплення «Яка освіта?», міг би мати два предикати розщеплення й виглядати інакше: освіта «вища» і «не вища». Тоді дерево рішень мало б інший вигляд.

Отже, для цієї задачі (як і для будь-якої іншої) може бути побудовано множина дерев рішень різної якості, з різною прогнозуючою точністю.

Якість побудованого дерева рішень досить сильно залежить від правильного вибору критерію розщеплення. Над розробкою й удосконаленням критеріїв працюють багато дослідників.

Метод дерев рішень часто називають «наївним» підходом. Але завдяки певній низці переваг, цей метод є одним із найбільш популярних для розв'язання задач класифікації.

2. Переваги дерев рішень

Розглянемо влесне ці переваги.

Інтуїтивність дерев рішень. Класифікаційна модель, представлена у вигляді дерева рішень, є інтуїтивною і спрощує розуміння розв'язуваної задачі.

Результат роботи алгоритмів конструювання дерев рішень, *на відміну, наприклад, від нейронних мереж, що представляють собою «чорні ящики»*, легко інтерпретується користувачем. Ця властивість дерев рішень не тільки важлива при віднесенні до певного класу нового об'єкта, але й корисна при інтерпретації моделі класифікації в цілому. Дерево рішень дозволяє зрозуміти й пояснити, чому конкретний об'єкт відноситься до того або іншого класу.

Дерева рішень дають можливість витягати правила з бази даних звичайною мовою. Приклад правила: Якщо Вік >35 і Дохід >200, то видати кредит.

Дерева рішень дозволяють створювати класифікаційні моделі в тих сферах, де аналітикові досить складно формалізувати знання.

Алгоритм конструювання дерева рішень не вимагає від користувача вибору вхідних атрибутів (незалежних змінних). На вхід алгоритму можна подавати всі існуючі атрибути, алгоритм сам вибере найбільш значимі серед них,

і тільки вони будуть використані для побудови дерева. У порівнянні, наприклад, з нейронними мережами, це значно полегшує користувачеві роботу, оскільки в нейронних мережах вибір кількості вхідних атрибутів суттєво впливає на час навчання.

Точність моделей, створених за допомогою дерев рішень, порівняно з іншими методами побудови класифікаційних моделей (статистичні методи, нейронні мережі).

Розроблений ряд масштабованих алгоритмів, які можуть бути використані для побудови дерев рішень на надвеликих базах даних. Масштабованість тут означає, що із зростанням кількості прикладів або записів бази даних час, затрачуваний на навчання, тобто побудову дерев рішень, зростає лінійно. Приклади таких алгоритмів: SLIQ, SPRINT.

Швидкий процес навчання. На побудову класифікаційних моделей за допомогою алгоритмів конструювання дерев рішень потрібно значно менше часу, ніж, наприклад, на навчання нейронних мереж.

Більшість алгоритмів конструювання дерев рішень мають можливість спеціальної обробки пропущених значень.

Багато класичних статистичних методів, за допомогою яких розв'язуються задачі класифікації, можуть працювати тільки із числовими даними, у той час як дерева рішень працюють і з числовими, і з категоріальними типами даних.

Багато статистичних методів є параметричними, і користувач повинен заздалегідь володіти певною інформацією, наприклад, знати вид моделі, мати гіпотезу про вид залежності між змінними, припускати, який вид розподілу мають дані. Дерева рішень, на відміну від таких методів, будують непараметричні моделі. Отже, дерева рішень здатні розв'язувати такі задачі Data Mining, у яких відсутня апріорна інформація про вид залежності між досліджуваними даними.

Процес конструювання дерева рішень. Нагадаємо, що розглянута нами задача класифікації відноситься до стратегії навчання з учителем, яке іноді називається індуктивним навчанням. У цих випадках усі об'єкти тренувального набору даних заздалегідь віднесені до одного з визначених класів.

Алгоритми конструювання дерев рішень складається з етапів «побудова» або «створення» дерева (tree building) і «скорочення» дерева (tree pruning). У ході створення дерева вирішуються питання вибору критерію розщеплення й зупинки навчання (якщо це передбачено алгоритмом). У ході етапу скорочення дерева вирішується питання відсікання деяких його гілок.

Розглянемо ці питання детальніше.

Критерій розщеплення. Процес створення дерева відбувається зверху вниз, тобто є спадним. У ході процесу алгоритм повинен знайти такий критерій розщеплення, іноді також називається критерієм розбивки, щоб розбити множину на підмножини, які б асоціювалися з даним вузлом перевірки. Кожний вузол перевірки повинен бути позначений певним атрибутом. Існує правило вибору атрибута: він повинен розбивати вихідну множину даних таким чином, щоб об'єкти підмножин, що одержуються в результаті цієї розбивки, були представниками одного класу або ж були максимально наближені до такої

розбивки. Остання фраза означає, що кількість об'єктів з інших класів, так званих «домішок», у кожному класі прагнула до мінімуму.

Існують різні критерії розщеплення. Найбільш відомі – міра ентропії й індекс Gini.

У деяких методах для вибору атрибута розщеплення використовується так звана міра інформативності підпросторів атрибутів, яка ґрунтується на ентропійному підході й відома за назвою «захід інформаційного виграшу» (information gain measure) або захід ентропії.

Інший критерій розщеплення, запропонований Брейманом (Breiman) та ін., реалізований в алгоритмі CART і називається індексом Gini. За допомогою цього індексу атрибут вибирається на підставі відстаней між розподілами класів.

Якщо дана множина *T*, що включає приклади з *n* класів, індекс Gini, тобто *gini(T)*, визначається за формулою:

$$gini(T) = 1 - \sum_{j=1}^{n} p_j^2, \qquad (9.1)$$

де *T* – поточний вузол, *pj* – імовірність класу *j* у вузлі *T*, *n* – кількість класів.

Велике дерево не означає, що воно «вдале». Чим більше окремих випадків описано в дереві рішень, тим менша кількість об'єктів потрапляє в кожний окремий випадок. Такі дерева називають «гіллястими» або «кущистими», вони складаються з невиправдано великої кількості вузлів і гілок, вихідна множина розбивається на велику кількість підмножин, що складаються із дуже малого числа об'єктів. У результаті «переповнення» таких дерев їх здатність до узагальнення зменшується, і побудовані моделі не можуть давати вірні відповіді.

У процесі побудови дерева, щоб його розміри не стали надмірно великими, використовують спеціальні процедури, які дозволяють створювати оптимальні дерева, так звані дерева «вдалих розмірів» (Breiman, 1984).

Який розмір дерева може вважатися оптимальним? Дерево повинно бути досить складним, щоб ураховувати інформацію з досліджуваного набору даних, але одночасно воно повинно бути досить простим. Інакше кажучи, дерево повинно використовувати інформацію, що поліпшує якість моделі, і ігнорувати ту інформацію, яка її не поліпшує.

Отут існує дві можливі стратегії. Перша полягає в нарощуванні дерева до певного розміру відповідно до параметрів, заданих користувачем.

Визначення цих параметрів може ґрунтуватися на досвіді й інтуїції аналітика, а також на деяких «діагностичних повідомленнях» системи, що конструюють дерево рішень.

Друга стратегія полягає у використанні набору процедур, що визначають «вдалий розмір» дерева, вони розроблені Бріманом, Куїлендом та ін. в 1984 році. Однак, як відзначають автори, не можна сказати, що ці процедури доступні починаючому користувачеві.

Процедури, які використовують для запобігання створення надмірно великих дерев, включають: скорочення дерева шляхом відсікання гілок; використання правил зупинки навчання.

Слід зазначити, що не всі алгоритми при конструюванні дерева працюють за однією схемою. Деякі алгоритми включають два окремі послідовні етапи: побудова дерева і його скорочення; інші чергують ці етапи в процесі своєї роботи для запобігання нарощування внутрішніх вузлів.

Зупинка побудови дерева. Розглянемо правило зупинки. Воно повинне визначити, чи є розглянутий вузол внутрішнім вузлом (при цьому він буде розбиватися далі) або ж він є кінцевим вузлом, тобто вузлом розв'язком.

Зупинка – такий момент у процесі побудови дерева, коли слід припинити подальші розгалуження.

Один із варіантів правил зупинки – «рання зупинка» (prepruning), вона визначає доцільність розбивки вузла. Перевага використання такого варіанта – зменшення часу на навчання моделі. Однак тут виникає ризик зниження точності класифікації. Тому рекомендується «замість зупинки використовувати відсікання» (Breiman, 1984).

Другий варіант зупинки навчання – обмеження глибини дерева. У цьому випадку побудова закінчується, якщо досягнута задана глибина.

Ще один варіант зупинки – задання мінімальної кількості прикладів, які будуть утримуватися в кінцевих вузлах дерева. При цьому варіанті розгалуження тривають до того моменту, поки всі кінцеві вузли дерева не будуть чистими або будуть містити не більш ніж задане число об'єктів.

Існує ще ряд правил, але слід зазначити, що жодне з них не має великої практичної цінності, а деякі можуть бути застосовні лише в окремих випадках.

Скорочення дерева або відсікання гілок. Вирішенням проблеми занадто гіллястого дерева є його скорочення шляхом відсікання (pruning) деяких гілок.

Якість класифікаційної моделі, побудованої за допомогою дерева рішень, характеризується двома основними ознаками: точністю розпізнавання й помилкою.

Точність розпізнавання розраховується як відношення об'єктів, правильно класифікованих у процесі навчання, до загальної кількості об'єктів набору даних, які брали участь у навчанні.

Помилка розраховується як відношення об'єктів, неправильно класифікованих у процесі навчання, до загальної кількості об'єктів набору даних, які брали участь у навчанні.

Відсікання гілок або заміну деяких гілок піддеревом слід проводити там, де ця процедура не призводить до зростання помилки. Процес проходить знизу вгору, тобто є висхідним. Це більш популярна процедура, ніж використання правил зупинки. Дерева, одержувані після відсікання деяких гілок, називають усіченими.

Якщо таке усічене дерево усе ще не є інтуїтивним і складне для розуміння, використовують витяг правил, які поєднують у набори для опису класів.

Кожний шлях від кореня дерева до його вершини або листка дає одне правило. Умовами правила є перевірки на внутрішніх вузлах дерева.

3. Алгоритми

Сьогодні існує велика кількість алгоритмів, що реалізують дерева рішень: CART, C4.5, CHAID, CN2, Newid, Itrule і інші. Атрибути набору даних можуть мати як дискретне, так і числове значення.

Алгоритм CART (Classification and Regression Tree), як видно з назви, розв'язує задачу класифікації й регресії. Він розроблений в 1974-1984 роках чотирма професорами статистики – Leo Breiman (Berkeley), Jerry Friedman (Stanford), Charles Stone (Berkeley) і Richard Olshen (Stanford).

Алгоритм CART призначений для побудови бінарного дерева рішень. Бінарні дерева також називають двійковими. Приклад такого дерева розглядався на початку лекції.

Інші особливості алгоритму CART:

- функція оцінки якості розбивки;
- механізм відсікання дерева;
- алгоритм обробки пропущених значень;
- побудова дерев регресії.

Кожний вузол бінарного дерева при розбивці має тільки двох нащадків, що називаються дочірніми галузями. Подальший поділ гілок залежить від того, чи багато вихідних даних описує дана гілка. На кожному кроці побудови дерева правило, формоване у вузлі, ділить задану множину прикладів на дві частини. Права його частина (гілка right) – це та частина множини, у якій правило виконується; ліва (гілка left) – та, для якої правило не виконується.

Функція оцінки якості розбивки, яка використовується для вибору оптимального правила, – індекс Gini – був описаний вище. Відзначимо, що дана оціночна функція заснована на ідеї зменшення невизначеності у вузлі. Допустимо, є вузол, і він розбитий на два класи. Максимальна невизначеність у вузлі буде досягнута при розбивці його на дві підмножини по 50 прикладів, а максимальна визначеність – при розбивці на 100 і 0 прикладів.

Правила розбивки. Нагадаємо, що алгоритм CART працює із числовими й категоріальними атрибутами. У кожному вузлі розбивка може йти тільки по одному атрибуту. Якщо атрибут є числовим, то у внутрішньому вузлі формується правило виду $x_i \le c$, Значення c у більшості випадків вибирається як середнє арифметичне двох сусідніх впорядкованих значень змінної x_i навчального набору даних. Якщо ж атрибут відноситься до категоріального типу, то у внутрішньому вузлі формується правило $x_i\ V(x_i)$, де $V(x_i)$ – деяка непорожня підмножина множин значень змінної x_i у навчальному наборі даних.

Механізм відсікання. Цим механізмом, що має назву minimal cost-complexity tree pruning, алгоритм CART принципово відрізняється від інших алгоритмів конструювання дерев рішень. У розглянутому алгоритмі відсікання – це деякий компроміс між одержанням дерева «підходящого розміру» і одержанням найбільш точної оцінки класифікації. Метод полягає в одержанні послідовності зменшуваних дерев, але дерева розглядаються не всі, а тільки «кращі представники».

Перехресна перевірка (V-fold cross-validation) є найбільш складною й одночасно оригінальною частиною алгоритму CART. Вона являє собою шлях вибору остаточного дерева, за умови, що набір даних має невеликий обсяг або ж

записи набору даних настільки специфічні, що розділити набір на навчальну й тестову вибірку не представляється можливим.

Отже, основні характеристики алгоритму CART: бінарне розщеплення, критерій розщеплення – індекс Gini, алгоритми minimal cost-complexity tree pruning і V-fold cross-validation, принцип «виростити дерево, а потім скоротити», висока швидкість побудови, обробка пропущених значень.

Алгоритм C4.5 будує дерево рішень з необмеженою кількістю гілок у вузлах. Даний алгоритм може працювати тільки з дискретним залежним атрибутом і тому може розв'язувати тільки задачу класифікації. C4.5 вважається одним із найвідоміших і широко використовуваних алгоритмів побудови дерев класифікації.

Для роботи алгоритму C4.5 необхідне дотримання таких вимог:

- Кожний запис набору даних повинен бути асоційованим з одним із визначених класів, тобто один з атрибутів набору даних повинен бути міткою класу.
- Класи повинні бути дискретними. Кожний приклад повинен однозначно відноситися до одного із класів.
- Кількість класів повинна бути значно менше кількості записів у досліджуваному наборі даних.

Остання версія алгоритму – алгоритм C4.8 – реалізована в інструменті Weka як J4.8 (Java). Комерційна реалізація методу: C5.0, розроблювач Rulequest, Австралія.

Алгоритм C4.5 повільно працює на надвеликих й зашумлених наборах даних.

Ми розглянули два відомі алгоритми побудови дерев рішень CART і C4.5. Обидва алгоритми є робастними, тобто стійкими до шумів і викидів даних.

Алгоритми побудови дерев рішень відрізняються такими характеристиками:

- вид розщеплення – бінарне (binary), множинне (multi-way);
- критерії розщеплення – ентропія, gini, інші;
- можливість обробки пропущених значень;
- процедура скорочення гілок або відсікання.;
- можливості витягування правил з дерев.

Жоден алгоритм побудови дерева не можна апріорі вважати найкращим або досконалим, підтвердження доцільності використання конкретного алгоритму повинно бути перевірене й підтверджене експериментом.

Розробка нових масштабованих алгоритмів. Найбільш серйозна вимога, яка зараз пред'являється до алгоритмів конструювання дерев рішень – це масштабованість, тобто алгоритм повинен мати масштабований метод доступу до даних.

Розроблений ряд нових масштабованих алгоритмів, серед них – алгоритм Sprint, запропонований Джоном Боярином і його колегами. Sprint, що є масштабованим варіантом розглянутого в лекції алгоритму CART, висуває мінімальні вимоги до об'єму оперативної пам'яті.

4. Метод опорних векторів

Метод опорних векторів (Support Vector Machine – SVM) відноситься до групи граничних методів. Він визначає класи за допомогою границь областей.

За допомогою даного методу розв'язуються задачі бінарної класифікації. В основі методу лежить поняття площин розв'язків. Площина (plane) розв'язку розділяє об'єкти з різною класовою приналежністю.

На рис. 9.1 наведений приклад, у якому беруть участь об'єкти двох типів. Поділяюча лінія задає границю, праворуч від якої – усі об'єкти типу brown (коричневий), а ліворуч – типу yellow (жовтий). Новий об'єкт, що потрапляє праворуч, класифікується як об'єкт класу brown або – як об'єкт класу yellow, якщо він розташувався ліворуч від поділяючої прямої. У цьому випадку кожний об'єкт характеризується двома вимірами.

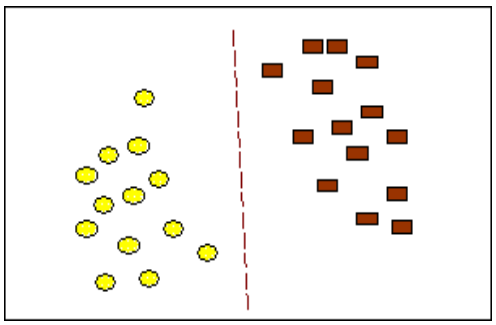


Рисунок 9.1 – Поділ класів прямою лінією

Ціль методу опорних векторів – знайти площину, що розділяє дві множини об'єктів; така площина показана на рис. 9.2. На цьому рисунку множина зразків поділена на два класи: жовті об'єкти належать класу A, коричневі – класу B.

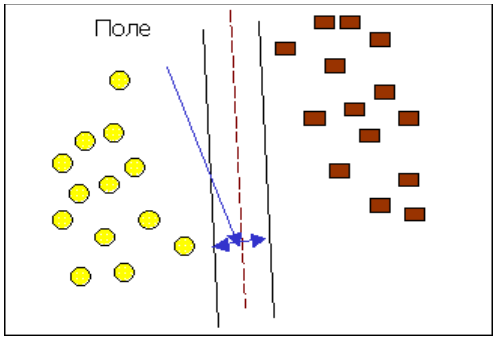


Рисунок 9.2 – До визначення опорних векторів

Метод відшукує зразки, що перебувають на границях між двома класами, тобто опорні вектори. П'ять векторів, які є опорними для даної множини, зображені на 9.3.

