МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
МИКОЛАЇВСЬКИЙ НАЦІОНАЛЬНИЙ АГРАРНИЙ УНІВЕРСИТЕТ
ФАКУЛЬТЕТ МЕНЕДЖМЕНТУ

Кафедра економічної кібернетики,
комп'ютерних наук та інформаційних технологій

ІНТЕЛЕКТУАЛЬНИЙ АНАЛІЗ ДАНИХ

Конспект лекцій
для здобувачів першого (бакалаврського) рівня вищої освіти
ОПП «Комп'ютерні науки» спеціальності F3 (122) «Комп'ютерні науки»
денної форми здобуття вищої освіти

Миколаїв – 2025

УДК 004.6
І-73

Друкується за рішенням науково-методичної комісії факультету менеджменту Миколаївського національного аграрного університету
(протокол №1 від 28 серпня 2025 року)

Укладачі:

О. Ю. Пархоменко – к.ф.-м.н., доцент, доцент кафедри економічної кібернетики, комп'ютерних наук та інформаційних технологій Миколаївського національного аграрного університету;

С. І. Тищенко – к.п.н., доцент, доцент кафедри економічної кібернетики, комп'ютерних наук та інформаційних технологій Миколаївського національного аграрного університету;

С. І. Ємельянов – PhD, старший викладач кафедри економічної кібернетики, комп'ютерних наук та інформаційних технологій Миколаївського національного аграрного університету;

О. О. Жебко – асистент кафедри економічної кібернетики, комп'ютерних наук та інформаційних технологій Миколаївського національного аграрного університету;

О. Є. Богатєнкова – асистент кафедри економічної кібернетики, комп'ютерних наук та інформаційних технологій Миколаївського національного аграрного університету.

Рецензенти:

В.В.Базаренко – заступник начальника Миколаївської обласної військової адміністрації з питань цифрового розвитку, цифрових трансформацій і цифровізації (CDTO);

Д.Л.Кошкін – к.т.н., доцент, доцент кафедри електроенергетики, електротехніки та електромеханіки Миколаївського національного аграрного університету.

Інтелектуальний аналіз даних : конспект лекцій для здобувачів першого (бакалаврського) рівня вищої освіти ОПП «Комп'ютерні науки» спеціальності F3 (122) «Комп'ютерні науки» денної форми здобуття вищої освіти / уклад. О. Ю. Пархоменко, С. І. Тищенко, С. І. Ємельянов, О. О. Жебко, О. Є. Богатєнкова . Миколаїв: МНАУ, 2025. 70 с.

І-73

УДК 004.6

ЗМІСТ

ПЕРЕДМОВА

Сучасний світ неможливо уявити без стрімкого зростання обсягів інформації. Кожну секунду генеруються терабайти даних: транзакції в банках, історії покупок в інтернет-магазинах, показники датчиків, пости в соціальних мережах, медичні картки пацієнтів. Однак самі по собі дані – це лише сировина. Справжню цінність вони набувають лише тоді, коли ми вміємо з них вилучати приховані знання, виявляти неочевидні закономірності та будувати прогнози. Саме цьому присвячений пропонований навчальний посібник з дисципліни «Інтелектуальний аналіз даних».

Сьогодні Data Mining перестав бути екзотичною наукою для вузького кола фахівців – це невід'ємна складова діяльності сучасного IT-спеціаліста. Володіння методами інтелектуального аналізу даних дозволяє не лише обробляти великі масиви інформації, але й приймати обґрунтовані рішення в бізнесі, науці, медицині, маркетингу та багатьох інших сферах. Саме формуванню цих прикладних компетенцій присвячений цей посібник, який ви тримаєте в руках.

Навчальний посібник є логічним продовженням теоретичного курсу з інтелектуального аналізу даних і містить п'ятнадцять лекцій, що охоплюють ключові аспекти діяльності сучасного аналітика даних. Структура посібника побудована за принципом «від простого до складного» – від знайомства з базовими поняттями до створення складних прогностичних моделей та нейронних мереж. Такий підхід дозволяє студентам поступово занурюватися в професію, закріплюючи теоретичні знання через розбір конкретних прикладів.

Перші лекції закладають фундамент, необхідний для подальшої роботи. Ви ознайомитеся з основними задачами Data Mining, опануєте потужне середовище MatLab, навчитеся готувати дані до аналізу – нормалізувати, стандартизувати, очищати від шумів та кодувати категоріальні ознаки. Окрема увага приділяється первинному статистичному аналізу, який дозволяє глибоко зрозуміти структуру даних ще до застосування складних алгоритмів.

Наступний блок лекцій присвячено фундаментальним методам навчання без учителя – кластеризації. Ви детально вивчите ієрархічні алгоритми, навчитеся обирати оптимальні міри близькості та методи зв'язку кластерів. Окремо розглядаються найпопулярніші алгоритми квадратичної похибки – чіткий k-means та нечіткий c-means, які ви навчитеся застосовувати в середовищі MatLab.

Значну увагу в посібнику приділено задачам класифікації та прогнозування. Ви познайомитеся з широким спектром методів: від класичного дискримінантного аналізу до сучасних підходів – дерев рішень, випадкового лісу, методу опорних векторів та наївного баєсівського класифікатора. Кожен метод супроводжується практичними прикладами реалізації в MatLab та SPSS.

Особливе місце посідають завдання з пошуку асоціативних правил, що дозволяють виявляти приховані закономірності в транзакційних даних. Ви детально розберете роботу класичного алгоритму Apriori та навчитеся

оцінювати знайдені правила за допомогою підтримки, достовірності та інших метрик.

Завершальні лекції знайомлять із найпотужнішими інструментами сучасного аналізу – нейронними мережами та методами зменшення розмірності. Ви не лише зрозумієте, як влаштований штучний нейрон, але й навчитеся створювати, навчати та використовувати багатошарові мережі для розв'язання задач класифікації та прогнозування часових рядів. Окремо розглядаються факторний аналіз та метод головних компонент, які дозволяють виявити приховані (латентні) змінні та зменшити розмірність простору ознак.

Кожна лекція містить чітко сформульований теоретичний матеріал, численні приклади розв'язання задач, а також питання для обговорення, які дозволяють студентам перевірити глибину засвоєння матеріалу. Така структура дозволяє використовувати посібник як для аудиторних занять під керівництвом викладача, так і для самостійного опанування дисципліни.