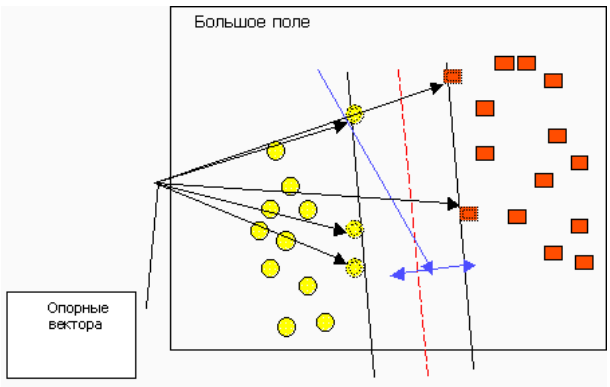


Рисунок 9.3 – Опорні вектори

Опорними векторами називаються об’єкти множини, що знаходяться на границях областей. Класифікація вважається гарною, якщо область між границями порожня.

5. Лінійний SVM

Розв’язання задачі бінарної класифікації за допомогою методу опорних векторів полягає в пошуку деякої лінійної функції, яка правильно розділяє набір даних на два класи. Розглянемо задачу класифікації, де число класів рівне двом.

Задачу можна сформулювати як пошук функції *f(x)*, що приймає значення менше нуля для векторів одного класу й більше нуля – для векторів іншого класу. Як вихідні дані для розв’язання поставленої задачі, тобто пошуку функції *f(x)*, що класифікує, надано тренувальний набір векторів простору, для яких відома їхня приналежність до одного із класів. Сімейство функцій, що класифікують, можна описати через функцію *f(x)*. Гіперплощина визначена вектором *a* і значенням *b*, тобто *f(x)=ax+b*. Розв’язок даної задачі проілюстрований на рис. 9.4.

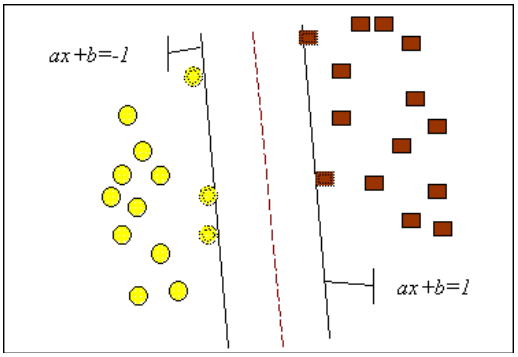


Рисунок 9.4 – Лінійний SVM

У результаті розв’язання задачі, тобто побудови SVM-моделі, знайдена функція, що приймає значення менше нуля для векторів одного класу й більше нуля – для векторів іншого класу. Для кожного нового об’єкта негативне або позитивне значення визначає приналежність об’єкта до одного із класів.

Найкращою функцією класифікації є функція, для якої очікуваний ризик мінімальний. Поняття очікуваного ризику в цьому випадку означає очікуваний рівень помилки класифікації.

Прямо оцінити очікуваний рівень помилки побудованої моделі неможливо, це можна зробити за допомогою поняття емпіричного ризику. Однак слід прийняти, що мінімізація останнього не завжди приводить до мінімізації очікуваного ризику. Це слід пам'ятати при роботі з відносно невеликими наборами тренувальних даних.

Емпіричний ризик – рівень помилки класифікації на тренувальному наборі.

Отже, у результаті розв'язання задачі методом опорних векторів для лінійно поділюваних даних ми одержуємо функцію класифікації, яка мінімізує верхню оцінку очікуваного ризику.

Однією з проблем, пов'язаних із розв'язком задач класифікації розглянутим методом, є та обставина, що не завжди можна легко знайти лінійну границю між двома класами.

У таких випадках один із варіантів – збільшення розмірності, тобто перенесення даних із площини в тривимірний простір, де можливо побудувати таку площину, яка ідеально розділить множину зразків на два класи. Опорними векторами в цьому випадку будуть служити об'єкти з обох класів, що є екстремальними.

Отже, за допомогою додавання так званого оператора ядра й додаткових розмірностей, знаходяться границі між класами у вигляді гіперплощин.

Однак слід пам'ятати: складність побудови SVM-моделі полягає в тому, що чим вища розмірність простору, тим складніше з ним працювати. Один із варіантів роботи з даними високої розмірності – це попереднє застосування якого-небудь методу зниження розмірності даних для виявлення найбільш істотних компонентів, а потім використання методу опорних векторів.

Як і будь-який інший метод, метод SVM має свої сильні й слабкі сторони, які слід враховувати при виборі цього методу.

Недолік методу полягає в тому, що для класифікації використовується не вся множина зразків, а лише їхня невелика частина, яка перебуває на границях.

Перевага методу полягає в тому, що для класифікації методом опорних векторів, на відміну від більшості інших методів, достатньо невеликого набору даних. При правильній роботі моделі, побудованої на тестовій множині, цілком можливе застосування даного методу на реальних даних.

Метод опорних векторів дозволяє:

- одержати функцію класифікації з мінімальною верхньою оцінкою очікуваного ризику (рівня помилки класифікації);
- використовувати лінійний класифікатор для роботи з нелінійно поділюваними даними, поєднуючи простоту з ефективністю.

6. Метод «найближчого сусіда»

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

Слід відразу зазначити, що метод «найближчого сусіда» («nearest neighbour») відноситься до класу методів, робота яких ґрунтується на зберіганні даних у пам'яті для порівняння з новими елементами. Із появою нового запису для прогнозування мають місце відхилення між цим записом і подібними наборами даних, та ідентифікується найбільш подібний (або близький сусід).

Наприклад, при розгляді нового клієнта банку, його атрибути порівнюються з усіма існуючими клієнтами даного банку (дохід, вік і т.д.). Множина «найближчих сусідів» потенційного клієнта банку вибирається на підставі найближчого значення доходу, віку і т.д.

При такому підході використовується термін «*k-найближчий сусід*» («*k-nearest neighbour*»). Термін означає, що вибирається *k* «верхніх» (найближчих) сусідів для їхнього розгляду як множини «найближчих сусідів». Оскільки не завжди зручно зберігати всі дані, іноді зберігається тільки множина «типових» випадків. У такому випадку використовуваний метод називають міркуванням за аналогією (Case Based Reasoning, CBR), міркуванням на основі аналогічних випадків, міркуванням по прецедентах.

Прецедент – це опис ситуації в комбінації з докладною вказівкою дій, що застосовують у даній ситуації.

Підхід, заснований на прецедентах, умовно можна поділити на такі етапи:

- збір докладної інформації про поставлене завдання;
- 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 обраного розв'язку до поточної проблеми, якщо це необхідно;
- перевірка коректності кожного нового отриманого розв'язку;
- занесення детальної інформації про новий прецедент у базу прецедентів.

Отже, висновок, заснований на прецедентах, являє собою такий метод аналізу даних, який робить висновок щодо даної ситуації за результатами пошуку аналогій, що зберігаються в базі прецедентів.

Даний метод за своєю суттю належить до категорії «навчання без вчителя», тобто являється технологією «що навчається самостійно», завдяки чому робочі характеристики кожної бази прецедентів із плином часу і накопиченням прикладів покращуються. Розробка баз прецедентів за конкретною предметною областю відбувається звичайною для людини мовою, отже, може бути виконана найбільш досвідченими співробітниками компанії – експертами або аналітиками, що працюють у цій предметній області.

Однак це не означає, що *CBR*-системи самостійно можуть ухвалювати рішення. Останнє завжди залишається за людиною, цей метод лише пропонує можливі варіанти розв'язку й указує на «найрозумніший» з її точки зору.

Переваги методу:

- Простота використання отриманих результатів.
- Розв'язки не унікальні для конкретної ситуації, можливе їх використання для інших випадків.
- Метою пошуку є не гарантовано вірний розв'язок, а кращий з можливих.

Недоліки методу «найближчого сусіда»:

- Цей метод не створює яких-небудь моделей або правил, що узагальнюють попередній досвід, – у виборі розв'язку вони ґрунтуються на всьому масиві доступних історичних даних, тому неможливо сказати, на якій підставі будуються відповіді.
- Існує складність вибору заходу «близькості» (метрики). Від цього заходу головним чином залежить обсяг множини записів, які потрібно зберігати в пам'яті для досягнення задовільної класифікації або прогнозу. Також існує висока залежність результатів класифікації від обраної метрики.
- При використанні методу виникає необхідність повного перебору навчальної вибірки при розпізнаванні, як наслідок цього – обчислювальна трудомісткість.
- Типові завдання цього методу – це завдання невеликої розмірності за кількістю класів і змінних.

За допомогою даного методу розв'язуються задачі класифікації й регресії.

Розглянемо докладно принципи роботи методу *k*-найближчих сусідів для розв'язання задач класифікації й регресії (прогнозування).

Розв'язання задачі класифікації нових об'єктів. Ця задача схематично зображена на рис. 9.5. Приклади (відомі екземпляри) відзначені «+» або «-», що визначають приналежність до відповідного класу, а новий об'єкт, який потрібно класифікувати, позначений кружечком. Нові об'єкти також називають точками запиту.

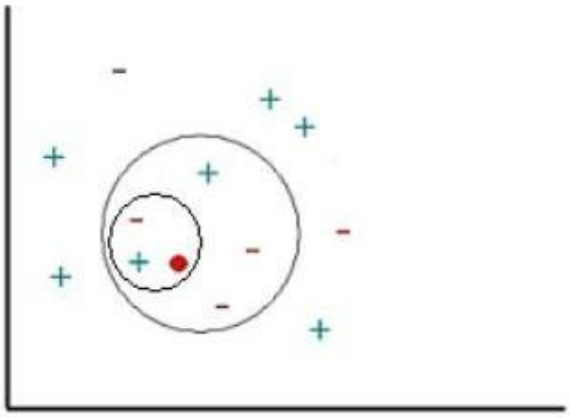


Рисунок 9.5 – Класифікація об'єктів множини при різному значенні параметра *k*

Наша мета полягає в оцінці (класифікації) відгуку точок запиту з використанням спеціально обраного числа їх найближчих сусідів. Інакше кажучи, ми прагнемо довідатися, до якого класу слід віднести точку запиту: знак «+» або знак «-».

Для початку розглянемо результат роботи методу *k*-найближчих сусідів із використанням одного найближчого сусіда. У цьому випадку відгук точки запиту буде класифікований як знак «плюс», тому що 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 обидва 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 графіку її границя відзначена червоним (сірим) колом). Оскільки в околиці утримується 2 точки зі знаком «+» і 3 точки зі знаком «-», алгоритм *k*-найближчих сусідів привласнить знак «-» відгуку точки запиту.

Розв'язання задачі прогнозування. Далі розглянемо принцип роботи методу *k*-найближчих сусідів для розв'язання задачі регресії. Регресійні задачі пов'язані з прогнозуванням значення залежної змінної за значеннями незалежних змінних набору даних.

Розглянемо графік, показаний на рис. 9.6. Зображений на ній набір точок (*зелені* прямокутники) отриманий за зв'язком між незалежною змінною *x* і залежною змінною *y* (*крива червоного кольору*). Заданий набір зелених об'єктів (тобто набір прикладів); застосуємо метод *k*-найближчих сусідів для прогнозування виходу точки запиту *X* за цим набором прикладів (зелені прямокутники).

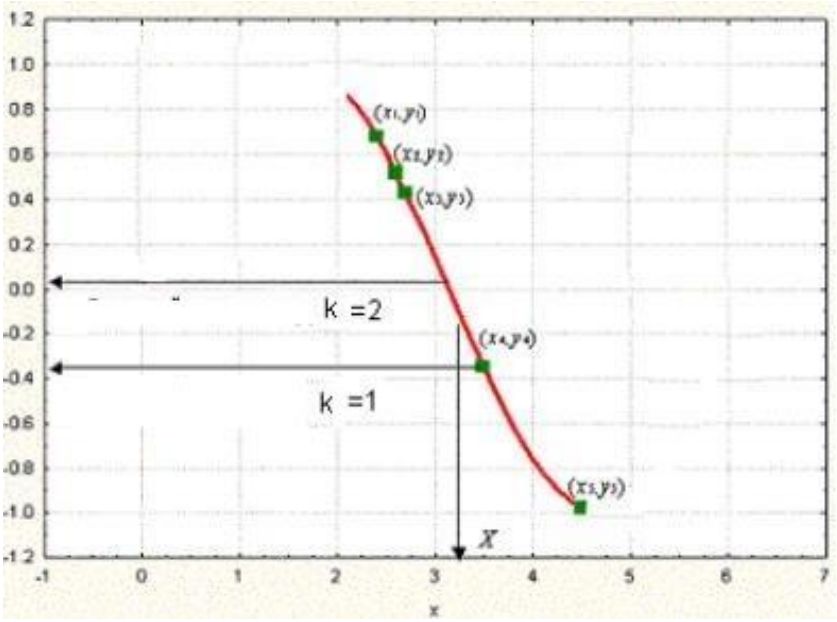


Рисунок 9.6 – Розв'язок задачі прогнозування при різних значеннях параметра *k*

Спочатку розглянемо як приклад метод *k*-найближчих сусідів з використанням одного найближчого сусіда, тобто при *k*, рівному одиниці. Шукаємо набір прикладів (зелені прямокутники) і виділяємо з їхнього числа найближчий до точки запиту *X*. Для цього випадку найближчий приклад – точка (x_4; y_4). Вихід x_4, таким чином, приймається як результат прогнозування виходу *X*. Отже, для одного найближчого сусіда можемо записати: вихід *Y* рівний y_4 ($Y = y_4$).

Далі розглянемо ситуацію, коли *k* рівне двом, тобто розглянемо два найближчі сусіди. У цьому випадку виділяємо вже дві найближчі до *X* точки. На

цьому графіку це точки y_3 і y_4 відповідно. Обчисливши середнє їхніх виходів, записуємо розв'язок для Y у вигляді $Y = (y_3 + y_4)/2$.

Розв'язання 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 й залежні змінні набору даних можуть бути як безперервними, так і категоріальними. Для безперервних залежних змінних задача розглядається як задача прогнозування, для дискретних змінних – як задача класифікації.

Прогнозування в задачі прогнозування виходить усередненням виходів k-найближчих сусідів, а розв'язок задачі класифікації заснований на принципі «за більшістю голосів».

Критичним моментом у використанні методу k-найближчих сусідів є вибір параметра k. Він один із найбільш важливих факторів, що визначають якість прогнозної або класифікаційної моделі.

Якщо обрано занадто мале значення параметра k, виникає ймовірність великого розкиду значень прогнозу. Якщо обране значення занадто велике, це може призвести до суттєвого зміщення моделі. Отже, повинно бути обране оптимальне значення параметра k. Тобто це значення повинно бути настільки великим, щоб звести до мінімуму ймовірність неправильної класифікації, і одночасно, досить малим, щоб k сусідів були розташовані досить близько до точки запиту.

Отже, розглядаємо k параметр, як згладжуючий, для якого потрібно знайти компроміс між силою розмаху (розкиду) моделі та її зміщеностю.

Один із варіантів оцінки параметра k – проведення крос-перевірки (Bishop, 1995). Така процедура реалізована, наприклад, у пакеті Statistica (Statsoft).

Крос-перевірка – відомий метод одержання оцінок невідомих параметрів моделі. Основна ідея методу – поділ вибірки даних на v «складок». V «складки» тут є випадковим чином виділені ізольовані підвибірки.

За фіксованим значенням k будується модель k-найближчих сусідів для одержання прогнозів на v-му сегменті (інші сегменти при цьому використовуються як приклади) і оцінюється помилка класифікації. Для регресійних задач найбільш часто за оцінку 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. По вичерпанню v «складок» (циклів), обчислені помилки усереднюються й використовуються як міра стабільності моделі (тобто міра якості прогнозування в точках запиту). Вищеописані дії повторюються для різних k, і значення, що відповідає найменшій помилці (або найбільшій класифікаційній точності), приймається як оптимальне (оптимальне в сенсі методу крос-перевірки).

Слід враховувати, що крос-перевірка – ємнісна з точки зору обчислень процедура, і необхідно надати час для роботи алгоритму, особливо якщо обсяг вибірки досить великий.

Другий варіант вибору значення параметра k – самостійно задати його значення. Однак цей спосіб слід використовувати, якщо є обґрунтовані припущення щодо можливого значення параметра, наприклад, про попередні дослідження подібних наборів даних.

Метод *k*-найближчих сусідів показує досить непогані результати в найрізноманітніших задачах.

Прикладом реального використання описаного вище методу є програмне забезпечення центру технічної підтримки компанії Dell, розроблене компанією Inference. Ця система допомагає співробітникам центру відповідати на велике число запитів, відразу пропонуючи відповіді на розповсюджені питання й 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 реалізують метод *k*-найближчих сусідів і *CBR*-метод, не дуже 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.

7. Байєсовська класифікація

Теоре́ма Ба́йєса – одна з основних теорем елементарної теорії ймовірностей, яка дозволяє визначити ймовірність якої-небудь події за умови, що сталася інша статистично взаємозалежна з нею подія. Іншими словами, за формулою Байєса можна більш точно перерахувати ймовірність, взявши до уваги раніше відому інформацію і дані нових спостережень. Формула Байєса може бути виведена з основних аксіом теорії ймовірностей, зокрема з умовної ймовірності. Особливість теореми Байєса полягає в тому, що для її практичного застосування потрібна велика кількість розрахунків, обчислень, тому байєсовські оцінки стали активно використовувати тільки після революції в комп'ютерних та мережевих технологіях. Теорема використовується не тільки в аналізі ймовірностей, а й активно застосовується для безлічі інших розрахунків. Психологічні експерименти показали, що люди часто невірно оцінюють ймовірність події, на основі отриманого досвіду (апостеріорна ймовірність), оскільки ігнорують саму ймовірність припущення (апріорна ймовірність). Тому правильний результат за формулою Байєса може суттєво відрізнятися від інтуїтивно очікуваного.

Формула Байєса:

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}, \tag{9.2}$$

де $P(A)$ – апріорна ймовірність гіпотези A;

$P(A|B)$ – ймовірність гіпотези A при настанні події B (апостеріорна ймовірність);

$P(B|A)$ – ймовірність настання події B при істинності гіпотези A;

$P(B)$ – ймовірність настання події B.

Формула виводиться із визначення умовної ймовірності:

$$P(A|B) = \frac{P(A \cup B)}{P(B)} \rightarrow P(A \cup B) = P(A|B)P(B) = P(B|A)P(A).$$

Байєсовський класифікатор – широкий клас алгоритмів класифікації, заснований на принципі максимуму апостеріорної ймовірності. Для класифікованого об'єкта обчислюються функції правдоподібності кожного з класів, за ними обчислюються апостеріорні ймовірності класів. Об'єкт відноситься до того класу, для якого апостеріорна ймовірність максимальна.

Апостеріорна ймовірність – умовна ймовірність випадкової події за умови того, що відомі апостеріорні дані, тобто отримані після досліду.

Альтернативні назви: байєсовське моделювання, байєсовська статистика, метод байєсовських мереж.

Споконвічно байєсовська класифікація використовувалася для формалізації знань експертів в експертних системах, зараз байєсовська класифікація також застосовується як один із методів Data Mining.

Так звана наївна класифікація або наївно-байєсовський підхід (naive-bayes approach) є найбільш простим варіантом методу, що використовує байєсовські мережі. При цьому підході вирішуються задачі класифікації, результатом роботи методу є так звані «прозорі» моделі.

«Наївна» класифікація – досить прозорий і зрозумілий метод класифікації. «Наївною» вона називається тому, що виходить із припущення про взаємну незалежність ознак.

Властивості наївної класифікації:

1. Використання всіх змінних і визначення всіх залежностей між ними.
2. Наявність двох припущень щодо змінних:

– усі змінні є однаково важливими;

– усі змінні є статистично незалежними, тобто значення однієї змінної нічого не говорить про значення іншої.

Більшість інших методів класифікації припускають, що перед початком класифікації ймовірність того, що об'єкт належить тому або іншому класу, однакова; але це не завжди правильно.

Допустимо, відомо, що певний відсоток даних належить конкретному класу. Виникає питання, чи можна використати цю інформацію при побудові моделі класифікації? Існує множина реальних прикладів використання цих апріорних знань, що допомагають класифікувати об'єкти. Типовий приклад із медичної практики. Якщо лікар відправляє результати аналізів пацієнта на додаткове дослідження, він відносить пацієнта до якогось певного класу. Яким чином можна застосувати цю інформацію? Можна використати її як додаткові дані при побудові класифікаційної моделі.

Відзначають такі переваги байєсовських мереж як методу Data Mining:

- у моделі визначаються залежності між усіма змінними, це дозволяє легко обробляти ситуації, в яких значення деяких змінних невідомі;
- байєсовські мережі досить просто інтерпретуються й дозволяють на етапі прогностичного моделювання легко проводити аналіз за сценарієм «що, якщо»;
- байєсовський метод дозволяє природно поєднувати закономірності, виведені з даних, і, наприклад, експертні знання, отримані в явному вигляді;
- використання байєсовських мереж дозволяє уникнути проблеми переучування (overfitting), тобто надлишкового ускладнення моделі, що є слабкою стороною багатьох методів (наприклад, дерев рішень і нейронних мереж).

Наївно-байєсовський підхід має такі недоліки:

- перемножувати умовні ймовірності коректно тільки тоді, коли всі вхідні змінні дійсно статистично незалежні; хоча часто даний метод показує досить гарні результати при недотриманні умови статистичної незалежності, але теоретично така ситуація повинна оброблятися більш складними методами, заснованими на навчанні байєсовських мереж;
- неможлива безпосередня обробка безперервних змінних – потрібно їхнє перетворення до інтервальної шкали, щоб атрибути були дискретними; однак такі перетворення іноді можуть приводити до втрати значимих закономірностей.

Питання для самоконтролю

1. Дайте визначення методу дерев рішень (decision trees)? Для чого він використовується?
2. Як використовуються бінарні дерева? Наведіть приклад.
3. Які існують переваги методу дерева рішень?
4. Які існують критерії розщеплення дерева рішень?
5. Які існують варіанти зупинки навчання дерева рішень?
6. Назвіть відомі алгоритми, що реалізують дерева рішень.

Тема 10. Методи кластерного аналізу. Ієрархічні методи

План

1. Кластерний аналіз.
2. Методи кластерного аналізу.
3. Ієрархічний кластерний аналіз.

Мета вивчення теми: вивчити методи кластерного аналізу; засвоїти особливості ієрархічних методів.

Перелік ключових слів та понять із теми

Кластерний аналіз, кластер, ієрархічні методи, дивизимні методи, евклідова відстань, дентдрограма

Теоретичні відомості з теми

1. Кластерний аналіз

Поняття кластеризації розглянуто в п’ятій темі курсу. У цій лекції опишемо поняття «кластер» із математичної точки зору, а також розглянемо методи розв’язання задач кластеризації – методи кластерного аналізу.

Термін кластерний аналіз, уперше введений Тріоном (Tryon) у 1939 році, містить більш 100 різних алгоритмів.

На відміну від задач класифікації, кластерний аналіз не вимагає апріорних припущень про набір даних, не накладає обмеження на показ досліджуваних об’єктів, дозволяє аналізувати показники різних типів даних (інтервальні дані, частоти, бінарні дані). При цьому необхідно пам’ятати, що змінні повинні вимірюватися в порівнюваних шкалах.

Кластерний аналіз дозволяє скорочувати розмірність даних, робити їх наглядними.

Кластерний аналіз може застосовуватися до сукупностей тимчасових рядів, тут можуть виділятися періоди схожості деяких показників і визначатися групи тимчасових рядів зі схожою динамікою.

Кластерний аналіз паралельно розбудовувався в декількох напрямках, таких як біологія, психологія, ін., тому більшість методів мають по дві й більш назв. Це суттєво ускладнює роботу при використанні кластерного аналізу.

Задачі кластерного аналізу можна об’єднати в такі групи:

1. Розробка типології або класифікації.
2. Дослідження корисних концептуальних схем групування об’єктів.
3. Представлення гіпотез на основі дослідження даних.
4. 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 розв’язуються декілька із зазначених задач.

Розглянемо приклад процедури кластерного аналізу.

Допустимо, маємо набір даних A, що складається з 14-ти прикладів, у яких є по дві ознаки *X* і *Y*. Дані в табличній формі не носять інформативний характер. Представимо змінні *X* і *Y* у вигляді діаграми розсіювання (рис. 10.1).

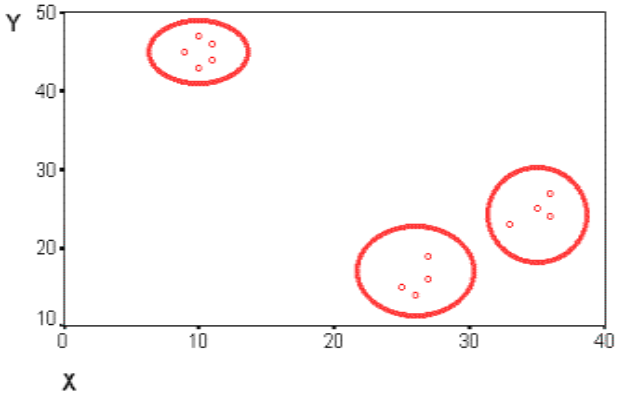


Рисунок 10.1 – Діаграма розсіювання змінних X і Y

На рисунку бачимо кілька груп «схожих» прикладів. Приклади (об'єкти), які за значеннями *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 й відмінності кластерів є відстань між точками на діаграмі розсіювання. Цю подібність можна «виміряти», вона дорівнює відстані між точками на графіку. Способів визначення міри відстані між кластерами, яку називають ще мірою близькості, існує небагато. Найпоширеніший спосіб – обчислення евклідової відстані між двома точками *i* та *j* на площині, коли відомі їхні координати *X* і *Y*:

$$D_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}, \tag{10.1}$$

Якщо нам потрібно знайти відстань між двома точками в просторі трьох вимірів (рис. 10.2), формула (10.1) набуває вигляду:

$$D = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2 + (z_1 - z_2)^2}, \tag{10.2}$$

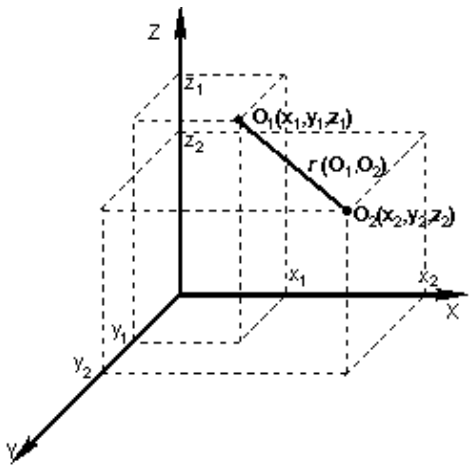


Рисунок 10.2 – Відстань між двома точками в просторі трьох вимірів

Кластер має такі математичні характеристики: центр, радіус, середньоквадратичне відхилення, розмір кластера.

Центр кластера – це середнє геометричне місце точок у просторі змінних.

Радіус кластера – максимальна відстань точок від центру кластера.

Як було відзначено в одній із попередніх тем, кластери можуть бути, такими, що перекриваються. Така ситуація виникає, коли виявляється перекриття кластерів. У 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 у міру подібності може бути віднесений до декільком кластерам.

Розмір кластера може бути визначений або за радіусом кластера, або за середньоквадратичним відхиленням об'єктів для цього кластера. Об'єкт належить до кластера, якщо відстань від об'єкта до центру кластера менше радіуса кластера. Якщо ця умова виконується для двох і більш кластерів, об'єкт є спірним. Неоднозначність може бути усунута експертом або аналітиком.

Робота кластерного аналізу опирається на два припущення. Перше припущення – розглянуті ознаки об'єкта в принципі допускають бажане розбиття сукупності об'єктів на кластери. Друге припущення – правильність вибору масштабу або одиниці вимірювання ознак.

Вибір масштабу в кластерному аналізі має велике значення. Розглянемо приклад. Уявимо собі, що дані ознаки x у наборі даних А на два порядки більші даних ознаки y: значення змінної x перебувають в діапазоні від 100 до 700, а значення змінної y – у діапазоні від 0 до 1.

Тоді, при розрахунках величини відстані між точками, що відображають положення об'єктів у просторі їх властивостей, змінна, що має більші значення, тобто змінна x, буде практично повністю домінувати над змінною з малими значеннями, тобто змінної y. У такий спосіб через неоднорідність одиниць виміру ознак стає неможливим 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 якоїсь величини, що відображає певні властивості конкретної ознаки. Існують різні способи нормування вихідних даних.

Два найпоширеніші способи:

- розподіл вихідних даних на середньоквадратичне відхилення відповідних змінних;
- обчислення Z-внеску або стандартизованого внеску.

Поряд зі стандартизацією змінних, існує варіант додавання до кожної з них певного коефіцієнта важливості, або ваги, яка би відображала значимість відповідної змінної. За ваги можуть виступати експертні оцінки, отримані в ході опитування експертів – фахівців предметної області. Отримані добутки нормованих змінних на відповідні ваги дозволяють одержувати відстані між точками в багатомірному просторі з урахуванням неоднакової ваги змінних.

У ході експериментів можливе порівняння результатів, отриманих з урахуванням експертних оцінок і без них, і вибір якіснішого з них.

2. Методи кластерного аналізу

Методи кластерного аналізу можна розділити на дві групи:

- ієрархічні;
- неієрархічні.

Кожна із груп включає безліч підходів і алгоритмів.Використовуючи різні методи кластерного аналізу, аналітик може одержати різні розв'язки для тих самих даних. Це вважається нормальним явищем.

Розглянемо ієрархічні й неієрархічні методи докладно.

Ієрархічні методи кластерного аналізу. Суть ієрархічної кластеризації полягає в послідовному об'єднанні менших кластерів у більші або поділі більших кластерів на менші.

Ієрархічні агломеративні методи (Agglomerative Nesting, AGNES). Ця група методів характеризується послідовним об'єднанням вихідних елементів і відповідним зменшенням числа кластерів.

На початку роботи алгоритму всі об'єкти є окремими кластерами. На першому кроці найбільш схожі об'єкти поєднуються в кластер. На наступних кроках об'єднання триває доти, поки всі об'єкти не будуть становити один кластер.

Ієрархічні дивизимні (ділені) методи (Divisive Analysis, DIANA). Ці методи є логічною протилежністю агломеративним методам. На початку роботи алгоритму всі об'єкти належать одному кластеру, який на наступних кроках ділиться на менші кластери, у результаті утворюється послідовність груп, що розщеплюються. Принцип роботи описаних вище груп методів у вигляді дендрограми показаний на рис. 10.3.

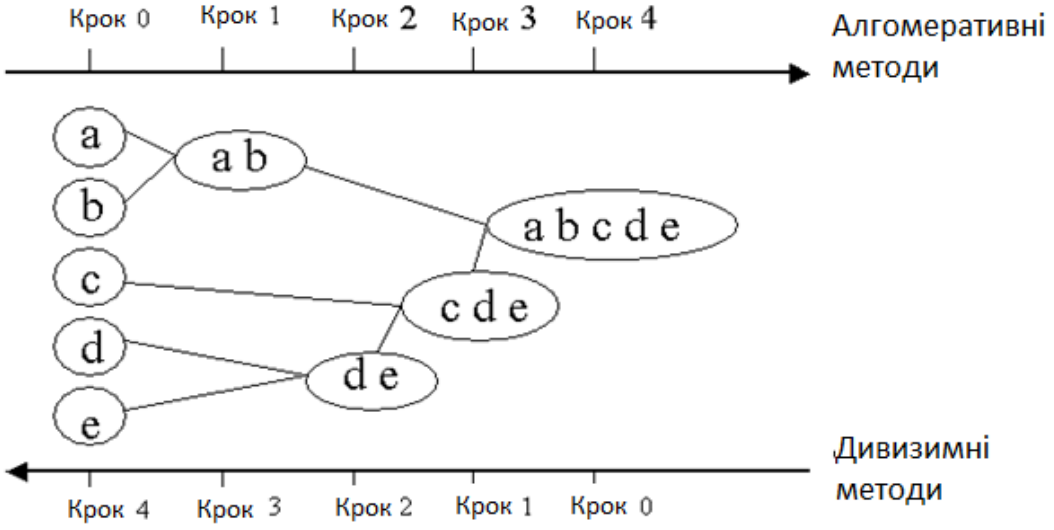


Рисунок 10.3 – Дендрограма агломеративних і дивизимних методів

Програмна реалізація алгоритмів кластерного аналізу широко представлена в різних інструментах Data Mining, які дозволяють вирішувати

завдання досить великої розмірності. Наприклад, агломеративні методи реалізовані в пакеті SPSS, дивизимні методи – у пакеті Statgraf.

Ієрархічні методи кластеризації різняться правилами побудови кластерів. За правила виступають критерії, які використовуються при вирішенні питання про «схожість» об'єктів при об'єднанні їх в групу (агломеративні методи) або поділу на групи (дивизимні методи).

Ієрархічні методи кластерного аналізу використовуються при невеликих обсягах наборів даних.

Перевагою ієрархічних методів кластеризації є їхня наочність.

Ієрархічні алгоритми пов'язані з побудовою дендрограм (від грецького dendron – «дерево»), які є результатом ієрархічного кластерного аналізу.

Дендрограма описує близькість окремих точок і кластерів один до одного, представляє в графічному вигляді послідовність об'єднання (поділу) кластерів.

Дендрограма (dendrogram) – деревоподібна діаграма, що містить *n* рівнів, кожний з яких відповідає одному з кроків процесу послідовного укрупнення кластерів. Дендрограму також називають деревоподібною схемою, деревом об'єднання кластерів, деревом ієрархічної структури.

Дендрограма являє собою вкладене угруповання об'єктів, яке змінюється на різних рівнях ієрархії.

Існує багато способів побудови дендограмм. У дендограмі об'єкти можуть розташовуватися вертикально або горизонтально. Приклад вертикальної дендрограми наведений на рис. 10.4.

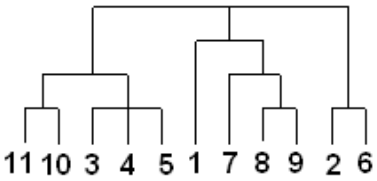


Рисунок 10.4 – Приклад дендрограмми

Числа 11, 10, 3 і т.д. відповідають номерам об'єктів або спостережень вихідної вибірки. Бачимо, що на першому кроці кожне спостереження представляє один кластер (вертикальна лінія), на другому кроці спостерігаємо об'єднання таких спостережень: 11 і 10; 3, 4 і 5; 8 і 9; 2 і 6. На другому кроці триває об'єднання в кластери: спостереження 11, 10, 3, 4, 5 і 7, 8, 9. Цей процес триває доти, поки всі спостереження не об'єднаються в один кластер.

Міри подібності. Для обчислення відстані між об'єктами використовуються різні міри подібності, їх називають також метриками або функціями відстаней. На початку теми розглянуто евклідову відстань, це найбільш популярна міра подібності.

Квадрат евклідової відстані. Для надання більшої ваги більш віддаленим один від одного об'єктів можемо скористатися квадратом евклідової відстані шляхом піднесення у квадрат стандартної евклідової відстані.

Манхеттенска відстань (відстань міських кварталів), також називається «хемінговою» або «сіті-блок» відстанню. Ця відстань розраховується як середня

різниць по координатах. У більшості випадків ця міра відстані приводить до результатів, подібних розрахункам відстані евкліда. Однак, для цієї міри вплив окремих викидів менший, ніж при використанні евклидової відстані, оскільки тут координати не підносяться до квадру.

Відстань Чебишева. Цю відстань варто використовувати, коли необхідно визначити два об'єкти як «різні», якщо вони відрізняються за якимось одним виміром.

Відсоток незгоди. Ця відстань обчислюється, якщо дані є категоріальними.

Методи об'єднання або зв'язки. Коли кожний об'єкт являє собою окремий кластер, відстані між цими об'єктами визначаються обраною мірою. Виникає таке питання – як визначити відстані між кластерами? Існують різні правила – методи об'єднання або зв'язки для двох кластерів.