Сподіваємось, що цей посібник стане для вас надійним провідником у світ інтелектуального аналізу даних, допоможе здобути необхідні теоретичні знання та практичні навички, які дозволять вам впевнено почуватися на реальних проектах. Пам'ятайте: аналіз даних – це не просто робота з числами, це мистецтво знаходити істину в хаосі інформації, і саме ви творите світ, у якому технології допомагають людям приймати правильні рішення.

Бажаємо успіхів і цікавих відкриттів!

Лекція 1
Вступ до інтелектуального аналізу даних

Передумови виникнення Data Mining

Сучасне суспільство характеризується стрімким зростанням обсягів інформації, що зберігається в цифровому форматі. Це явище, яке часто називають «інформаційним вибухом», стало потужним каталізатором для розвитку нових підходів до роботи з даними. Повсякденна діяльність підприємств, наукових установ, державних органів та окремих осіб супроводжується накопиченням величезних масивів структурованих, слабкоструктурованих і навіть зовсім неструктурованих даних різноманітних форматів. Це можуть бути бази даних транзакцій, текстові документи, геопросторові дані, мультимедійна інформація, показники сенсорів та багато іншого.

Паралельно з процесом накопичення даних відбувається постійне вдосконалення апаратного забезпечення, що дозволяє зберігати дедалі більші обсяги інформації за доступну ціну, а також програмних засобів і технологій для її передачі та обробки. Виникла парадоксальна ситуація: людство навчилося ефективно збирати та зберігати дані, але виявилося неспроможним повною мірою використовувати прихований у них потенціал. Традиційні методи статистичного аналізу та управління базами даних виявилися недостатньо ефективними для виявлення складних, неочевидних закономірностей у величезних масивах інформації. Саме ця потреба в перетворенні «сирих» даних на цінне знання, придатне для прийняття обґрунтованих рішень, призвела до виникнення та бурхливого розвитку нового міждисциплінарного напряму – інтелектуального аналізу даних, або Data Mining.

Основні поняття та визначення

Термін «Data Mining» не має однозначного перекладу українською мовою. Дослівно він перекладається як «видобуток даних» або «розкопка даних», що дуже влучно відображає суть процесу. Інші варіанти перекладу, що зустрічаються у фаховій літературі, – це «розвідка даних», «глибинний аналіз даних», «просіювання інформації», «вилучення даних» та найбільш поширений сьогодні – «інтелектуальний аналіз даних». У цьому курсі ми будемо використовувати як україномовний термін «інтелектуальний аналіз даних», так і його англомовний відповідник Data Mining, розуміючи їх як синоніми.

В основі інтелектуального аналізу даних лежить концепція шаблонів. Шаблон (pattern) – це закономірність, яка є характерною для певного набору даних. Ця закономірність має бути неочевидною, нетривіальною, але при цьому практично корисною та доступною для інтерпретації людиною. Наприклад, шаблоном може бути стійке правило: «клієнти, які купують хліб та молоко, у

80% випадків також купують масло». Саме пошук таких прихованих шаблонів є головною метою Data Mining.

Таким чином, інтелектуальний аналіз даних можна визначити як процес виявлення в необроблених, великих за обсягом даних нових, раніше невідомих, нетривіальних, практично корисних і доступних для інтерпретації знань, необхідних для прийняття рішень у різноманітних сферах людської діяльності. Цей процес включає в себе збір, очищення, опис та моделювання даних. Для подання отриманих знань у Data Mining слугують моделі, які є формалізованим описом виявлених закономірностей.

Data Mining та Data Science: співвідношення понять

В сучасному IT-середовищі часто можна почути терміни Data Science та Data Mining. Іноді їх використовують як синоніми, однак між ними існує певна, хоча й не завжди чітко окреслена, межа. Data Science, або наука про дані, є ширшою міждисциплінарною галуззю, яка охоплює весь життєвий цикл даних: їх отримання, зберігання, обробку, аналіз та візуалізацію. Data Science більше зосереджена на дослідженні даних, пошуку фундаментальних закономірностей і створенні прогностичних моделей з використанням найсучасніших методів машинного навчання, статистики та інших дисциплін. Вона працює з усіма видами даних.

Data Mining, з іншого боку, часто розглядають як складову частину Data Science. Його можна охарактеризувати як прикладний процес, націлений на отримання конкретного практичного результату. Data Mining більше асоціюють зі збором, попередньою підготовкою, перетворенням та візуалізацією даних, особливо структурованих, а також із застосуванням алгоритмів для вирішення конкретних бізнес-задач. У цьому курсі ми виходитимемо з того, що Data Mining – це процес, який включає повний цикл перетворення даних на знання, необхідні для прийняття рішень у певній предметній області.

Міждисциплінарний характер Data Mining

Інтелектуальний аналіз даних є яскравим прикладом міждисциплінарної області, яка виникла та розвивається на перетині багатьох фундаментальних та прикладних наук. Він увібрав у себе ідеї, методи та підходи з таких дисциплін:

Теорія ймовірностей та математична статистика надають апарат для оцінки достовірності закономірностей, перевірки гіпотез та роботи з випадковими величинами.

Теорія інформації та математична логіка є основою для формалізації знань, вимірювання кількості інформації та побудови логічних висновків.

Розпізнавання образів та штучний інтелект пропонують алгоритми для класифікації, кластеризації та виявлення складних структур у даних.

Машинне навчання – це ядро багатьох алгоритмів Data Mining, яке дозволяє комп’ютерам навчатися на даних без явного програмування.

Теорія баз даних забезпечує ефективне зберігання, пошук та управління великими обсягами даних.

Візуалізація даних допомагає представити результати аналізу у зрозумілому для людини графічному вигляді та виявити візуальні закономірності.

У сучасних програмних системах Data Mining реалізовано величезну кількість методів та алгоритмів, які часто інтегрують у собі підходи одразу з кількох перелічених дисциплін.

Основні задачі інтелектуального аналізу даних

Data Mining охоплює широкий спектр задач, які умовно можна поділити на дві великі категорії: дескриптивні (описові) та предикативні (передбачувальні). Дескриптивні задачі спрямовані на покращення розуміння даних, виявлення прихованих структур та закономірностей. Предикативні задачі будують модель на основі наявних даних, яка потім використовується для прогнозування значень для нових, раніше невідомих об’єктів.