Метод найближчого сусіда або одиночний зв'язок. Тут відстань між двома кластерами визначається відстанню між двома найбільш близькими об'єктами (найближчими сусідами) у різних кластерах. Цей метод дозволяє виділяти кластери як завгодно складної форми за умови, що різні частини таких кластерів з'єднані ланцюжками близьких один до одного елементів. У результаті роботи цього методу кластери представляються довгими «ланцюжками» або «волокнистими» кластерами, «зчепленими разом» тільки окремими елементами, які випадково виявилися ближче інших 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 із різних «ділянок». Якщо ж кластери мають до певної міри подовжену форму або їх природній тип є «ланцюговим», то цей метод не слід використовувати.

Метод Варда (Ward's method). За відстань між кластерами береться приріст суми квадратів відстаней об'єктів до центрів кластерів, одержуваний у результаті їх об'єднання (Ward, 1963). На відміну від інших методів кластерного аналізу для оцінки відстаней між кластерами, тут використовуються методи дисперсійного аналізу. На кожному кроці алгоритму поєднуються такі два кластери, які приводять до мінімального збільшення цільової функції, тобто внутрішньо групової суми квадратів. Цей метод спрямований на об'єднання близько розташованих кластерів і «прагне» створювати кластери малого розміру.

Метод незваженого попарного середнього (метод незваженого попарного арифметичного середнього – unweighted pair-group method using arithmetic averages, UPGMA (Sneath, Sokal, 1973)). За відстань між двома кластерами береться середня відстань між усіма парами об'єктів у них. Цей метод слід використовувати, якщо об'єкти дійсно походять із різних «ділянок», у випадках присутності кластерів «ланцюгового» типу, при припущенні нерівних розмірів кластерів.

Метод зваженого попарного середнього (метод зваженого попарного арифметичного середнього – weighted pair-group method using arithmetic averages, WPGM A (Sneath, Sokal, 1973)). Цей метод схожий на метод незваженого попарного середнього, різниця полягає лише в тому, що тут як ваговий

коефіцієнт використовується розмір кластера (число об’єктів, що втримуються в кластері). Рекомендується використовувати саме при наявності припущення про кластери різних розмірів.

Незважений центроїдний метод (метод незваженого попарного центроїдного усереднення – unweighted pair-group method using the centroid average (Sneath and Sokal, 1973)). За відстань між двома кластерами в цьому методі береться відстань між їхніми центрами ваги.

Зважений центроїдний метод (метод зваженого попарного центроїдного усереднення – weighted pair-group method using the centroid average, WPGMC (Sneath, Sokal 1973)). Цей метод схожий на попередній, різниця полягає в тому, що для обліку різниці між розмірами кластерів (числа об’єктів у них), використовуються ваги. Використовують переважно у випадках, якщо є припущення щодо істотних відмінностей у розмірах кластерів.

3. Ієрархічний кластерний аналіз

Розглянемо процедуру ієрархічного кластерного аналізу в пакеті SPSS (SPSS), в якому ця процедура передбачає угруповання як об’єктів (рядків матриці даних), так і змінних (стовпців). Можна вважати, що в останньому випадку роль об’єктів відіграють змінні, а роль змінних – стовпці.

У цьому методі реалізується ієрархічний агломеративний алгоритм, зміст якого полягає в такому. Перед початком кластеризації всі об’єкти вважаються окремими кластерами, у ході алгоритму вони поєднуються. Спочатку вибирається пара найближчих кластерів, які поєднуються в один кластер. У результаті кількість кластерів стає рівним *N-1*. Процедура повторюється, поки всі класи не об’єднаються. На будь-якому етапі об’єднання можна перервати, одержавши потрібне число кластерів. Отже, результат роботи алгоритму агрегування залежить від способів обчислення відстані між об’єктами й визначення близькості між кластерами.

Для визначення відстані між парою кластерів можуть бути сформульовані різні підходи. З урахуванням цього в SPSS передбачені такі методи:

- Середня відстань між кластерами (Between-groups linkage), установлюється за замовчуванням.
- Середня відстань між усіма об’єктами пари кластерів з урахуванням відстаней усередині кластерів (Within-groups linkage).
- Відстань між найближчими сусідами – найближчими об’єктами кластерів (Nearest neighbor).
- Відстань між самими далекими сусідами (Furthest neighbor).
- Відстань між центрами кластерів (Centroid clustering) або центроидный метод. Недоліком цього методу є те, що центр об’єднаного кластера обчислюється як середнє центрів поєднуваних кластерів, без обліку їх обсягу.
- Метод Варда.
- Метод медіан – той же центроїдний метод, але центр об’єднаного кластера обчислюється як середнє всіх об’єктів (Median clustering).

Приклад ієрархічного кластерного аналізу. Порядок агломерації (протокол об'єднання кластерів) представлених раніше даних наведено в таблиці 10.1. У протоколі зазначені такі позиції:

- Stage – стадії об'єднання (крок);
- Cluster Combined – поєднувані кластери (після об'єднання кластер ухвалює мінімальний номер з номерів поєднуваних кластерів);
- Coefficients – коефіцієнти.

Так, у колонку Cluster Combined можна побачити порядок об'єднання в кластери: на першому кроці були об'єднані спостереження 9 і 10, вони утворюють кластер під номером 9, кластер 10 в оглядовій таблиці більше не з'являється. На наступному кроці відбувається об'єднання кластерів 2 і 14, далі 3 і 9, і т.д.

Таблиця 10.1 − Порядок агломерації

Stage	Cluster Combined	Coefficients	Результат
1	1	-	-
2	2	14	1,461E-02
3	3	9	1,461E-02
4	5	8	1,461E-02
5	6	7	1,461E-02
6	3	13	3,490E-02
7	2	11	3,651E-02
8	4	5	4,144E-02
9	2	6	5,118E-02
10	4	12	0,105
11	1	3	0,120
12	1	4	1,217
13	1	2	7,516

У колонку Coefficients наведена кількість кластерів, яку варто було б уважати оптимальною; під значенням цього показника мається на увазі відстань між двома кластерами, визначене на підставі обраної міри відстані. У нашому випадку це квадрат відстані, обчислений із використанням стандартизованих значень. Процедура стандартизації використовується для виключення ймовірності того, що класифікацію будуть визначати зміни, що мають найбільший розкид значень. У SPSS застосовуються такі види стандартизації:

- Z-Шкали (Z-Scores). Зі значень змінних віднімається їхнє середнє, і ці значення діляться на стандартне відхилення.
- Розкид від -1 до 1. Лінійним перетворенням змінних домагаються розкиду значень від -1 до 1.
- Розкид від 0 до 1. Лінійним перетворенням змінних домагаються розкиду значень від 0 до 1.
- Максимум 1. Значення змінних діляться на їхній максимум.
- Середнє 1. Значення змінних діляться на їхнє середнє.
- Стандартне відхилення 1. Значення змінних діляться на стандартне відхилення.

Крім того, можливі перетворення самих відстаней, зокрема, можна відстані замінити їхніми абсолютними значеннями, це актуально для коефіцієнтів кореляції. Можна також усі відстані перетворити так, щоб вони змінювалися від 0 до 1.

Визначення кількості кластерів. Існує проблема визначення числа кластерів. Іноді можна апріорно визначити це число. Однак у більшості випадків число кластерів визначається в процесі агломерації/поділу безлічі об'єктів.

Процесу угруповання об'єктів в ієрархічному кластерному аналізі відповідає поступове зростання коефіцієнта, який називається критерієм E. Стрибкоподібне збільшення значення критерію E можна визначити як характеристику числа кластерів, які дійсно існують у досліджуваному наборі даних. Отже, цей спосіб зводиться до визначення стрибкоподібного збільшення деякого коефіцієнта, який характеризує перехід від сильно зв'язаного до слабко зв'язаного стану об'єктів.

У таблиці 10.1 ми бачимо, що значення поля Coefficients збільшується стрибкоподібно, отже, об'єднання в кластери слід зупинити, інакше буде відбуватися об'єднання кластерів, що перебувають на відносно великій відстані один від одного. У цьому прикладі це стрибок з 1,217 до 7,516. Оптимальним уважається кількість кластерів, рівне різниці кількості спостережень (14) і кількості кроків до стрибкоподібного збільшення коефіцієнта (12).

Отже, після створення двох кластерів об'єднань більше проводити не слід, хоча візуально очікувалася поява трьох кластерів.

Агрегування даних може бути презентовано графічно у вигляді дендрограми. Вона визначає об'єднані кластери й значення коефіцієнтів на кожному кроці агломерації (відображені значення коефіцієнтів, наведені на шкалі від 0 до 25).

Дендрограма для цього прикладу наведена на рис. 10.5. Розріз дерева агрегування вертикальною рисою дав нам два кластери, що складаються із 9 і 5 об'єктів.

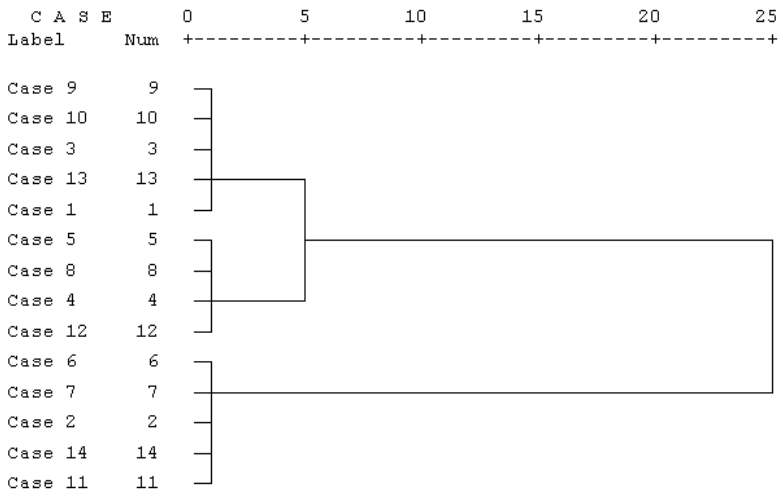


Рисунок 10.5 – Дендрограмма процесу злиття

На верхній лінії по горизонталі відзначені номери кроків алгоритму, усього алгоритму треба було 25 кроків для об'єднання всіх об'єктів в один кластер.

Питання для самоконтролю

1. Які існують групи задач кластерного аналізу?
2. Як вимірюється відстань між двома точками? Які функції відстані ви знаєте?
3. Дайте визначення математичним характеристикам кластеру: центр, радіус, середньоквадратичне відхилення, розмір кластера?
4. Які найпоширеніші способи нормування (normalization) змінних?
5. Які існують групи кластерного аналізу? Чим вони відрізняються?
6. Як відбувається визначення кількості кластерів?

Тема 11. Методи кластерного аналізу. Ітеративні методи

План

1. Алгоритми неієрархічної кластеризації.
2. Факторний аналіз.
3. Ітеративні методи кластеризації.
4. Порівняльний аналіз ієрархічних і неієрархічних методів кластеризації.

Мета вивчення теми: вивчити неієрархічні алгоритми кластеризації, факторний аналіз, ітеративні методи кластеризації.

Перелік ключових слів та понять із теми

Кластерний аналіз, кластеризація, неієрархічні кластеризація, факторний аналіз, ітеративні методи, алгоритм k-середніх

Теоретичні відомості з теми

1. Алгоритми неієрархічної кластеризації

При великій кількості спостережень ієрархічні методи кластерного аналізу непридатні. У таких випадках використовують **неієрархічні методи**, засновані на поділі, які являють собою **ітеративні методи дроблення вихідної сукупності**. У процесі розподілу нові кластери формуються доти, поки не буде виконане **правило зупинки**.

Така неієрархічна кластеризація полягає в поділі набору даних на певну кількість окремих кластерів. **Існує два підходи**. Перший полягає у визначенні границь кластерів як найбільш щільних ділянок у багатомірному просторі вихідних даних, тобто визначення кластера там, де є велике «згущення точок». Другий підхід полягає в мінімізації міри відмінності об’єктів.

Найпоширеніший серед неієрархічних методів алгоритм *k*-середніх, також називають **швидким кластерним аналізом**. Повний опис алгоритму можна знайти в роботі Хартігана і Вонга (Hartigan and Wong, 1978). На відміну від ієрархічних методів, які не вимагають попередніх припущень щодо числа кластерів, для можливості використання цього методу необхідно мати гіпотезу про найбільш імовірну кількість кластерів.

Алгоритм *k*-середніх будує *k* кластерів, розташованих на максимально можливо великих відстанях один від одного. Основний тип задач, які вирішує алгоритм *k*-середніх, – наявність припущень (гіпотез) щодо числа кластерів, при цьому вони повинні бути різні настільки, наскільки це можливо. Вибір числа *k* може базуватися на результатах попередніх досліджень, теоретичних міркуваннях або інтуїції.

Загальна ідея алгоритму: задане фіксоване число *k* кластерів спостереження зіставляється кластерам так, що середні в кластері (для всіх змінних) максимально можливо відрізняються одна від одної.

Опис алгоритму:

1. Первісний розподіл об’єктів по кластерах.

Вибирається число *k*, і на першому кроці ці точки вважаються «центрами» кластерів. Кожному кластеру відповідає один центр.

Вибір початкових центроїдів може здійснюватися в такий спосіб:

- вибір «-спостережень для максимізації початкової відстані;
- випадковий вибір *k*-спостережень;
- вибір перших *k*-спостережень.

У результаті кожний об’єкт призначений певному кластеру.

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 триває доти, 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.

На рис. 11.1 наведений приклад роботи алгоритму *k*-середніх для *k*, рівного двом.

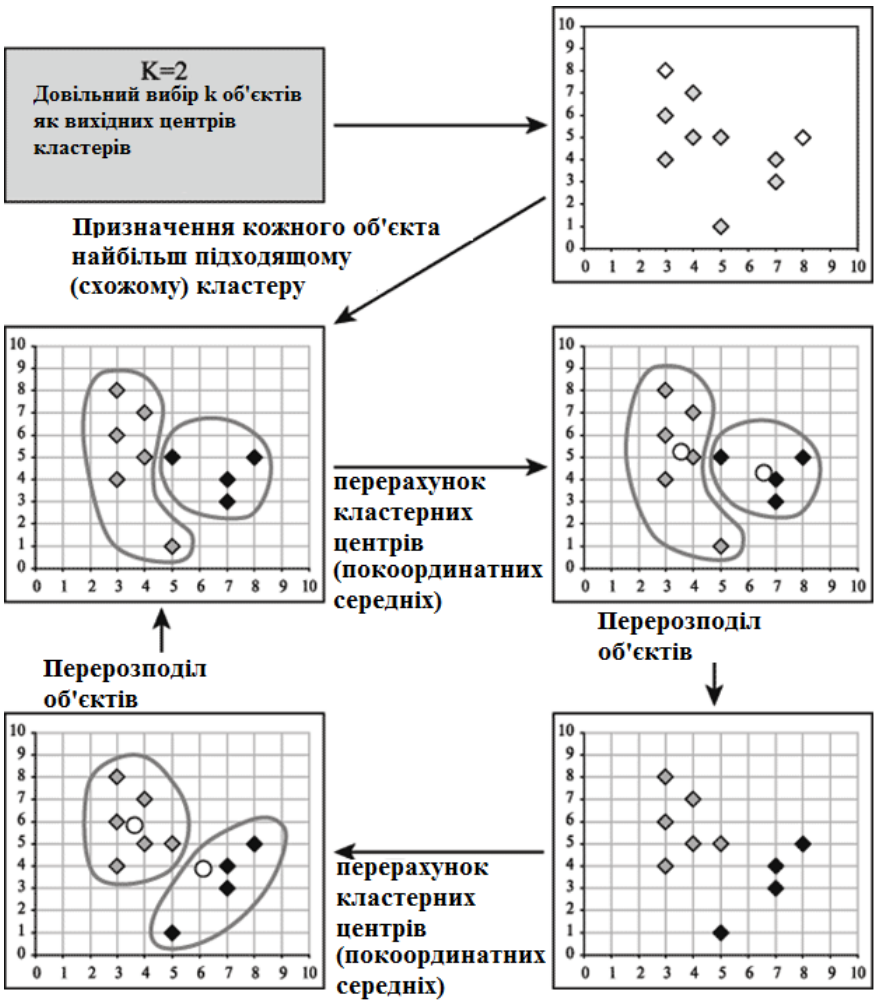


Рисунок 11.1 – Приклад роботи алгоритму *k*-середніх (*k*=2)

Вибір числа кластерів є складним питанням. Якщо немає припущень щодо цього числа, рекомендують створити 2 кластера, потім 3, 4, 5 і т.д., порівнюючи отримані результати.

Перевірка якості кластеризації. Після одержання 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 гарній кластеризації повинні бути отримані дуже відмінні середні для всіх вимірів або хоча б більшої їхньої частини.

Переваги алгоритму k-середніх:

- простота використання;
- швидкість використання;
- зрозумілість і прозорість алгоритму.

Недоліки алгоритму k-середніх:

- алгоритм занадто 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.

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

Розглянемо приклад. Є база даних клієнтів фірми, яких слід розбити на однорідні групи. Кожний клієнт описується за допомогою 25 змінних.

Використання такого великого числа змінних призводить до виділення кластерів нечіткої структури. У результаті аналітикові досить складно інтерпретувати отримані кластери.

Більш зрозумілі й прозорі результати кластеризації можуть бути отримані, якщо замість множини вихідних змінних використовувати якісь узагальнені змінні або критерії, що містять у стислому вигляді інформацію про зв’язки між змінними. Тобто виникає задача зниження розмірності даних. Вона може вирішуватися за допомогою різних методів; один із найпоширеніших – факторний аналіз.

2. Факторний аналіз

Факторний аналіз – це метод, що застосовується для вивчення взаємозв’язків між значеннями змінних.

Взагалі, факторний аналіз переслідує **дві мети**:

- скорочення числа змінних;
- класифікацію змінних – визначення структури взаємозв’язків між змінними.

Відповідно, факторний аналіз може використовуватися для розв’язання задач скорочення розмірності даних або для розв’язання задач класифікації.

Критерії або головні фактори, виділені в результаті факторного аналізу, містять у стислому вигляді інформацію про існуючі зв'язки між змінними. Ця інформація дозволяє одержати кращі результати кластеризації та краще пояснити семантику кластерів. Самим факторам може бути повідомлений певний зміст.

За допомогою факторного аналізу велике число змінних зводиться до меншого числа незалежних величин, які називаються факторами.

Фактор в «стислому» вигляді містить інформацію про декілька змінних. В один фактор поєднуються змінні, які сильно корелюють між собою. У результаті факторного аналізу відшукуються такі комплексні фактори, які якомога більш повно пояснюють зв'язки між розглянутими змінними.

На першому кроці факторного аналізу здійснюється стандартизація значень змінних, необхідність якої була розглянута в попередній лекції.

Факторний аналіз опирається на гіпотезу про те, що аналізовані змінні є непрямими проявами порівняно невеликого числа якихось схованих факторів.

Факторний аналіз – це сукупність методів, орієнтованих на виявлення й аналіз схованих залежностей між спостережуваними змінними. Сховані залежності також називають латентними.

Один із методів факторного аналізу – метод головних компонентів – заснований на припущенні про незалежність факторів один від одного.

3. Ітеративні методи кластеризації

Ітеративна кластеризація в SPSS. Звичайно в статистичних пакетах реалізований широкий арсенал методів, що дозволяє спочатку провести скорочення розмірності набору даних (наприклад, за допомогою факторного аналізу), а потім уже безпосередньо кластеризацію (наприклад, методом швидкого кластерного аналізу). Розглянемо цей варіант проведення кластеризації в пакеті SPSS.

Для скорочення розмірності вихідних даних скористаємося факторним аналізом. Для цього виберемо в меню: Analyze (Аналіз)/Data Reduction (Перетворення даних)/Factor (Факторний аналіз):

За допомогою кнопки Extraction/(Відбір) слід вибрати метод відбору. Ми залишимо обраний за замовчуванням аналіз головних компонентів, який згадувався вище. Також слід вибрати метод обертання – виберемо один із найбільш популярних – метод Варимакса. Для збереження значень факторів у вигляді змінних у закладці «Значення» необхідно поставити оцінку «Save as variables» (Зберегти як змінні).

У результаті цієї процедури користувач одержує звіт «Пояснена сумарна дисперсія», за яким видна кількість відібраних факторів – це ті компоненти, власні значення яких перевищують одиницю.

Отримані значення факторів, яким звичайно привласнюються назви fact1_1, fact1_2 і т.д., використовуємо для проведення кластерного аналізу методом *k*-середніх. Для проведення швидкого кластерного аналізу виберемо в меню: Analyze (Аналіз)/Classify(Класифікувати)/K-Means Cluster: (Кластерный аналіз методом *k*-середніх).

У діалоговому вікні K Means Cluster Analysis (Кластерний аналіз методом *k*-середніх) необхідно помістити факторні змінні fact1_1, fact1_2 і т.д. у поле тестованих змінних. Тут же необхідно вказати кількість кластерів і кількість ітерацій.

У результаті цієї процедури одержуємо звіт з висновком значень центрів сформованих кластерів, кількості спостережень у кожному кластері, а також з додатковою інформацією, заданої користувачем.

Таким чином, алгоритм *k*-середніх ділить сукупність вихідних даних на задану кількість кластерів. Для можливості візуалізації отриманих результатів слід скористатися одним із графіків, наприклад, діаграмою розсіювання. Однак традиційна візуалізація можлива для обмеженої кількості вимірів, оскільки людина може сприймати тільки тривимірний простір. Тому, якщо ми аналізуємо більш трьох змінних, слід використовувати спеціальні багатомірні методи представлення інформації.

Ітеративні методи кластеризації різняться вибором таких параметрів:

- початкової точки;
- правилом формування нових кластерів;
- правилом зупинки.

Вибір методу кластеризації залежить від кількості даних і від того, чи є необхідність працювати одночасно з декількома типами даних.

У пакеті SPSS, наприклад, при необхідності роботи з кількісними (наприклад, дохід) і з категоріальними (наприклад, родиний стан) змінними, а також з досить великими обсягами даних використовується метод Двоетапного кластерного аналізу, який являє собою масштабовану процедуру кластерного аналізу, що дозволяє працювати з даними різних типів.

Для цього на першому етапі роботи записи попередньо кластеризуются у велику кількість суб-кластерів. На другому етапі отримані суб-кластери групуються в необхідну кількість. Якщо ця кількість невідома, процедура сама автоматично визначає її. За допомогою цієї процедури банківський працівник може, наприклад, виділяти групи людей, одночасно використовуючи такі показники як вік, стать і рівень доходу. Отримані результати дозволяють визначити клієнтів, вхідних у групи ризику неповернення кредиту.

Процес кластерного аналізу. Рекомендовані етапи.

У загальному випадку всі етапи кластерного аналізу взаємозалежні, і розв'язки, прийняті на одному з них, визначають дії на наступних етапах.

Аналітикові слід вирішити, використовувати всі спостереження чи виключити деякі дані або вибірки з набору даних:

- Вибір метрики й методу стандартизації вихідних даних.
- Визначення кількості кластерів (для ітеративного кластерного аналізу).
- Визначення методу кластеризації (правила об'єднання або зв'язки).

На думку багатьох фахівців, вибір методу кластеризації є вирішальним при визначенні форми й специфіки кластерів.

Аналіз результатів кластеризації. На цьому етапі вирішуються такі питання: чи не є отримана розбивка на кластери випадковою; чи є розбивка надійною й стабільною на підвибірках даних; чи існує взаємозв'язок між

результатами кластеризації й змінними, які не брали участь у процесі кластеризації; чи можна інтерпретувати отримані результати кластеризації.

Перевірка результатів кластеризації. Результати кластеризації також повинні бути перевірені формальними й неформальними методами. Формальні методи залежать від того методу, який використовувався для кластеризації. Неформальні методи включають такі процедури перевірки якості кластеризації:

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

Один із варіантів перевірки якості кластеризації – використання декількох методів і порівняння отриманих результатів. Відсутність подібності не буде означати некоректність результатів, але присутність схожих груп вважається ознакою якісної кластеризації.

Складності й проблеми, які можуть виникнути при застосуванні кластерного аналізу.

Як і будь-які інші методи, методи кластерного аналізу мають певні слабкі сторони, тобто деякі складності, проблеми й обмеження.

При проведенні кластерного аналізу слід враховувати, що результати кластеризації залежать від критеріїв розбивки сукупності вихідних даних. При зниженні розмірності даних можуть виникнути певні викривлення, за рахунок узагальнень можуть згубитися деякі індивідуальні характеристики об’єктів.

Існує ряд складностей, які слід продумати перед проведенням кластеризацію, зокрема:

- складність вибору характеристик, на основі яких проводиться кластеризація. Необдуманий вибір приводить до неадекватної розбивки на кластери й, як наслідок, – до неправильного розв’язання задачі;
- складність вибору методу кластеризації. Цей вибір вимагає непоганого знання методів і передумов їх використання. Щоб перевірити ефективність конкретного методу в певній предметній області, доцільно застосувати таку процедуру: розглядають декілька апріорі різних між собою груп і перемішують їхніх представників між собою випадковим чином. Далі проводиться кластеризація для відновлення вихідної розбивки на кластери. Частка збігів об’єктів у виявлених і вихідних групах є показником ефективності роботи методу;
- проблема вибору числа кластерів. Якщо немає ніяких відомостей щодо можливого числа кластерів, необхідно провести ряд експериментів і, за результатами перебору різного числа кластерів, вибрати оптимальне їхнє число;
- проблема інтерпретації результатів кластеризації. Форма кластерів у більшості випадків визначається вибором методу об’єднання. Однак слід враховувати, що конкретні методи прагнуть створювати кластери певних форм, навіть якщо в досліджуваному наборі даних кластерів насправді немає.

4. Порівняльний аналіз ієрархічних і неієрархічних методів

кластеризації

Порівняльний аналіз ієрархічних і неієрархічних методів кластеризації. Перед проведенням кластеризації в аналітика може виникнути питання, якій групі методів кластерного аналізу віддати перевагу. Вибираючи між ієрархічними й неієрархічними методами, необхідно враховувати такі їхні особливості.

Неієрархічні методи виявляють більш високу стабільність стосовно шумів і викидів, некоректного вибору метрики, включення незначущих змінних у набір, що брав участь у кластеризації. Ціною, яку доводиться платити за ці переваги методу, є слово «апріорі». Аналітик повинен заздалегідь визначити кількість кластерів, кількість ітерацій або правило зупинки, а також деякі інші параметри кластеризації. Це особливо складно починаючим фахівцям.

Якщо немає припущень щодо числа кластерів, рекомендують використовувати ієрархічні алгоритми. Однак, якщо обсяг вибірки не дозволяє це зробити, можливий шлях – проведення низки експериментів із різною кількістю кластерів, наприклад, почати розбивку сукупності даних із двох груп і, поступово збільшуючи їх кількість, порівнювати результати. За рахунок такого «варіювання» результатів досягається значно більша гнучкість кластеризації.

Ієрархічні методи, на відміну від неієрархічних, відмовляються від визначення кількості кластерів, а будують повне дерево вкладених кластерів.

Складності ієрархічних методів кластеризації: обмеження обсягу набору даних; вибір міри близькості; негнучкість отриманих класифікацій.

Перевага цієї групи методів у порівнянні з неієрархічними методами – їх наочність і можливість одержати детальне представлення про структуру даних.

При використанні ієрархічних методів існує можливість досить легко ідентифікувати викиди в наборі даних і, як результат, 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, про який уже згадувалося в цій лекції. Це питання кластеризації всієї сукупності даних або ж її вибірки. Названий аспект вагомий для обох розглянутих груп методів, однак він більш критичний для ієрархічних методів. Ієрархічні методи не можуть працювати з більшими 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 розв’язку задач кластеризації припустима нестатистична інтерпретація отриманих результатів, а також досить велика різноманітність варіантів поняття кластера. Така нестатистична інтерпретація дає можливість аналітикові одержати задовольняючі його результати кластеризації, що при використанні інших методів часто буває скрутним.

Нові алгоритми й деякі модифікації алгоритмів кластерного аналізу. Методи, які ми розглянули в цій і попередній лекціях, є «класикою» кластерного аналізу. До останнього часу основним критерієм, по якому оцінювався алгоритм

кластеризації, була якість кластеризації: вважалося, щоб увесь набір даних вміщався в оперативній пам'яті.

Однак зараз, у зв'язку з появою надвеликих баз даних, з'явилися нові вимоги, яким повинен задовольняти алгоритм кластеризації. Основна з них, як уже згадувалося в попередній темі, – 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 й використання структур даних, підтримуваних СУБД.

Розроблені алгоритми, у яких методи ієрархічної кластеризації інтегровані з іншими методами. До таких алгоритмів ставляться: BIRCH, CURE, CHAMELEON, ROCK.

Алгоритм BIRCH (Balanced Iterative Reducing and Clustering using Hierarchies) запропонований Тьян Зангом і його колегами.

Завдяки узагальненим представленням кластерів, швидкість кластеризації збільшується, алгоритм при цьому має більше масштабування.

У цьому алгоритмі реалізований двоетапний процес кластеризації.

У ході першого етапу формується попередній набір кластерів. На другому етапі до виявлених кластерів застосовуються інші алгоритми кластеризації – придатні для роботи в оперативній пам'яті.

Розглянемо аналогію, що описує цей алгоритм. Якщо кожний елемент даних уявити собі як бусинку, що лежить на поверхні стола, то кластери бусин можна «замінити» тенісними кульками й перейти до більш детального вивчення кластерів тенісних кульок. Кількість бусин може виявитися досить велике, однак діаметр тенісних кульок можна підібрати таким чином, щоб на другому етапі можна було, застосувавши традиційні алгоритми кластеризації, визначити дійсну складну форму кластерів.

Алгоритм Wavecluster являє собою алгоритм кластеризації на основі хвильових перетворень. На початку роботи алгоритму дані узагальнюються шляхом накладення на простір даних багатомірних ґрат. На подальших кроках алгоритму аналізуються не окремі точки, а узагальнені характеристики точок, що потрапили в одне гніздо ґрат. У результаті такого узагальнення необхідна інформація розміщується в оперативній пам'яті. На наступних кроках для визначення кластерів алгоритм застосовує хвильове перетворення до узагальнених даних.

Головні особливості Wavecluster:

1) складність реалізації;

2) алгоритм може виявляти кластери довільних форм;

3) алгоритм не чутливий до шумів;

4) алгоритм застосуємо тільки до даних низкої розмірності.

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

Викладемо коротко суть алгоритму. Алгоритм CLARA витягає множину зразків із 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 обраного за зразок набору даних. Гарна кластеризація на обраному наборі може не дати гарну кластеризацію на всій множині даних.

Алгоритми Clarans, CURE, Dbscan формулює задачу кластеризації як випадковий пошук у графові. У результаті роботи цього алгоритму сукупність вузлів графа являє собою розбивку множини даних на число кластерів, визначене користувачем. «Якість» отриманих кластерів визначається за допомогою критеріальної функції. Алгоритм Clarans сортує всі можливі розбивки множини даних у пошуках прийнятного розв'язку. Пошук розв'язку зупиняється в тому вузлі, де досягається мінімум серед визначеного 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 BIRCH, Clarans, CURE, Dbscan є та обставина, що вони вимагають задання деяких порогів щільності точок, а це не завжди прийнятне. Ці обмеження зумовлені 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 – подолати недоліки алгоритмів, що існують на сьогодні.

Питання для самоконтролю

1. За яких умов використовуються неієрархічні методи кластерного аналізу?
2. Наведіть опис алгоритму *k*-середніх? Приведіть приклад при *k*=2.
3. Як відбувається перевірка якості кластеризації?
4. Які переваги алгоритму *k*-середніх?
5. Які існують недоліки алгоритму *k*-середніх?
6. Дайте визначення поняттю факторний аналіз?

Тема 12. Методи пошуку асоціативних правил

План

1. Класифікація нейронних мереж.
2. Вибір структури нейронної мережі.
3. Карти Кохонена.
4. Карта входів та виходів нейронів.
5. Що таке асоціативні правила?
6. Алгоритми пошуку асоціативних правил.
7. Методи пошуку асоціативних правил.

Мета вивчення теми: засвоїти поняття нейронної мережі; вивчити асоціативні правила та алгоритм їх пошуку.

Перелік ключових слів та понять із теми

Нейронна мережа, карта Кохонена, мережа Хопфілда, асоціативне правило, транзакцій на база даних

Теоретичні відомості з теми

1. Класифікація нейронних мереж

Одна з можливих класифікацій нейронних мереж – за спрямованістю зв'язків. Нейронні мережі бувають зі зворотними зв'язками й без зворотних зв'язків.

Мережі без зворотних зв'язків:

- мережі зі зворотним поширенням помилки. Мережі цієї групи характеризуються фіксованою структурою, ітераційним навчанням, коректуванням ваг за помилками.
- інші мережі (когнитрон, неокогнитрон, інші складні моделі).

Перевагами мереж без зворотних зв'язків є простота їх реалізації й гарантоване одержання відповіді після проходження даних по шарах.

Недоліком цього виду мереж вважається мінімізація розмірів мережі – нейрони багаторазово беруть участь в обробці даних.

Менший обсяг мережі полегшує процес навчання.

Мережі зі зворотними зв'язками:

- мережі Хопфілда (задачі асоціативної пам'яті).
- мережі Кохонена (задачі кластерного аналізу).

Перевагами мереж зі зворотними зв'язками є складність навчання, викликана більшим числом нейронів для алгоритмів того самого рівня складності.

Недоліки цього виду мереж – необхідність спеціальних умов, що гарантують збіжність обчислень.

Інша класифікація нейронних мереж: мережі прямого поширення й рекуррентні мережі.

Мережі прямого поширення:

- Персептрони.
- Мережа Back Propagation.
- Мережа зустрічного поширення.
- Карта Кохонена.

Рекуррентні мережі. Характерна риса таких мереж – наявність блоків динамічної затримки й зворотних зв'язків, що дозволяє їм обробляти динамічні моделі:

- Мережа Хопфілда.
- Мережа Елмана – мережа, що складається із двох шарів, у якій схований шар охоплений динамічним зворотним зв'язком, що дозволяє врахувати передісторію спостережуваних процесів і нагромадити інформацію для вироблення правильної стратегії управління. Ці мережі застосовуються в системах управління об'єктами, що рухаються.

Нейронні мережі можуть навчатися з учителем або без нього.

При навчанні з учителем для кожного навчального вхідного прикладу потрібне знання правильної відповіді або функції оцінки якості відповіді. Таке навчання називають керованим. Нейронній мережі пред'являються значення вхідних і вихідних сигналів, а вона за певним алгоритмом підбудовує ваги синоптичних зв'язків. У процесі навчання проводиться корегування ваг мережі за результатами порівняння фактичних вихідних значень із вхідними, відомими заздалегідь.

При навчанні без учителя розкривається внутрішня структура даних або кореляції між зразками в наборі даних. Виходи нейронної мережі формуються самостійно, а ваги змінюються за алгоритмом, що враховує тільки вхідні й похідні від них сигнали. Це навчання називають також некерованим. У результаті такого навчання об'єкти або приклади розподіляються по категоріях, самі категорії та їх кількість можуть бути заздалегідь не відомі.

Підготовка даних для навчання. При підготовці даних для навчання нейронної мережі необхідно звертати увагу на такі істотні моменти.

Кількість спостережень у наборі даних. Слід враховувати той фактор, що чим більше розмірність даних, тим більше часу потрібно для навчання мережі.

Робота з викидами. Слід визначити наявність викидів і оцінити необхідність їх присутності у вибірці.

Навчальна вибірка повинна бути представницькою (репрезентативною).

Навчальна вибірка не повинна містити протиріч, тому що нейронна мережа однозначно зіставляє вихідні значення із вхідними.

Нейронна мережа працює тільки із числовими вхідними даними, тому важливим етапом при підготовці даних є перетворення й кодування даних.

При використанні на вхід нейронної мережі слід подавати значення з того діапазону, на якому вона навчалася. Наприклад, якщо при навчанні нейронної мережі на один з її входів подавалися значення від 0 до 10, то при її застосуванні на вхід слід подавати значення із цього ж діапазону або прилеглих.

Існує поняття нормалізації даних. Метою нормалізації значень є перетворення даних до вигляду, який найбільше підходить для обробки, тобто

дані, що надходять на вхід, повинні мати числовий тип, а їх значення повинні бути розподілені в певному діапазоні. Нормалізатор може приводити дискретні дані до набору унікальних індексів або перетворювати значення, що лежать в довільному діапазоні, у конкретний діапазон. Нормалізація виконується шляхом розподілу кожного компонента вхідного вектора на довжину вектора, що перетворює вхідний вектор в одиничний.