До основних задач Data Mining належать такі:

Класифікація є однією з найпоширеніших задач навчання з учителем. Її суть полягає у встановленні функціональної залежності, яка дозволяє віднести новий об’єкт до одного із заздалегідь визначених класів на основі його характеристик. Наприклад, на основі даних про клієнта банку (вік, дохід, кредитна історія) класифікаційна модель може віднести його до класу «надійний позичальник» або «ненадійний позичальник».

Кластеризація – це задача навчання без учителя, яка полягає в групуванні об’єктів на основі їхніх властивостей таким чином, що об’єкти всередині однієї групи (кластера) є максимально подібними, а об’єкти з різних груп – максимально відмінними. На відміну від класифікації, класи заздалегідь не відомі. Прикладом може бути сегментація клієнтів інтернет-магазину на групи зі схожими патернами покупок для створення персоналізованих пропозицій.

Пошук асоціацій спрямований на виявлення закономірностей між подіями, які відбуваються одночасно. Класичним прикладом є аналіз споживчого кошика, який дозволяє знайти товари, які часто купують разом. Таке правило може виглядати як: «якщо клієнт придбав пиво, то з ймовірністю 60% він також придбає чипси».

Пошук послідовностей є логічним розвитком задачі пошуку асоціацій. Тут закономірності встановлюються між подіями, пов’язаними в часі. Наприклад, аналіз послідовності покупок клієнта може показати, що після придбання смартфона через деякий час часто купують захисне скло або чохол.

Прогнозування – це задача визначення тенденцій та майбутніх значень певних показників на основі аналізу їхньої динаміки в минулому. Прикладами можуть бути прогнозування обсягів продажів, курсу валют, погодних умов.

Виявлення відхилень (або аналіз викидів) займається пошуком даних, які суттєво відрізняються від загальної сукупності. Такі відхилення можуть свідчити про помилки в даних, або, що більш цікаво, про унікальні, аномальні

явища. Наприклад, виявлення нехарактерних транзакцій по банківській картці може сигналізувати про шахрайство.

Аналіз зв'язків досліджує залежності між об'єктами в наборі даних. Це може бути кореляційний аналіз, або більш складні методи для виявлення причинно-наслідкових зв'язків.

Візуалізація є не просто окремою задачею, а й важливим інструментом Data Mining. Вона полягає у створенні графічних образів даних, які дозволяють аналітику наочно побачити структуру даних, кластери, тренди, викиди та інші закономірності, що важко виявити в табличному вигляді.

Підбивання підсумків має на меті створення компактного та змістовного опису для окремих груп об'єктів або для всього набору даних. Наприклад, для кожного виділеного кластеру клієнтів можна створити узагальнений портрет: «чоловіки віком 25–35 років із середнім доходом, які цікавляться електронікою».

Процес розв'язання задачі Data Mining

Розв'язання будь-якої задачі інтелектуального аналізу даних являє собою структурований процес, який включає кілька послідовних етапів. Першим і найважливішим етапом є постановка задачі, де чітко формулюються цілі аналізу з точки зору предметної області. Наступний етап – первинний аналіз даних, під час якого проводиться ознайомлення з наявними даними, вивчаються їхні характеристики, типи, розподіли. Далі йде найбільш трудомісткий етап – підготовка даних до аналізу, який включає очищення даних від шуму та викидів, обробку пропущених значень, нормалізацію та інші перетворення. Лише після цього розпочинається безпосередньо інтелектуальний аналіз даних із використанням обраного методу або алгоритму. Ключовим етапом є інтерпретація отриманих результатів, де аналітик оцінює знайдені закономірності з точки зору їхньої значущості, новизни та практичної корисності. Завершальний етап – це використання отриманих знань для прийняття рішень у відповідній предметній області.

Цей процес добре ілюструє схема перетворення даних. Спочатку з загального сховища відбувається відбір цільових даних, релевантних для поставленої задачі. Далі ці дані проходять етап попередньої обробки для підвищення їхньої якості. Після цього здійснюється перетворення даних у формат, придатний для застосування конкретних алгоритмів. На етапі інтелектуального аналізу даних відбувається безпосередній пошук закономірностей та побудова моделей. І, нарешті, на етапі інтерпретації відбувається отримання нових знань, які можна використовувати на практиці.

Інструментарій та сфери застосування Data Mining

Головною особливістю Data Mining є поєднання потужного математичного інструментарію, від класичних статистичних методів до новітніх кібернетичних алгоритмів, із сучасними інформаційними технологіями. На сьогодні існує величезна кількість програмних засобів, які реалізують алгоритми

інтелектуального аналізу даних. Це можуть бути як потужні аналітичні платформи (Weka, IBM SPSS, SAS), так і математичні пакети з відповідними бібліотеками (MatLab, MathCAD, Python з бібліотеками scikit-learn, pandas). Програмісти при розробці власних рішень можуть використовувати широкий спектр спеціалізованих бібліотек.

Сфери застосування Data Mining практично необмежені. Він активно використовується в банківській справі для скорингу та виявлення шахрайства, у маркетингу для сегментації клієнтів та аналізу ринкового кошика, у страхових компаніях для оцінки ризиків, у роздрібній торгівлі для управління асортиментом, у телекомунікаціях для аналізу трафіку, в охороні здоров'я для діагностики захворювань, у веб-аналітиці для аналізу поведінки користувачів та в роботі рекомендаційних систем. Важливо пам'ятати, що жоден, навіть найдосконаліший, програмний засіб не здатен повністю замінити аналітика. Успішне застосування технологій Data Mining потребує глибокого розуміння як самої предметної області, так і принципів роботи алгоритмів, що використовуються.