2. Вибір структури нейронної мережі

Вибір структури нейронної мережі зумовлюється специфікою й складністю розв'язуваної задачі. Для розв'язання деяких типів задач розроблені оптимальні конфігурації.

У більшості випадків вибір структури нейронної мережі визначається на основі об'єднання досвіду й інтуїції розроблювача.

Однак існують основні принципи, якими слід керуватися при розробці нової конфігурації:

1) можливості мережі зростають зі збільшенням кількості гнізд мережі, щільності зв'язків між ними й кількості виділених шарів;

2) введення зворотних зв'язків поряд зі збільшенням можливостей мережі піднімає питання про динамічну стабільність мережі;

3) складність алгоритмів функціонування мережі (у тому числі, наприклад, введення декількох типів синапсів – збуджуючих, гальмуючих та ін.) також сприяє посиленню потужності нейронної мережі.

Питання про необхідні й достатні властивості мережі для розв'язання того або іншого типу задач являє собою цілий напрямок нейронної комп'ютерної науки. Оскільки проблема синтезу нейронної мережі суттєво залежить від розв'язуваної задачі, дати загальні докладні рекомендації важко. Очевидно, що процес функціонування НМ (нейронної мережі), тобто сутність дій, які вона здатна виконувати, залежить від величин синаптичних зв'язків. Розроблювач мережі повине задати певну структуру НМ, що відповідає якому-небудь завданню, та знайти оптимальні значення всіх змінних вагових коефіцієнтів (деякі синаптичні зв'язки можуть бути постійними).

3. Карти Кохонена

Карти Кохонена, карти, що самоорганізуються (Self-Organizing Maps). Мережі, що називаються картами Кохонена, – це один із різновидів нейронних мереж, однак вони принципово відрізняються від розглянутих вище, оскільки використовують неконтрольоване навчання. Нагадаємо, що при такому навчанні навчальна множина складається лише зі значень вхідних змінних, у процесі навчання немає порівняння виходів нейронів з еталонними значеннями. Можна сказати, що така мережа вчиться розуміти структуру даних.

Ідея мережі Кохонена належить фінському вченому Тойво Кохонену (1982 рік). Основний принцип роботи мереж – введення в правило навчання нейрона інформації щодо його розташування.

В основі ідеї мережі Кохонена лежить аналогія із властивостями людського мозку. Кора головного мозку людину являє собою плоский аркуш зі згорнутими складками. Отже, можна сказати, що вона має певні топологічні властивості (ділянки, відповідальні за близькі частини тіла, примикають одна до одної й усе зображення людського тіла відображається на цю двовимірну поверхню).

Задачі, що розв'язуються за допомогою карт Кохонена. Карти, що самоорганізуються, можуть використовуватися для розв'язання таких задач, як моделювання, прогнозування, пошук закономірностей у великих масивах даних, виявлення наборів незалежних ознак і стискання інформації.

Найпоширеніше застосування мереж Кохонена – розв'язання задачі класифікації без учителя, тобто кластеризації.

Нагадаємо, що при такій постановці задачі задано набір об'єктів, кожному з яких зіставлений рядок таблиці (вектор значень ознак). Потрібно розбити вихідну множину на класи, тобто для кожного об'єкта знайти клас, до якого він належить.

У результаті одержання нової інформації про класи можлива корекція існуючих правил класифікації об'єктів.

Два з розповсюджених застосувань карт Кохонена: розвідницький аналіз даних і виявлення нових явищ.

Розвідницький аналіз даних. Мережа Кохонена здатна розпізнавати кластери в даних, а також установлювати близькість класів. Отже, користувач може поліпшити своє розуміння структури даних, щоб потім уточнити нейромережеву модель. Якщо в даних розпізнані класи, то їх можна позначити, після чого мережа зможе вирішувати задачу класифікації. Мережі Кохонена можна використовувати й у тих задачах класифікації, де класи вже задані, – тоді перевага буде в тому, що мережа зможе виявити подібність між різними класами.

Виявлення нових явищ. Мережа Кохонена розпізнає кластери в навчальних даних і відносить усі дані до тих або інших кластерів. Якщо після цього мережа зустрінеться з набором даних, несхожим ні на один із відомих зразків, то вона не зможе класифікувати такий набір і тим самим виявить його новизну.

Мережа Кохонена, на відміну від багатошарової нейронної мережі, дуже проста; вона являє собою два шари: вхідний і вихідний. Її також називають самоорганізованою картою. Елементи карти розташовуються в деякому просторі, як правило, двовимірному. Мережа Кохонена зображена на рис. 12.1.

Мережа Кохонена навчається методом послідовних наближень. У процесі навчання таких мереж на входи подаються дані, але мережа при цьому підбудовується не під еталонне значення виходу, а під закономірності у вхідних даних. Починається навчання з обраного випадковим чином вихідного розташування центрів.

У процесі послідовної подачі на вхід мережі навчальних прикладів визначається найбільш схожий нейрон (той, у якого скалярний добуток ваг і поданого на вхід вектора мінімальні). Цей нейрон оголошується переможцем і є центром при підстроюванні ваг у сусідніх нейронів. Таке правило навчання припускає «змагальне» навчання з урахуванням відстані нейронів від «нейрона-переможця».

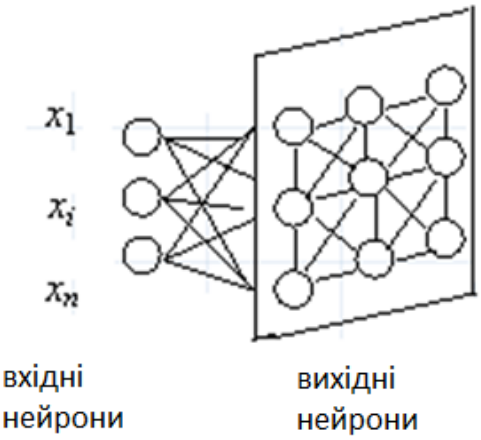


Рисунок 12.1 – Мережа Кохонена

Навчання при цьому полягає не в мінімізації помилки, а в підстроюванні ваг (внутрішніх параметрів нейронної мережі) для найбільшого збігу із вхідними даними.

Основний ітераційний алгоритм Кохонена послідовно проходить ряд епох, на кожній з яких обробляється один приклад із навчальної вибірки. Вхідні сигнали послідовно пред'являються мережі, при цьому бажані вихідні сигнали не визначаються. Після пред'явлення достатнього числа вхідних векторів синаптичні ваги мережі стають здатні визначити кластери. Ваги організують так, що топологічно близькі вузли чутливі до схожих вхідних сигналів.

У результаті роботи алгоритму центр кластера встановлюється в певній позиції, задовільним чином кластеризують приклади, для яких даний нейрон є «переможцем». У результаті навчання мережі необхідно визначити міру сусідства нейронів, тобто околицю нейрона-переможця.

Околиця являє собою кілька нейронів, які оточують нейрона-переможця.

Спочатку до околиці належить велика кількість нейронів, далі її розмір поступово зменшується. Мережа формує топологічну структуру, у якій схожі приклади утворюють групи прикладів, що близько перебувають на топологічній карті.

Отриману карту можна використовувати як засіб візуалізації при аналізі даних. У результаті навчання карта Кохонена класифікує вхідні приклади на кластери (групи схожих прикладів) і візуально відображає багатомірні вхідні дані на площині нейронів.

Унікальність методу карт, що самоорганізуються, полягає в перетворенні *n*-вимірного простору в двовимірний. Застосування двовимірних сіток пов'язане з тим, що існує проблема відображення просторових структур більшої розмірності.

Маючи таке представлення даних, можна візуально визначити наявність або відсутність взаємозв'язку у вхідних даних.

Нейрони карти Кохонена розташовують у вигляді двомірної матриці, розфарбовують цю матрицю залежно від аналізованих параметрів нейронів.

На рис. 12.2 наведений приклад карти Кохонена.

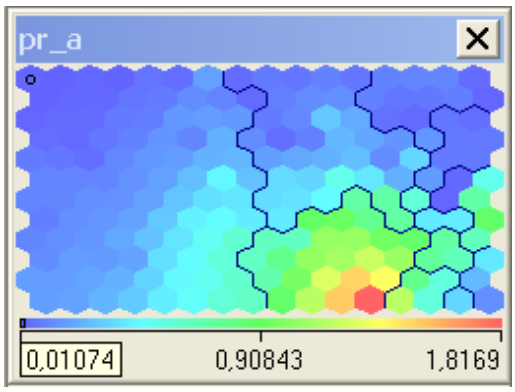


Рисунок 12.2 – Приклад карти Кохонена

Що ж означає її розфарбування? На рис. 12.3 наведене розфарбування карти, а точніше, її *i*-ої ознаки (показника pr_a), у тривимірному представленні. Як бачимо, темно-сині ділянки на карті відповідають найменшим значенням показника, червоні – найвищим.

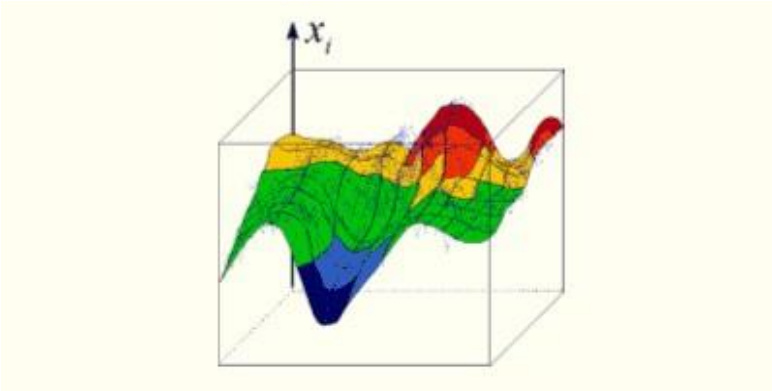


Рисунок 12.3 – Розфарбування *i*-ої ознаки в тривимірному просторі

Тепер, повертаючись до рисунку рис. 12.2, можна сказати, які об’єкти мають найбільші значення розглянутого показника (група об’єктів, позначена червоним кольором), а які – найменші значення (група об’єктів, позначена синім кольором).

Отже, карти Кохонена (як і географічні карти) можна відображати:

- у двомірному вигляді, тоді карта розфарбовується відповідно до рівня виходу нейрона;
- у тривимірному вигляді.

У результаті роботи алгоритму одержуємо такі карти:

- карта входів нейронів;
- карта виходів нейронів;
- спеціальні карти.

Координати кожної карти визначають положення одного нейрона. Так, координати визначають нейрон, який перебуває на перетинанні 15-го стовпця з 30-м поруч у матриці нейронів. Розглянемо, що ж являють собою ці карти.

4. Карта входів та виходів нейронів

Карта входів нейронів. Ваги нейронів підбудовуються під значення вхідних змінних і відображають їхню внутрішню структуру. Для кожного входу малюється своя карта, розфарбована у відповідності зі значенням конкретної ваги нейрона.

При аналізі даних використовують кілька карт входів.

На одній із карт виділяють область певного кольору – це означає, що відповідні вхідні приклади мають приблизно однакове значення відповідного входу. Колірний розподіл нейронів із цієї області аналізується на інших картах для визначення схожих або відмітних характеристик. Приклад розглянутих карт входів буде наведений нижче.

Карта виходів нейронів. На карту виходів нейронів проектується взаємне розташування досліджуваних вхідних даних. Нейрони з однаковими значеннями виходів утворюють кластери – замкнені області на карті, які включають нейрони з однаковими значеннями виходів.

Спеціальні карти. Це карта кластерів, матриця відстаней, матриця щільності потрапляння та інші карти, які характеризують кластери, отримані в результаті навчання мережі Кохонена.

Важливо розуміти, що між усіма розглянутими картами існує взаємозв'язок – усі вони є різними розфарбуваннями тих самих нейронів. Кожний приклад із навчальної вибірки має те саме розташування на всіх картах.

Приклад розв'яання задачі. Програмне забезпечення, що дозволяє працювати з картами Кохонена, зараз представлене великою кількістю інструментів. Це можуть бути як інструменти, що включають тільки реалізацію методу карт, що самоорганізуються, так і нейропакети з цілим набором структур нейронних мереж, серед яких – і карти Кохонена; також цей метод реалізований у деяких універсальних інструментах.

До інструментарію, що включає реалізацію методу карт Кохонена, відносяться Somine, Statistica, Neuroshell, Neuroscalp, Deductor і багато інших. Для розв'язання задачі будемо використовувати аналітичний пакет Deductor.

Нехай є база даних комерційних банків із показниками діяльності за поточний період. Необхідно провести їх кластеризацію, тобто виділити однорідні групи банків на основі показників із бази даних, усього показників – 21.

Вихідна таблиця перебуває у файлі «banks.xls». Вона містить показники діяльності комерційних банків за звітний період.

Спочатку імпортуємо дані з xls-файлу в середовище аналітичного пакета.

На першому кроці майстра запускаємо майстер обробки й вибираємо зі списку метод обробки «Карта Кохонена». Далі слід настроїти призначення стовпців, тобто для кожного стовпця вибрати одне із призначень: вхідне, вихідне, не використовується та інформаційне. Укажемо всім стовпцям, відповідним до показників діяльності банків, призначення «Вхідний». «Вихідний» не призначаємо.

Наступний крок пропонує розбити вихідну множину на навчальну, тестову й валідаційну. За замовчуванням, програма пропонує розбити множину на навчальну – 95% і тестову – 5%.

На кроці 5, зображеному на рис. 12.4 пропонується настроїти параметри карти: кількість гнізд по *X* і по *Y*, їх форму (шестикутну або чотирикутну).

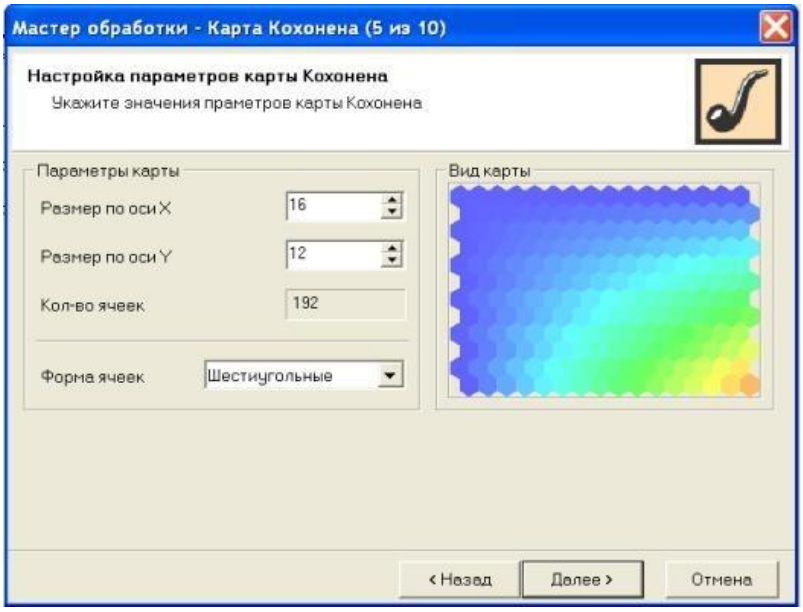


Рисунок 12.4 – Крок 5 «Налаштування параметрів карти Кохонена»

На шостому кроці «Налаштування параметрів зупинки навчання», проілюстрованому на рис. 12.5, установлюємо параметри зупинки навчання й установлюємо епоху, по досягненню якої навчання буде припинено.

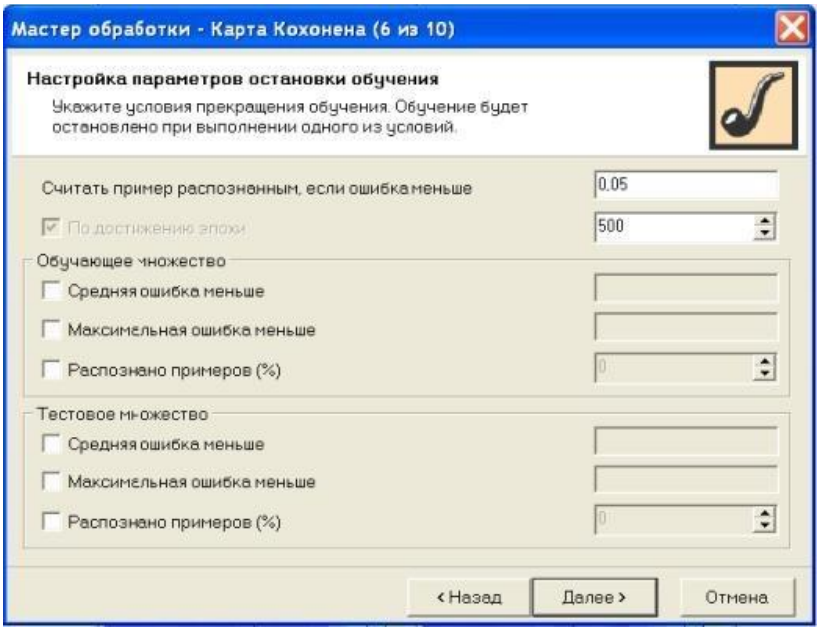


Рисунок 12.5 – Крок 6 «Налаштування параметрів зупинки навчання»

На сьомому кроці, представленому на рис. 12.6, настроюються інші параметри навчання: спосіб початкової ініціалізації, тип функції сусідства. Можливі два варіанти кластеризації: автоматичне визначення числа кластерів із відповідним рівнем значимості й фіксована кількість кластерів (визначається користувачем). Оскільки невідома кількість кластерів, виберемо автоматичне визначення їх кількості**.**

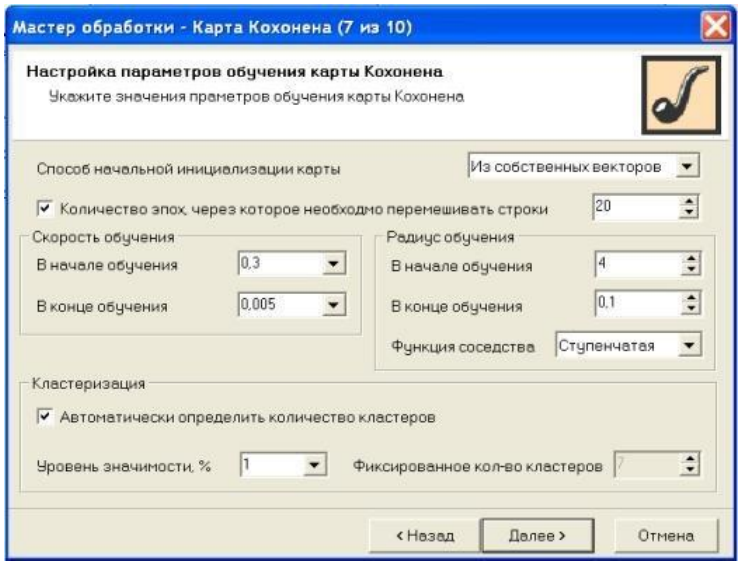


Рисунок 12.6 – Крок 7 «Налаштування параметрів зупинки навчання»

На восьмому кроці запускаємо процес навчання мережі – необхідно натиснути на кнопку «Пуск» і дочекатися закінчення процесу навчання. Під час навчання можемо спостерігати зміну кількості розпізнаних прикладів і поточні значення помилок.