Питання для обговорення

1. Які передумови (технологічні та соціальні) призвели до виникнення та розвитку інтелектуального аналізу даних?
2. Розкрийте сутність поняття «шаблон» (pattern) у контексті Data Mining. Якими властивостями він повинен володіти?
3. Проаналізуйте різні варіанти перекладу терміну «Data Mining» українською мовою. Який з них, на вашу думку, найбільш точно відображає сутність процесу?
4. Поясніть співвідношення між поняттями «Data Mining» та «Data Science». У чому полягає основна відмінність між ними як процесами?
5. Чому Data Mining вважають міждисциплінарною областю? Назвіть основні науки, на які він спирається, та поясніть їхню роль.
6. Дайте означення та наведіть приклади таких задач Data Mining: класифікація, кластеризація, пошук асоціацій.
7. Чим відрізняються дескриптивні (описові) задачі Data Mining від предикативних (передбачувальних)? Наведіть приклади.
8. Поясніть різницю між навчанням з учителем (supervised learning) та без учителя (unsupervised learning). До яких задач Data Mining застосовується кожен з підходів?
9. Опишіть схему перетворення даних для виявлення знань. З яких основних етапів складається цей процес?
10. У чому полягає головна особливість Data Mining як технології? Чому наявність програмних засобів не замінює високої кваліфікації аналітика?

Лекція 2
Вступ до MatLab та попередня обробка даних

Місце MatLab в інтелектуальному аналізі даних

Перш ніж заглиблюватись у складні алгоритми інтелектуального аналізу, необхідно опанувати інструменти, за допомогою яких ці алгоритми будуть реалізовані. Одним із найпотужніших і найпоширеніших середовищ для наукових та інженерних розрахунків, у тому числі й для задач Data Mining, є MatLab. Назва цього програмного комплексу походить від словосполучення Matrix Laboratory, що підкреслює його основну ідею – виконання матричних обчислень. Вся інформація в MatLab, від простих чисел до складних масивів даних, інтерпретується як матриці, що робить його надзвичайно зручним для роботи з табличними даними, які є основою аналізу.

MatLab – це не просто калькулятор. Це інтегроване інтерактивне середовище, яке поєднує в собі мову програмування високого рівня, величезну бібліотеку готових функцій для чисельних методів, статистики та машинного навчання, а також потужні засоби для створення двовимірної та тривимірної графіки. Така інтеграція дозволяє виконувати повний цикл аналізу даних: від їх імпорту та первинного дослідження до реалізації складних алгоритмів і візуалізації отриманих результатів. Особливістю MatLab є його інтерпретуючий характер. Це означає, що кожна введена команда виконується негайно, що полегшує навчання, налагодження та експериментування з кодом. Користувач може створювати власні програми у вигляді сценаріїв та функцій, які зберігаються у файлах з розширенням m-файли, а дані та змінні – у бінарних файлах з розширенням mat-файли.

Основи роботи в середовищі MatLab

Робота в MatLab організована у вигляді сесій. Основним вікном для спілкування з програмою є Command Window, або вікно команд. Саме сюди користувач вводить інструкції, і тут же відображаються результати обчислень або повідомлення про помилки. Всі змінні, створені під час поточної сесії, відображаються у вікні Workspace, яке називають робочою областю. Це своєрідна пам’ять програми, де можна побачити назви змінних, їхні розміри та типи. Навігація файловою системою для доступу до m-файлів та mat-файлів здійснюється у вікні Current Folder, або поточної папки.

Роботу в командному режимі можна розпочати негайно, використовуючи MatLab як потужний калькулятор. Арифметичні вирази вводяться у командному рядку, і після натискання клавіші Enter одразу обчислюються. Для створення змінної достатньо просто присвоїти їй значення, наприклад, вказавши, що змінна A дорівнює сумі добутку числа на синус кута та кореня з числа. MatLab автоматично визначить тип змінної та розмірність. Якщо не вказати ім’я змінної, результат обчислення буде присвоєно спеціальній змінній, яка має назву ans. Важливо пам’ятати, що крапка з комою в кінці команди

пригнічує виведення результату на екран, що дуже корисно при створенні проміжних обчислень у великих програмах. Для очищення вікна команд використовується команда clc, а для видалення змінних з робочої області – команда clear.

Збереження поточного стану робочої області є критично важливою функцією. Виконавши серію розрахунків і створивши набір змінних, їх можна зберегти у mat-файлі за допомогою команди save або через графічний інтерфейс. У наступній сесії ці дані можна відновити командою load, що забезпечує неперервність досліджень.

Матриці як основа обчислень

Оскільки MatLab – це матрична лабораторія, вміння працювати з матрицями є фундаментальним. Матриця створюється переліком її елементів у квадратних дужках. Елементи в рядку розділяються пробілами, а самі рядки – крапкою з комою. Доступ до окремого елемента здійснюється за допомогою індексів у круглих дужках. Особливу роль відіграє оператор двокрапки, який використовується для створення діапазонів чисел, наприклад, для створення послідовності від одного числа до іншого з певним кроком, та для виділення частин матриць. За допомогою двокрапки можна звернутися до цілого рядка матриці або до кількох її елементів одночасно.

Програмування в MatLab: сценарії та функції

Для виконання послідовності дій, особливо при роботі з великими обсягами даних, зручно створювати програми. Найпростішим типом програм є файл-сценарій, який також називають Script-файлом. Це звичайний текстовий файл з розширенням m, який містить набір команд MatLab. При його виклику команди виконуються одна за одною. Сценарій працює з тими самими глобальними змінними, що й командне вікно. Його зручно використовувати для автоматизації рутинних задач, таких як завантаження даних, їх обробка та побудова графіків.

Більш потужним засобом є файли-функції. Вони також зберігаються у m-файлах, але мають чітко визначену структуру, яка починається з ключового слова function. Функція може приймати вхідні аргументи та повертати вихідні значення. На відміну від сценарію, всі змінні, створені всередині функції, є локальними і не впливають на робочу область. Це робить функції ідеальним інструментом для реалізації алгоритмів, які можна багаторазово використовувати з різними наборами даних. Наприклад, можна створити функцію для обчислення певного математичного виразу, зберегти її, а потім викликати з командного вікна або з іншої програми, підставляючи різні вхідні значення.

Візуалізація даних у MatLab

Візуалізація є невід’ємною частиною інтелектуального аналізу даних. Вона дозволяє побачити структуру даних, виявити приховані закономірності, оцінити

розподіл значень та перевірити якість попередньої обробки. Найпоширенішою функцією для побудови графіків є plot. Її базовий виклик будує графік залежності одного набору даних від іншого. Можна легко змінювати колір та тип лінії, додавати сітку, підписи осей та легенду.