По закінченню навчання в списку візуалізаторів виберемо «Карту Кохонена» і візуалізатор «Що-Якщо». На останньому кроці будуємо відображення карти Кохонена, цей крок проілюстрований на рис. 12.7.

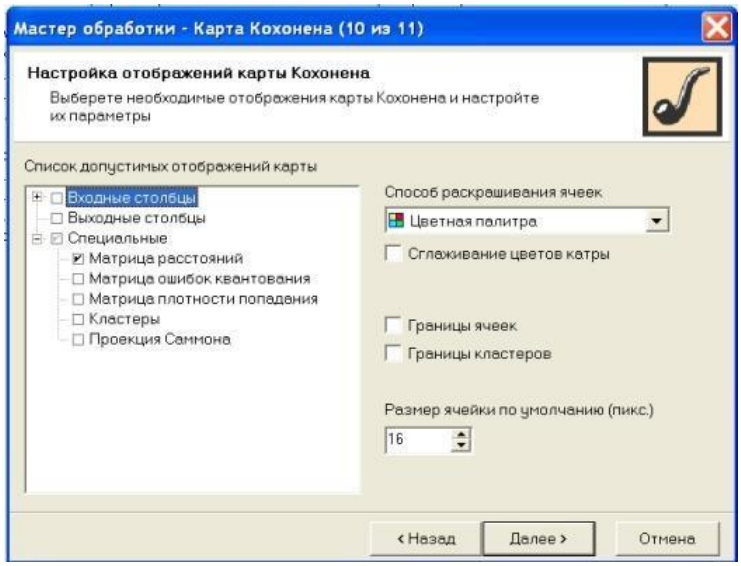


Рисунок 12.7 – Крок 10 «Налаштування відображень карти Кохонена»

Укажемо відображення всіх вхідних, вихідних стовпців, кластерів, а також поставимо прапорець «Границі кластерів» для чіткого відображення границь.

Карти входів. При аналізі карт входів рекомендують використовувати відразу кілька карт. Досліджуємо фрагмент карти, що складається з карт трьох входів, який наведений на рис. 12.8

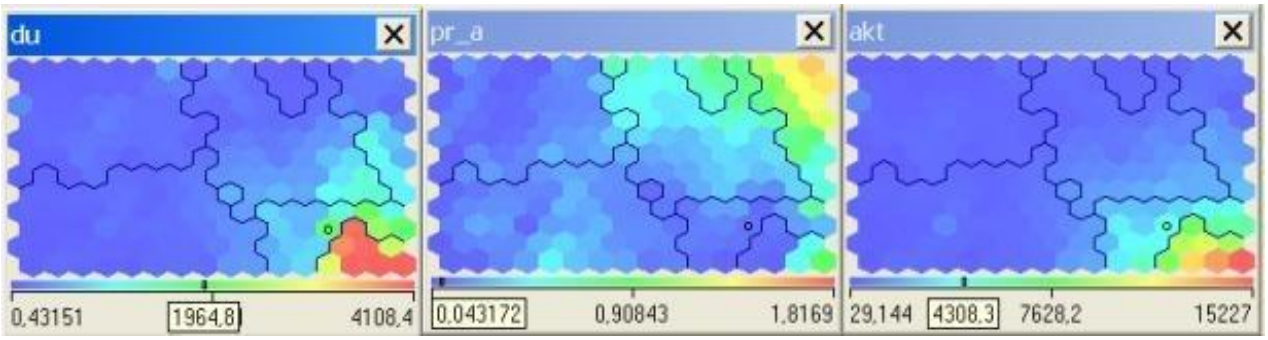


Рисунок 12.8 – Карти трьох входів

На одній із карт виділяємо область із найбільшими значеннями показника. Далі має сенс вивчити ці ж нейрони на інших картах.

На першій карті найбільші значення мають об’єкти, розташовані в правому нижньому куті. Розглядаючи одночасно три карти, ми можемо сказати, що ці ж об’єкти мають найбільші значення показника, зображеного на третій карті. Також по розфарбуванню першої й третьої карти можна зробити висновок, що існує взаємозв’язок між цими показниками.

Також ми можемо визначити, наприклад, таку характеристику: кластер, розташований у правому верхньому куті, характеризується низькими значеннями показників *du* (депозити юридичних осіб) і *akt* (активи банку) і високими значеннями показників *pr_a* (прибутковість активів).

Ця інформація дозволяє таким чином охарактеризувати кластер, що перебуває в правому верхньому куті: це банки з невеликими активами, невеликими притягнутими депозитними коштами від юридичних осіб, але з найбільш прибутковими активами, тобто це група невеликих, але найбільш прибуткових банків.

Це лише фрагмент висновку, який можна зробити, досліджуючи карту.

На рисунку 12.9 наведена ілюстрація карт входів і виходів, остання – це карта кластерів. Тут бачимо кілька карт входів (показників діяльності банків) і сформовані кластери, кожний з яких виділений окремим кольором.

Для знаходження конкретного об’єкта на карті необхідно натиснути правою кнопкою миші на досліджуваному об’єкті й вибрати пункт «Знайти комірку на карті». Виконання цієї процедури показане на рис. 12.10. У результаті можемо бачити як сам об’єкт, так і значення того виміру, який переглядаємо. Таким чином, можна оцінити положення аналізованого об’єкта, а також порівняти його з іншими об’єктами.

У результаті застосування карт, що самоорганізуються, багатовимірний простір вхідних факторів був представлений в двовимірному вигляді, в якому його достатньо зручно аналізувати.

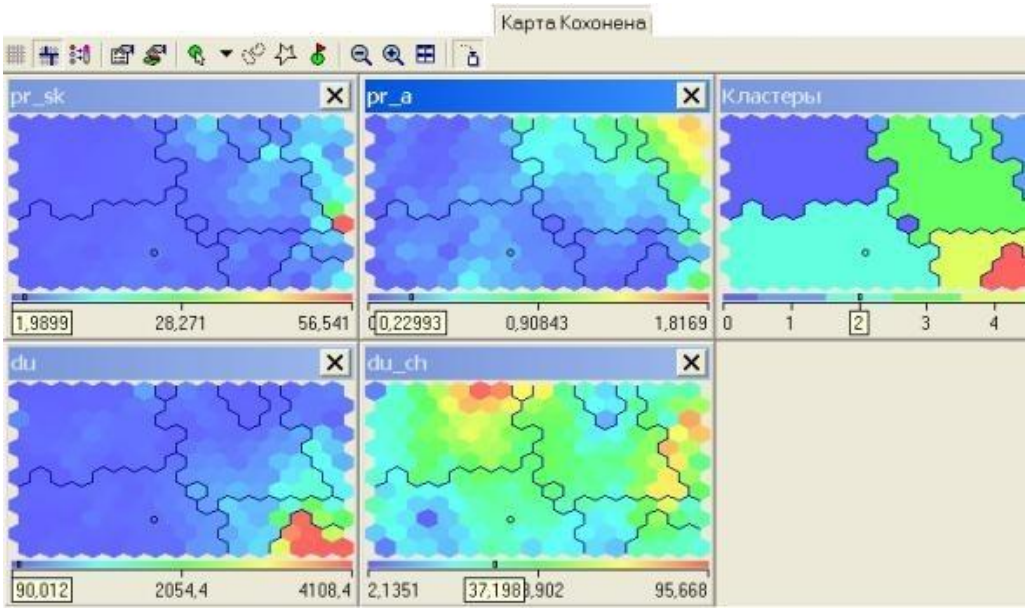


Рисунок 12.9 – Карти входів і виходів

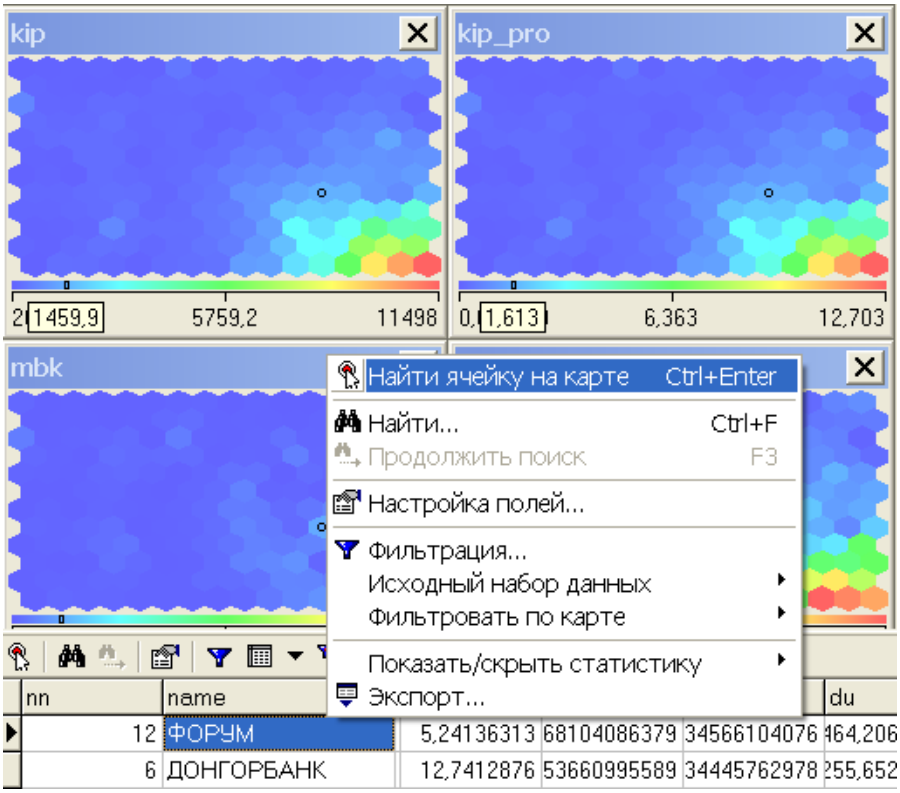


Рисунок 12.10 – Комірка на карті

Банки були класифіковані на 7 груп, для кожної з яких можливе визначення конкретних характеристик, виходячи з розфарбування відповідних показників.

5. Що таке асоціативні правила?

Асоціація – одна із задач Data Mining. Метою пошуку асоціативних правил (association rule) є знаходження закономірностей між зв’язаними подіями в базах даних.

Дуже часто покупці купують не один товар, а декілька. У більшості випадків між цими товарами існує взаємозв'язок. Так, наприклад, покупець, що купує ноутбук, швидше за все, захоче придбати також сумку. Ця інформація може бути використана для розміщення товару на прилавках.

Асоціативні правила, часто знаходять застосування:

- аналіз Web-блогів;
- у роздрібній торгівлі: визначення товарів, які варто просувати спільно; вибір місця розташування товару в магазині; аналіз споживчого кошика; прогнозування попиту;
- перехресні продажі: якщо є інформація про те, що клієнти придбали продукти А, Б і В, то які з них найімовірніше куплять продукт Г;
- маркетинг: пошук ринкових сегментів, тенденцій купівельної поведінки;
- сегментація клієнтів: виявлення загальних характеристик клієнтів компанії, виявлення груп покупців;
- оформлення каталогів, аналіз збутових кампаній фірми, визначення послідовностей покупок клієнтів (яка покупка піде за покупкою товару А);

Наведемо простий приклад асоціативного правила: покупець, що купує ноутбук, придбає до нього мишку з імовірністю 50%.

Введення в асоціативні правила. Уперше задача пошуку асоціативних правил (association rule mining) була запропонована для знаходження типових шаблонів покупок, здійснених у супермаркетах, тому іноді її ще називають аналізом ринкового кошика (market basket analysis).

Ринковий кошик – це набір товарів, придбаних покупцем у рамках однієї окремо взятої транзакції.

Транзакції є досить характерними операціями, ними, наприклад, можуть описуватися результати відвідувань різних магазинів.

Транзакція – це множина подій, які відбулися одночасно.

Реєструючи всі бізнес-операції протягом усього часу своєї діяльності, торговельні компанії накопичують величезні кількості транзакцій. Кожна така транзакція являє собою набір товарів, куплених покупцем за один візит.

Отримані в результаті аналізу шаблони включають перелік товарів і число транзакцій, які містять дані набори.

Транзакційна або операційна база даних (Transaction database) являє собою двовимірну таблицю, яка складається з номера транзакції (TID) і переліку покупок, придбаних під час цієї транзакції.

TID – унікальний ідентифікатор, що визначає кожну угоду або транзакцію.

Приклад транзакційної бази даних, що складається з купівельних транзакцій, наведено в таблиці 12.1. У таблиці перший стовпчик (TID) визначає номер транзакції, у другому стовпчику таблиці наведені товари, придбані під час певної транзакції**.**

На основі наявної бази даних нам потрібно знайти закономірності між подіями, тобто покупками.

Шаблони, що часто зустрічаються, або зразки. Допустимо, є транзакційна база даних D.

Таблиця 12.1 − Транзакційна база даних

TID	Покупки
100	Карта пам’яті, DVD-диск, USB-подовжувач
200	DVD-диск, USB-подовжувач, WEB-камера
300	Комп’ютерна миша, DVD-диск, WEB-камера
400	USB-подовжувач, DVD-диск, Карта пам’яті, Комп’ютерна миша
500	DVD-диск, Карта пам’яті
600	VGA-кабель

Присвоємо значенням товарів змінні (таблиця 12.2).
Карта пам’яті = *a*
DVD-диск = *b*
USB-подовжувач = *c*
Комп’ютерна миша = *d*
WEB-камера = *e*
VGA-кабель = *f*

Таблиця 12.2 − Набори товарів, що часто зустрічаються

TID	Покупки	TID	Покупки
100	Флешка, DVD-диск, USB-подовжувач	100	*a, b, c*
200	DVD-диск, USB-подовжувач, WEB-камера	200	*b, c, e*
300	Комп’ютерна миша, DVD-диск, WEB-камера	300	*d, b, e*
400	USB- подовжувач, DVD-диск, Флешка, Комп’ютерна миша	400	*c, b, a, d*
500	DVD-диск, Флешка, USB-подовжувач	500	*b, a, c*
600	VGA-кабель	600	*f*

Розглянемо набір товарів (Itemset), що включає, наприклад, (флешка, DVD-диск, USB-подовжувач). Виразимо цей набір за допомогою змінних:

$$abc=\{a,b,c\}$$

Підтримка. Цей набір товарів зустрічається в нашій базі даних три рази, тобто підтримка цього набору товарів рівна 3:

$$SUP(abc)=3.$$

При мінімальному рівні підтримки, рівному трьом, набір товарів *abc* є шаблоном, що часто зустрічається.

min_sup=3, {Карта пам’яті, DVD-диск, USB-подовжувач} – частий шаблон, що зустрічається.

Підтримкою називають кількість або відсоток транзакцій, що містять певний набір даних.

Для даного набору товарів підтримка, виражена у відсотковому відношенні, рівна 50%.

SUP(*abc*)=(3/6)*100%=50%

Підтримку іноді також називають забезпеченням набору.

Отже, набір становить інтерес, якщо його підтримка вище заданого користувачем мінімального значення (min support). Ці набори називають такими, що часто зустрічаються (frequent**).**

6. Алгоритми пошуку асоціативних правил

Характеристики асоціативних правил. Асоціативне правило має вигляд: «з події А випливає подія В». У результаті такого виду аналізу встановлюємо закономірність такого виду: «Якщо в транзакції зустрівся набір товарів (або набір елементів) А, то можна зробити висновок, що в цій же транзакції повинен з'явитися набір елементів В)». Встановлення таких закономірностей дає можливість знаходити дуже прості й зрозумілі правила, які називають асоціативними.

Основними характеристиками асоціативного правила є підтримка й вірогідність правила.

Розглянемо правило «з покупки флешки випливає покупка USB-подовжувача» для бази даних, яка була наведена вище в таблиці 14.1. Поняття підтримки набору вже розглянуто. Існує поняття підтримки правила.

Правило має підтримку *S*, якщо *s*% транзакцій із усього набору містять одночасно набори елементів А і В або, інакше кажучи, містять обидва товари.

Флешка – це товар А, USB-подовжувач – це товар В. Підтримка правила «з покупки флешки випливає покупка USB-подовжувача» рівна 3, або 50%.

Вірогідність правила показує, яка ймовірність того, що з події А випливає подія В.

Правило «з А випливає В» справедливе з вірогідністю *C*, якщо *c*% транзакцій з усієї множини, що містить набір елементів А, також містять набір елементів В.

Якщо кількість транзакцій, що містять USB-подовжувач, рівне чотирьом, а кількість транзакцій, що містять також і флешку, рівне трьом, то вірогідність правила рівна (3/4)*100%, тобто 75%.

Вірогідність правила «з покупки USB-подовжувача випливає покупка флешки» рівна 75%, тобто 75% транзакцій, що містять товар А, також містять товар В.

Границі підтримки й вірогідності асоціативного правила. За допомогою використання алгоритмів пошуку асоціативних правил аналітик може одержати всі можливі правила вигляду «з А випливає В», з різними значеннями підтримки й вірогідності. Однак у більшості випадків, кількість правил необхідно обмежувати заздалегідь установленими мінімальними й максимальними значеннями підтримки й вірогідності.