Ще одним потужним інструментом візуалізації, особливо корисним для аналізу табличних даних, є теплова карта, або heatmap. Теплова карта представляє дані у вигляді матриці, де кожна комірка забарвлена у колір, що відповідає її числовому значенню. Це дозволяє миттєво оцінити загальну картину: де знаходяться максимальні значення, де мінімальні, чи є однорідні області або, навпаки, різкі перепади. В MatLab для створення теплової карти використовується функція HeatMap, яка дозволяє гнучко налаштовувати кольорову палітру для кращого сприйняття інформації.

Попередня обробка даних: нормалізація та стандартизація

Більшість алгоритмів Data Mining чутливі до масштабу ознак. Якщо одна ознака вимірюється в одиницях, а інша – в тисячах, то ознака з більшим масштабом домінуватиме в розрахунках, навіть якщо вона не є більш значущою. Уявімо, що ми аналізуємо дані про квартири. Ознака площа може мати значення від 30 до 150 квадратних метрів, а ознака поверх – від 1 до 24. Якщо ми спробуємо аналізувати відстані між об'єктами без попередньої обробки, то різниця в площах, яка вимірюється десятками, повністю перекриє різницю в поверхах, яка вимірюється одиницями. Саме тому попередня обробка даних, зокрема нормалізація та стандартизація, є обов'язковим етапом підготовки даних до аналізу.

Перед застосуванням цих перетворень корисно розрахувати базові статистичні показники, які описують набір даних. До мір центральної тенденції належать середнє арифметичне, яке є чутливим до викидів, медіана – значення, яке ділить впорядкований ряд навпіл і є стійкішим до викидів, та мода – найчастіше значення. Міри мінливості, такі як дисперсія та стандартне відхилення, показують, наскільки сильно розкидані значення навколо середнього. Знання мінімуму та максимуму також є необхідним для деяких методів нормалізації.

Нормалізація даних – це процес перетворення значень ознаки з одного діапазону в інший, найчастіше в діапазон від 0 до 1. Найпоширенішим методом є лінійна min-max нормалізація. Суть її полягає в тому, що від кожного значення віднімається мінімальне значення по всій вибірці, а потім ця різниця ділиться на розмах варіації, тобто на різницю між максимальним і мінімальним значеннями. В результаті найменше значення стає 0, найбільше – 1, а всі проміжні значення лінійно масштабуються в інтервал між 0 та 1. Повертаючись до нашого прикладу з квартирами, після min-max нормалізації площа 30 м² стане 0, а площа 150 м² стане 1. Так само поверх 1 стане 0, а поверх 24 стане 1. Таким чином, обидві ознаки отримують однакову вагу для подальшого аналізу.

Стандартизація даних переслідує дещо іншу мету: привести всі ознаки до такого вигляду, щоб вони мали середнє значення 0 та стандартне відхилення 1. Найпоширенішим методом є z-стандартизація. В цьому випадку від кожного значення віднімається середнє значення вибірки, а отримана різниця ділиться на

стандартне відхилення. Після такого перетворення одиницею виміру стає одне стандартне відхилення. Це дозволяє відповісти на питання: на скільки стандартних відхилень конкретне значення відхиляється від середнього. Якщо, наприклад, для певної квартири стандартизоване значення площі дорівнює 1,5, це означає, що її площа на півтора стандартних відхилення більша за середню площу. Такий підхід також робить різні за масштабом ознаки порівнянними між собою.

MatLab надає всі необхідні інструменти для цих перетворень. Можна обчислювати середні значення, стандартні відхилення, мінімуми та максимуми для матриць даних, використовуючи вбудовані функції. Використовуючи ці функції, аналітик може легко реалізувати будь-яку формулу нормалізації чи стандартизації. Гнучкість мови дозволяє створювати власні функції, наприклад, для min-max нормалізації, які можна буде використовувати в подальших дослідженнях, автоматизуючи процес підготовки даних.

Таким чином, перший крок у світ інтелектуального аналізу даних полягає в опануванні інструменту MatLab та фундаментальних процедур підготовки даних, а саме нормалізації та стандартизації. Це та основа, на якій будуватиметься все подальше вивчення алгоритмів класифікації, кластеризації та пошуку асоціацій. Розуміння того, як правильно підготувати дані, часто є важливішим за вибір найскладнішого алгоритму, оскільки якість вхідних даних безпосередньо визначає якість отриманих результатів.

Питання для обговорення

1. Яке основне призначення середовища MatLab? Що означає його назва і на чому базуються обчислення?
2. Опишіть структуру головного вікна MatLab. Які функції виконують вікна Command Window, Workspace та Current Folder?
3. Як створити змінну в MatLab? Яких правил слід дотримуватися при присвоєнні імен? Яке ім’я отримує змінна, якщо його не вказати?
4. Поясніть призначення файлів з розширеннями .mat, .m та .asv у середовищі MatLab.
5. Як зберегти змінні поточної сесії у файл та як їх відновити при наступному сеансі роботи? Наведіть приклади команд.
6. Як створити матрицю в MatLab? Поясніть, як отримати доступ до окремого елемента, рядка або стовпця матриці.
7. У чому полягає різниця між файлом-сценарієм (Script) та файлом-функцією (Function) у MatLab? Які області дії змінних у кожному з них?
8. Для чого використовуються графічні засоби MatLab? Як побудувати простий графік функції та теплову карту (heatmap)?
9. Яка мета нормалізації та стандартизації даних? У яких випадках застосування цих процедур є необхідним?
10. Поясніть суть min-max нормалізації та z-стандартизації. Як ці перетворення впливають на набір даних?

Лекція 3
Набори даних, шкали вимірювання та попередня обробка даних

Структура набору даних

Будь-яке дослідження в галузі інтелектуального аналізу даних починається з наявності вихідного матеріалу – набору даних. Набір даних – це необроблена інформація, яка має бути представлена у формі, придатній для зберігання, передачі та подальшого аналізу за допомогою методів Data Mining. Найзручнішим способом організації такого набору є двовимірна таблиця, де рядки відповідають окремим об’єктам, а стовпці – їхнім характеристикам, або атрибутам.

Об’єкт у цьому контексті – це конкретний елемент досліджуваної предметної області. Це може бути клієнт банку, пацієнт лікарні, товар в інтернет-магазині чи результат наукового експерименту. Кожен об’єкт описується набором атрибутів, які також називають ознаками, характеристиками або змінними. Змінна – це спільна для всіх об’єктів характеристика, прояв якої змінюється від об’єкта до об’єкта. Наприклад, для клієнтів банку спільними змінними можуть бути вік, сімейний стан та рівень доходу. Конкретні значення цих змінних для окремого клієнта (наприклад, вік 32 роки, дохід 1200 умовних одиниць) і будуть складати опис об’єкта.

Типи шкал вимірювання

У процесі підготовки даних важливо розуміти, в якій шкалі виміряна та чи інша ознака. Шкала – це правило, за яким характеристикам об’єктів присвоюються певні значення. Тип шкали визначає, які математичні та логічні операції можна застосовувати до цих значень, а отже, і які методи аналізу будуть коректними. Всі шкали поділяються на дві великі групи: категоріальні (якісні) та числові (метричні).

До категоріальних належать номінальна та порядкова шкали. Номінальна шкала використовується для позначення належності об’єкта до певної категорії. Значення в номінальній шкалі не можна впорядкувати. Прикладами є стать, національність, професія або місто проживання. Ми можемо сказати, що два об’єкти належать до однієї категорії або до різних, але не можемо стверджувати, що одна категорія "більша" або "краща" за іншу. Порядкова шкала, на відміну від номінальної, дозволяє впорядковувати об’єкти. Вона задає певну послідовність, але відстань між її градаціями не є строго визначеною. Типовими прикладами є військові звання, оцінки в школі, рівень освіти. Ми знаємо, що "вища освіта" є вищим рівнем, ніж "середня", але на скільки саме вона вища в числовому вираженні, визначити неможливо.

Числові шкали, своєю чергою, поділяються на інтервальні та шкали відношень. Інтервальна шкала не тільки впорядковує об’єкти, але й дозволяє визначити різницю між значеннями. Однак нуль у цій шкалі є умовним і не

означає відсутності ознаки. Класичним прикладом є температура за Цельсієм. Ми можемо сказати, що 20°C на 10°C тепліше за 10°C, але не можемо стверджувати, що 20°C вдвічі тепліше за 10°C, оскільки нуль шкали обрано умовно (точка замерзання води). Шкала відношень є найбільш інформативною. Вона має всі властивості попередніх шкал, але, що найважливіше, має абсолютну нульову точку, яка дійсно означає відсутність ознаки. Саме це дозволяє нам говорити про відношення: "один об’єкт у скількись разів більший за інший". Прикладами є вік, вага, зріст, дохід. Людина з доходом 2000 умовних одиниць заробляє вдвічі більше, ніж людина з доходом 1000 одиниць.

Окремо виділяють дихотомічну шкалу, яка є окремим випадком номінальної і містить лише дві категорії. Такі ознаки називають бінарними. Вони можуть бути симетричними, де обидва значення рівнозначні, як-от стать, або асиметричними, де одне значення є важливішим за інше, наприклад, результат медичного тесту, де позитивний результат несе набагато більше інформації, ніж негативний.

Основні завдання підготовки даних

Реальні дані, зібрані з різних джерел, майже ніколи не бувають ідеальними. Вони можуть бути неповними, тобто містити пропуски в значеннях атрибутів. Вони можуть бути неузгодженими, коли, наприклад, назви об’єктів в різних джерелах записані по-різному. Нарешті, вони майже завжди містять шум та викиди. Шум – це випадкові відхилення, які не несуть корисної інформації, а викиди – це аномальні значення, які різко відрізняються від основної маси даних. Тому етап підготовки даних, або Data Preparation, є критично важливим і, за різними оцінками, займає до 80% часу в будь-якому проекті з інтелектуального аналізу.

Процес підготовки даних включає кілька ключових завдань. Перше – це вибірка даних, тобто відбір тих об’єктів та їхніх ознак, які безпосередньо стосуються поставленої задачі. Далі йде інтеграція даних, якщо вони надходять з різних джерел і їх необхідно об’єднати. Найбільш трудомістким є етап очищення даних, метою якого є відновлення цілісності, усунення логічних суперечностей та уніфікація даних. Після очищення настає черга перетворення даних, яке включає нормалізацію, стандартизацію, дискретизацію та інші процедури, що приводять дані до вигляду, придатного для роботи алгоритмів. Окремо стоїть завдання генерації нових ознак, коли на основі наявних даних створюються похідні, які можуть краще відображати приховані закономірності. Завершальними етапами є скорочення даних, тобто зменшення кількості ознак без втрати важливої інформації, та форматування – синтаксичні зміни, що не впливають на суть, але роблять дані зручнішими для обробки.

Очищення даних: робота з пропусками та викидами

Очищення даних передбачає перевірку зібраної інформації на наявність різноманітних проблем. Це можуть бути недопустимі дані, які виходять за межі дозволеного діапазону, наприклад, вік 200 років. Це можуть бути логічно

непослідовні дані, коли, скажімо, у клієнта вказано стать "чоловіча" та діагноз "вагітність". Виявлені помилки можна виправити, якщо відомо, якими мають бути правильні значення, або виключити такі об’єкти з подальшого аналізу.

Особливу увагу слід приділяти пропускам у даних. Їх не можна просто ігнорувати. Існує кілька стратегій обробки пропусків. Найпростіша – виключити об’єкти з пропущеними значеннями з аналізу. Однак це може призвести до втрати великої кількості даних. Інший підхід – заповнити пропуски. Для числових змінних це часто роблять, підставляючи середнє арифметичне значення по всій вибірці. Для категоріальних змінних використовують моду, тобто найбільш часте значення. Існують і складніші методи, коли пропущене значення прогнозується на основі інших ознак того самого об’єкта.

Не менш важливим є виявлення викидів – значень, які різко виділяються на загальному фоні. Наприклад, дохід у 900 тисяч умовних одиниць серед доходів у кілька тисяч, швидше за все, є або помилкою, або рідкісним, але унікальним випадком. Для виявлення таких аномалій часто використовують простий статистичний підхід, заснований на квартилях. Перший квартиль – це значення, лівіше від якого лежить 25% усіх даних. Третій квартиль – значення, лівіше від якого лежить 75% даних. Різниця між ними називається інтерквартильним розмахом. Евристичне правило стверджує, що викидами можна вважати ті значення, які виходять за межі інтервалу, що дорівнює півтора інтерквартильним розмахам ліворуч від першого квартиля та праворуч від третього. Такі значення або виключають, або замінюють на граничні допустимі.