Якщо значення підтримки правила занадто велике, то в результаті роботи алгоритму будуть знайдені правила очевидні й добре відомі. Занадто низьке значення підтримки призведе до знаходження дуже великої кількості правил, які, можливо, будуть у більшості необґрунтованими, але не відомими й не

очевидними для аналітика. Отже, необхідно визначити такий інтервал («золоту середину»), який з одного боку забезпечить знаходження неочевидних правил, а з іншого – їх обґрунтованість.

Якщо рівень вірогідності занадто малий, то цінність правила викликає серйозні сумніви. Наприклад, правило з вірогідністю в 3% тільки умовно можна назвати правилом.

7. Методи пошуку асоціативних правил

Алгоритм AIS. Перший алгоритм пошуку асоціативних правил, що називався AIS, (запропонований Agrawal, Imielinski and Swami) був розроблений співробітниками дослідного центру IBM Almaden у 1993 році. Із цієї роботи виник інтерес до асоціативних правил; на середину 90-х років минулого століття припадає пік дослідницьких робіт у цій області, і з того часу щороку з'являється кілька нових алгоритмів.

В алгоритмі AIS кандидати множини наборів генеруються й підраховуються «на льоту», під час сканування бази даних.

Алгоритм SETM. Створення цього алгоритму було мотивовано бажанням використовувати мову SQL для обчислення наборів товарів, що часто зустрічаються. Як і алгоритм AIS, SETM також формує кандидатів «на льоту», ґрунтуючись на перетвореннях бази даних. Щоб використовувати стандартну операцію об'єднання мови SQL для формування кандидата, SETM відокремлює формування кандидата від їхнього підрахунку.

Незручність алгоритмів AIS і SETM – надмірне генерування й підрахунок занадто багатьох кандидатів, які в результаті не є такими, що часто зустрічаються. Для поліпшення їх роботи був запропонований алгоритм Apriori.

Робота даного алгоритму складається з декількох етапів, кожний з етапів складається з таких кроків:

- формування кандидатів;
- підрахунок кандидатів.

Формування кандидатів (candidate generation) – етап, на якому алгоритм, скануючи базу даних, створює множину *i*-елементних кандидатів (*i* – номер етапу). На цьому етапі підтримка кандидатів не розраховується.

Підрахунок кандидатів (candidate counting) – етап, на якому обчислюється підтримка кожного *i*-елементного кандидата. Тут же здійснюється відсікання кандидатів, підтримка яких менша мінімуму, установленого користувачем (min_sup).

Решту *i*-елементних наборів називаємо такими, що часто зустрічаються.

Розглянемо роботу алгоритму Apriori на прикладі бази даних D. Ілюстрація роботи алгоритму наведена на рис. 12.11. Мінімальний рівень підтримки рівний 3.

На першому етапі відбувається формування одноелементних кандидатів. Далі алгоритм підраховує підтримку одноелементних наборів. Набори з рівнем підтримки менше встановленого, тобто 3, відкидаються. У цьому прикладі це набори *e* і *f*, які мають підтримку, рівну 1. набори товарів, що залишилися,

вважаються одноелементними наборами, що часто зустрічаються, товарів – це набори *a, b, c, d.*

Далі відбувається формування двоелементних кандидатів, підрахунок їх підтримки й відсікання наборів з рівнем підтримки, меншим 3. Двоелементні набори товарів, що залишилися, вважаються двоелементними наборами, що часто зустрічаються, *ab*, *ac*, *bd*, беруть участь у подальшій роботі алгоритму.

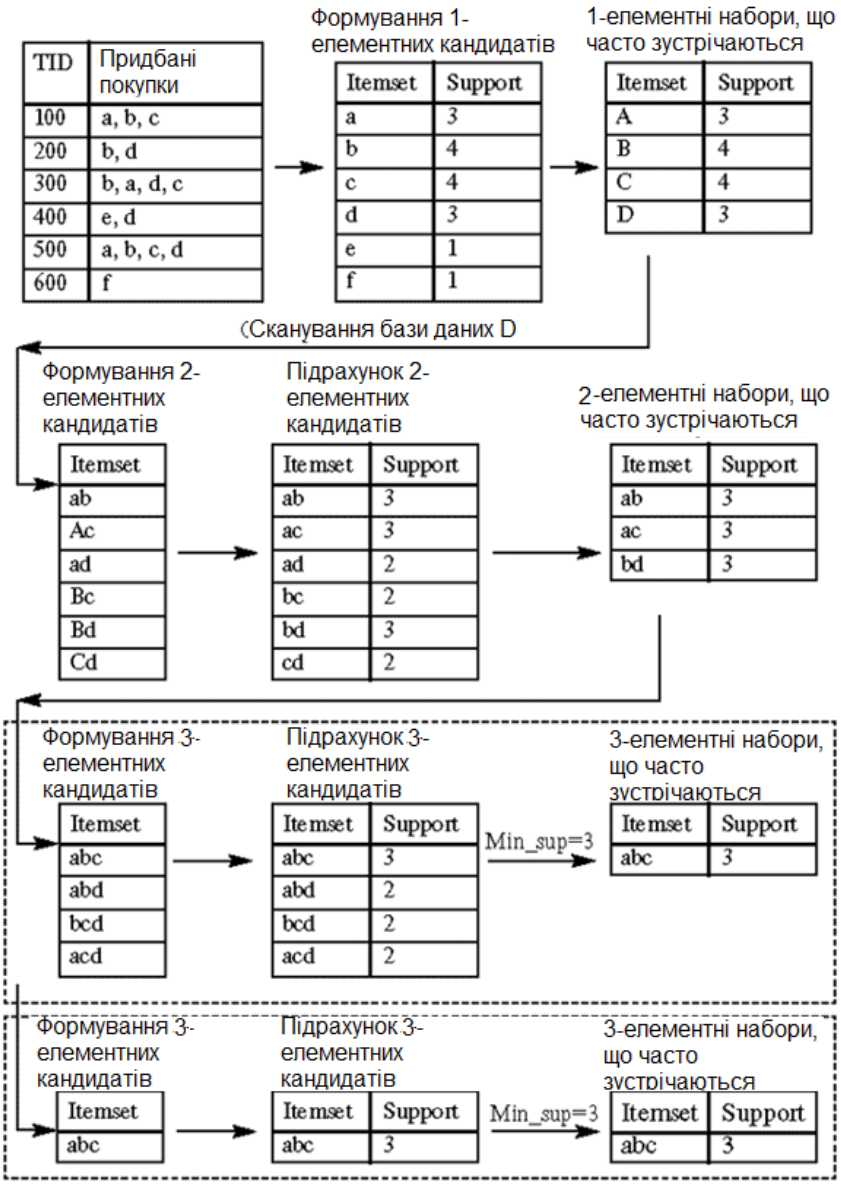


Рисунок 12.11 – Алгоритм Apriori

Якщо дивитися на роботу алгоритму прямолінійно, на останньому етапі алгоритм формує трьохелементні набори товарів: *abc, abd, bcd, acd*, підраховує їхню підтримку й відтинає набори з рівнем підтримки, меншим 3. Набір товарів *abc* може бути названий таким, що часто зустрічається.

Однак алгоритм Apriori зменшує кількість кандидатів, відсікаючи – aprior – тих, які свідомо не можуть стати такими, що часто зустрічаються, на основі інформації про відсічених кандидатів на попередніх етапах роботи алгоритму.

Відсікання кандидатів відбувається на основі припущення про те, що в наборі, що часто зустрічається, товарів усі підмножини повинні бути такими, що часто зустрічаються. Якщо в наборі наявна підмножина, яку на попередньому етапі було визначено такою, що нечасто зустрічається, цей кандидат уже не включається у формування й підрахунок кандидатів.

Так набори товарів *ad, bc, cd* були відкинуті як такі, що нечасто зустрічаються, алгоритм не розглядав товарів *abd, bcd, acd*.

При розгляді цих наборів формування трьохелементних кандидатів відбувалося б за схемою, наведеною у верхньому пунктирному прямокутнику. Оскільки алгоритм апріорі відкинув набори, що свідомо нечасто зустрічаються, останній етап алгоритму відразу визначив набір *abc* як єдиний триелементний набір, що часто зустрічається (етап наведений у нижньому пунктирному прямокутнику).

Алгоритм Apriori розраховує також підтримку наборів, які не можуть бути відсічені апріорі. Це так звана негативна область (negative border), до неї належать набори-кандидати, які зустрічаються рідко, їх самих не можна віднести до таких, що часто зустрічаються, але всі підмножини даних наборів є такими, що часто зустрічаються.

Різновиди алгоритму Apriori. Залежно від розміру найдовшого набору, що часто зустрічається, алгоритм Apriori сканує базу даних певну кількість разів. Різновиди алгоритму Apriori, що є його оптимізацією, запропоновані для скорочення кількості сканувань бази даних, кількості наборів-кандидатів або того й іншого. Були запропоновані такі різновиди алгоритму Apriori: Aprioritid і Apriorihybrid.

Aprioritid. Цікава особливість цього алгоритму – те, що база даних D не використовується для підрахунку підтримки кандидатів набору товарів після першого проходу.

Із цією метою використовується кодування кандидатів, виконане на попередніх проходах. У наступних проходах розмір закодованих наборів може бути набагато меншим, ніж база даних, і в такий спосіб заощаджуються значні ресурси.

Apriorihybrid. Аналіз часу роботи алгоритмів Apriori і Aprioritid показує, що в більш ранніх проходах Apriori досягає більшого успіху, ніж Aprioritid; однак Aprioritid працює краще Apriori у більш пізніх проходах. Крім того, вони використовують ту саму процедуру формування наборів-кандидатів. Заснований на цьому спостереженні алгоритм Apriorihybrid запропонований, щоб об’єднати кращі властивості алгоритмів Apriori і Aprioritid. Apriorihybrid використовує алгоритм Apriori у початкових проходах і переходить до алгоритму Aprioritid, коли очікується, що закодований набір первісної множини наприкінці проходу буде відповідати можливостям пам’яті. Однак перемикання від Apriori до Aprioritid вимагає залучення додаткових ресурсів.

Один із них – алгоритм DHP, або алгоритм хешування (J. Park, M. Chen and P. Yu, 1995 рік). В основі його роботи – імовірнісний підрахунок наборів-кандидатів, що здійснюється для скорочення кількості підрахованих кандидатів на кожному етапі виконання алгоритму Apriori. Скорочення забезпечується за

рахунок того, що кожний з *k*-елементних наборів-кандидатів крім кроку скорочення проходить крок хешування. В алгоритмі на *k*-1 етапі під час вибору кандидата створюється так звана хеш-таблиця. Кожний запис хеш-таблиці є лічильником усіх підтримок *k*-елементних наборів, які відповідають цьому запису в хеш-таблиці. Алгоритм використовує цю інформацію на етапі *k* для скорочення множини *k*-елементних наборів-кандидатів. Після скорочення підмножини, як це відбувається в Apriori, алгоритм може вилучити набір-кандидат, якщо його значення в хеш-таблиці менше граничного значення, установленого для забезпечення.

До інших удосконалених алгоритмів відносяться: PARTITION, DIC, алгоритм «вибіркового аналізу».

PARTITION алгоритм (A. Savasere, E. Omiecinski and S. Navathe, 1995 рік). Цей алгоритм розбивки (поділу) полягає в скануванні транзакційної бази даних шляхом поділу її на розділи, які не перетинаються, кожний з яких може вміститися в оперативній пам'яті. На першому кроці в кожному з розділів за допомогою алгоритму Apriori визначаються «локальні» набори даних, що часто зустрічаються. На другому підраховується підтримка кожного такого набору щодо всієї бази даних. Отже, на другому етапі визначається множина усіх потенційних наборів даних, що зустрічаються.

Алгоритм DIC, Dynamic Itemset Counting (S. Brin R. Motwani, J. Ullman and S. Tsur, 1997 рік). Алгоритм розбиває базу даних на кілька блоків, кожний з яких відзначається так званими «початковими точками» (start point), і потім циклічно сканує базу даних.

Питання для самоконтролю

1. Поясніть як відбувається процес навчання з вчителем нейронної мережі?
2. Як відбувається підготовка даних для навчання?
3. Дайте визначення поняттю карти Кохонена?
4. Для розв'язання яких задач можна застосовувати карти Кохонена?
5. Для розв'язання яких задач можна застосовувати інструмент Data Mining асоціативні правила?
6. Які існують методи пошуку асоціативних правил?

КОНТРОЛЬНІ ПИТАННЯ ДО РОЗДІЛУ ІІ

1. Яка зі стадій Data mining може вважатися додатковою, невід'ємною?
2. Що таке якість даних?
3. Які цілі підготовки даних до аналізу? Які завдання підготовки даних?
4. Розкрийте сутність ієрархічних і ітеративних методів кластеризації.
5. Зазначте особливості кластеризації в якісних і кількісних шкалах.
6. Опишіть метод нечіткої кластеризації fuzzy c-means.
7. Як оцінити якість побудованої моделі для задачі кластеризації?
8. Що таке чітка і нечітка кластеризація?
9. Які є підходи до розрахунку відстані між кластерами?
10. Розкрийте сутність методу кластеризації *k*-середніх.
11. Розкрийте сутність та особливості прогнозування часових рядів.
12. У чому полягає завдання пошуку асоціативних правил? Наведіть практичний приклад.
13. Що таке сильне асоціативне правило?
14. Із яких двох кроків складається пошук асоціативних правил?
15. У чому полягає принцип Apriori?
16. Як формуються правила із знайдених частих наборів?
17. Опишіть алгоритм Apriori.
18. Що означають параметри support, confidence, lift, conviction, які застосовуються в алгоритмі Apriori?
19. Опишіть візуальний аналіз даних (Visual Mining), а саме його етапи, переваги і недоліки.
20. Надайте характеристику засобів візуалізації за типами даних, інструментами візуалізації.

РЕКОМЕНДОВАНА ЛІТЕРАТУРА

Основна:

1. Шумейко А. А. Интеллектуальный анализ данных (Введение в Data Mining). Днепропетровск : Белая Е. А., 2015. 212 с.
2. Бахрушин В. Є. Методи аналізу даних : навч. посіб. Запоріжжя : КПУ, 2011. 268 с.
3. Гладій Г. М. Інтелектуальний аналіз даних. Тернопіль : ТНЕУ, 2014. 54 с.
4. Барсегян А. А., Куприянов М. С., Степаненко В. В., Холод И. И. Технологии анализа данных. Data Mining, Visual Mining, Text Mining, OLAP. Санкт-Петербург : БХВ-Петербург, 2007. 324 с.
5. Джулій В. М., Горбатюк О. М. Структура, функції системи інтелектуальної обробки даних. *Вимірювальна та обчислювальна техніка в технологічних процесах*. Хмельницький, 2014. № 3. С. 106–110.
6. Галузинський Г. П., Гордієнко І. В. Перспективні технологічні засоби оброблення інформації : навч.-метод. посіб. для самост. вивчення дисципліни. Київ : КНЕУ, 2002. 279 с.
7. Квєтний Р. Н., Кислиця Л. М., Коцюбинський В. Ю., Усов В. В. Інформаційна технологія прийняття рішень на основі прогнозування часових рядів з подвійною довгою пам’яттю : монографія. Вінниця : ВНТУ, 2012. 140 с.

Додаткова:

1. Ситник В. Ф. Системи підтримки прийняття рішень: навч. посіб. Київ : КНЕУ, 2004. 628 с.
2. Введение в OLAP: часть 1. Основы OLAP. URL : http://www.olap.ru/basic/OLAP_intro1.asp (дата звернення: 29.04.2020).
3. Братушка С. М., Новак С. М., Хайлук С. О. Системи підтримки прийняття рішень: навч. посіб. Суми : ДВНЗ «УАБС НБУ», 2010. 265 с.
4. Путренко В. В. Системні основи інтелектуального аналізу геопросторових даних. *System Research & Information Technologies*. Київ, 2015. № 3. С. 20–33.
5. Бідюк П. І., Савченко С. М., Савченко А. С. Методи інтелектуального аналізу даних у прогнозуванні конкурентоспроможності підприємств. *Підприємництво та інновації*. Київ, 2018. № 5. С. 7–16.

Електронні ресурси:

1. Data Mining від Oracle: сьогодення і майбутнє. URL : http://citforum.ru/database/oracle/data_mining_solutions (дата звернення: 29.04.2020).
2. Data Mining: Інформація. URL : http://www.intuit.ru/department/ database/datamining (дата звернення: 29.04.2020).
3. ViDa stands for Visualization of Data. URL: http://bioinfo-out.curie.fr/projects/vidaexpert (дата звернення: 29.04.2020).

ВИКОРИСТАНА ЛІТЕРАТУРА:

1. Черняк О.І., Захарченко П. В. Інтелектуальний аналіз даних: підручник. Київ : Знання, 2014. 599 с.
2. Ситник В. Ф., Краснюк М. Т. Інтелектуальний аналіз даних (дейтамайнінг) : навч. посіб. Київ : КНЕУ, 2007. 376 с.
3. Олійник А. О., Субботін С. О., Олійник О. О. Інтелектуальний аналіз даних : навч. посіб. Запоріжжя : ЗНТУ, 2012. 278 с.
4. Марченко О. О., Россада Т. В. Актуальні проблеми Data Mining : навч. посіб. Київ, 2017. 150 с.
5. Ланде Д. В., Субач І. Ю., Бояринова Ю. Є. Основи теорії і практики інтелектуального аналізу даних у сфері кібербезпеки: навч. посіб. Київ : ІСЗЗІ КПІ ім. Ігоря Сікорського, 2018. 297 с.

Навчальне видання
(українською мовою)

Сергій Миколайович Іванов
Наталія Костянтинівна Максишко
Дмитро Олексійович Бречко

ІНТЕЛЕКТУАЛЬНИЙ АНАЛІЗ ДАНИХ

конспект лекцій для здобувачів ступеня вищої освіти бакалавра спеціальності «Економіка» освітньо-професійної програми «Економічна кібернетика»

Рецензент Г.Ю. Кучерова
Відповідальний за випуск *Н.К. Максишко*
Коректор *В.В. Рянічева*