Перетворення даних: нормалізація, дискретизація, кодування

Після очищення дані часто потребують перетворення. Нормалізація та стандартизація, про які детально йшлося в попередній лекції, мають на меті привести всі числові ознаки до єдиного масштабу, щоб жодна з них не домінувала над іншими через більший діапазон значень. Min-max нормалізація лінійно масштабує значення в заданий інтервал, наприклад, від 0 до 1. Z-стандартизація перетворює дані так, що їхнє середнє стає рівним нулю, а стандартне відхилення – одиниці.

Іншим важливим видом перетворення є дискретизація. Вона застосовується до неперервних числових ознак і полягає в заміні точних числових значень на інтервальні або якісні мітки. Наприклад, замість точного віку людини ми можемо вказати, до якої вікової групи вона належить: "молодший", "дорослий", "старший". Або ж замінити значення зарплати інтервалами: "низька", "середня", "висока". Дискретизація може бути корисною для спрощення моделі та підвищення її інтерпретованості. Іноді інтервали об’єднують в ієрархічні структури, створюючи, наприклад, дерево понять для віку.

Окремої уваги потребують категоріальні ознаки. Більшість алгоритмів машинного навчання працюють лише з числами. Тому категоріальні значення

потрібно закодувати. Найпоширенішим методом є dummy-кодування, яке ще називають створенням фіктивних змінних. Ідея полягає в тому, що для кожної унікальної категорії вихідної ознаки створюється нова бінарна ознака. Якщо об’єкт належить до певної категорії, відповідна бінарна ознака отримує значення 1, а всі інші – 0. Наприклад, для ознаки "місто" з трьома можливими значеннями "Київ", "Харків", "Одеса" ми створимо три нові ознаки. Для об’єкта з містом "Київ" перша ознака дорівнюватиме 1, а дві інші – 0.

Однак dummy-кодування має свої обмеження. Якщо ознака має дуже багато унікальних значень, кількість нових бінарних ознак стає величезною, що ускладнює аналіз. Це називається "прокляттям розмірності". Крім того, якщо певна категорія зустрічається в даних дуже рідко, створена для неї бінарна ознака буде майже завжди нульовою і не нестиме корисної інформації. У таких випадках застосовують альтернативні підходи. Наприклад, рідкісні категорії можна об’єднати в одну групу "інші". Інший, більш витончений підхід, особливо корисний у задачах класифікації, полягає в заміні категорії на ймовірність належності до певного класу. Наприклад, якщо ми знаємо, що 67% клієнтів з Києва повертають кредити, то замість назви міста ми можемо підставити значення 0,67, яке відображає цю закономірність.

Також у процесі підготовки даних може застосовуватися агрегація, тобто згортання деталізованих даних до більш узагальненого вигляду. Для числових даних це може бути підсумовування, обчислення середнього, мінімуму або максимуму. Агрегація дозволяє перейти від даних на рівні окремих транзакцій до даних на рівні клієнтів або регіонів.

Таким чином, попередня обробка даних є фундаментом успішного аналізу. Ретельне виконання всіх її етапів – від розуміння типів шкал до кодування категоріальних змінних – значно підвищує якість результатів і дозволяє отримати дійсно цінні знання з наявних даних.

Питання для обговорення

1. Яке основне призначення середовища MatLab? Що означає його назва і на чому базуються обчислення?
2. Опишіть структуру головного вікна MatLab. Які функції виконують вікна Command Window, Workspace та Current Folder?
3. Як створити змінну в MatLab? Яких правил слід дотримуватися при присвоєнні імен? Яке ім’я отримує змінна, якщо його не вказати?
4. Поясніть призначення файлів з розширеннями .mat, .m та .asv у середовищі MatLab.
5. Як зберегти змінні поточної сесії у файл та як їх відновити при наступному сеансі роботи? Наведіть приклади команд.
6. Як створити матрицю в MatLab? Поясніть, як отримати доступ до окремого елемента, рядка або стовпця матриці.
7. У чому полягає різниця між файлом-сценарієм (Script) та файлом-функцією (Function) у MatLab? Які області дії змінних у кожному з них?

8. Для чого використовуються графічні засоби MatLab? Як побудувати простий графік функції та теплову карту (heatmap)?
9. Яка мета нормалізації та стандартизації даних? У яких випадках застосування цих процедур є необхідним?
10. Поясніть суть min-max нормалізації та z-стандартизації. Як ці перетворення впливають на набір даних?

Лекція 4
Первинний статистичний аналіз даних

Роль статистики в інтелектуальному аналізі даних

Перш ніж застосовувати складні алгоритми машинного навчання, необхідно ретельно вивчити наявні дані. Цей етап називається первинним статистичним аналізом. Він є фундаментом, на якому будується вся подальша робота. Статистичні методи дозволяють отримати узагальнені характеристики набору даних, виявити приховані закономірності, оцінити достовірність отриманих висновків та підготувати дані до застосування більш складних методів Data Mining. Без розуміння статистичної природи даних будь-яке моделювання буде неефективним, а його результати – недостовірними.

У статистичному аналізі ми майже ніколи не працюємо з усією сукупністю об’єктів, яка називається генеральною сукупністю. Натомість ми маємо справу з вибіркою – частиною об’єктів, відібраних для дослідження. Саме на основі аналізу вибірки ми робимо висновки про властивості всієї генеральної сукупності. Тому коректне формування вибірки та розуміння статистичних законів, за якими вона існує, є запорукою успішного аналізу.

Варіаційний ряд та його види

Перший крок у дослідженні числової ознаки – це впорядкування даних. Вихідні дані, отримані, наприклад, у результаті спостережень, зазвичай являють собою невпорядкований набір чисел. Для зручності аналізу на його основі будують варіаційний ряд – це послідовність значень ознаки, розміщених у порядку зростання, де кожне унікальне значення зустрічається лише один раз. Таке унікальне значення називається варіантою.

Для кожної варіанти важливо знати, як часто вона зустрічається у вихідній вибірці. Ця кількість називається частотою. Сума всіх частот дорівнює обсягу вибірки. Якщо поділити частоту кожної варіанти на загальний обсяг вибірки, ми отримаємо відносну частоту, яка є емпіричною оцінкою ймовірності появи цього значення.

Залежно від типу даних варіаційні ряди поділяються на дискретні та інтервальні. Дискретний ряд будується для ознак, які можуть набувати лише скінченної кількості значень, наприклад, кількість дітей у сім’ї. Для неперервних ознак, таких як зріст, вага або час, будують інтервальний варіаційний ряд. У цьому випадку вся область значень розбивається на певну кількість інтервалів, і підраховується частота попадання значень у кожен з них.

Кількість інтервалів може визначатися дослідником або за спеціальними правилами, які враховують обсяг вибірки. Таблиця, що містить варіанти (або інтервали) та відповідні їм частоти, задає емпіричний закон розподілу випадкової величини, тобто показує, як розподіляються ймовірності між можливими значеннями.

Графічне представлення розподілу

Візуалізація є потужним інструментом первинного аналізу. Вона дозволяє побачити форму розподілу, виявити симетрію, наявність викидів та інші особливості, які важко помітити в таблиці. Для дискретних рядів використовують полігон частот – ламану лінію, що з'єднує точки з координатами, де по горизонталі відкладено варіанти, а по вертикалі – їхні частоти або відносні частоти.

Для інтервальних рядів найпоширенішим графічним інструментом є гістограма. Вона складається з прямокутників, основою яких є інтервали, а висота відповідає частоті або відносній частоті. Гістограма дає наочне уявлення про щільність розподілу ймовірностей.

Ще одним важливим графіком є емпірична функція розподілу. Вона показує, яка частка об'єктів вибірки має значення ознаки, менші за певне число. Це зростаюча функція, яка починається від 0 і досягає 1. Її можна побудувати, обчисливши накопичені частоти – послідовну суму частот від мінімального значення до максимального. Емпірична функція розподілу є основою для оцінки теоретичної функції розподілу генеральної сукупності.

Числові характеристики вибірки

Для кількісного опису вибірки використовують числові характеристики, які поділяються на міри центральної тенденції та міри розсіювання. До мір центральної тенденції належать середнє арифметичне, мода та медіана.

Середнє арифметичне є найпоширенішою характеристикою, яка показує типовий рівень ознаки. Воно обчислюється як сума всіх значень, поділена на їх кількість. Однак середнє є чутливим до викидів – аномально великих або малих значень. Якщо у вибірці є викид, середнє може суттєво зміститися і перестати бути репрезентативним.

Мода – це значення, яке зустрічається у вибірці найчастіше. Вона не чутлива до викидів, але може бути не єдиною або взагалі відсутньою.

Медіана – це значення, яке ділить впорядковану вибірку навпіл: половина значень менша за медіану, половина – більша. Медіана є стійкою до викидів характеристикою, і її часто використовують разом із середнім для оцінки асиметрії розподілу. Якщо середнє і медіана близькі, розподіл є симетричним. Якщо середнє значно більше за медіану, це вказує на наявність великих викидів праворуч.

Міри розсіювання показують, наскільки сильно значення розкидані навколо центру. Найпростішою мірою є розмах варіації – різниця між максимальним та мінімальним значеннями. Однак він також дуже чутливий до

викидів. Найважливішими мірами є дисперсія та стандартне відхилення. Дисперсія характеризує середній квадрат відхилень значень від середнього. Стандартне відхилення – це квадратний корінь з дисперсії. Воно має ту саму розмірність, що й сама ознака, тому його зручно інтерпретувати. Наприклад, якщо середній зріст студентів 175 см, а стандартне відхилення 10 см, це означає, що більшість студентів мають зріст в інтервалі 165–185 см.

Оцінка параметрів генеральної сукупності

Характеристики, обчислені за вибіркою, називаються оцінками параметрів генеральної сукупності. Вони бувають точковими та інтервальними. *Точкова оцінка* – це одне число, яке є наближеним значенням параметра. Наприклад, вибіркове середнє є точковою оцінкою математичного сподівання генеральної сукупності.

Однак точкова оцінка не дає уявлення про її точність та надійність. Наскільки вона може відрізнятися від істинного значення? Відповідь на це питання дають інтервальні оцінки. Довірчий інтервал – це інтервал, який із заданою ймовірністю (надійністю) накриває невідоме істинне значення параметра. Чим ширший інтервал, тим нижча точність оцінки, але вища надійність. Зазвичай використовують надійність 95% або 99%. Ширина довірчого інтервалу залежить від обсягу вибірки та мінливості даних: чим більша вибірка і чим менший розкид, тим вужчим буде інтервал, а отже, точнішою оцінка.

Статистичні гіпотези та критерії згоди

У процесі аналізу даних часто виникає потреба перевірити певні припущення. Наприклад, чи відповідає розподіл досліджуваної ознаки нормальному закону? Чи є різниця між двома вибірками статистично значущою? Для цього формулюють статистичні гіпотези.

Основна гіпотеза, яку перевіряють, називається нульовою. Гіпотеза, що їй суперечить, – *альтернативною*. Рішення про прийняття або відхилення нульової гіпотези приймають на основі статистичного критерію. Це певне правило, яке дозволяє визначити, чи суперечать вибіркові дані нульовій гіпотезі.

Одним з найпоширеніших критеріїв є критерій згоди Пірсона, який використовують для перевірки гіпотези про відповідність емпіричного розподілу теоретичному. Суть його полягає у порівнянні емпіричних частот, отриманих з вибірки, з теоретичними частотами, які мали б місце, якби дані підпорядковувалися передбачуваному закону. Чим менша різниця між ними, тим більше підстав прийняти гіпотезу про відповідність. Розраховане за спеціальною формулою емпіричне значення критерію порівнюють із критичним значенням, яке залежить від обраного рівня значущості та кількості інтервалів. Якщо емпіричне значення менше за критичне, нульова гіпотеза приймається.

Інструменти статистичного аналізу в MS Excel

Для проведення первинного статистичного аналізу зручно використовувати прикладні програмні засоби, зокрема, MS Excel. Він надає широкий набір статистичних функцій та спеціалізований інструментарій – Пакет аналізу.

Статистичні функції Excel дозволяють обчислити всі необхідні числові характеристики: середнє, медіану, моду, дисперсію, стандартне відхилення та багато інших. Для роботи з варіаційними рядами існує функція ЧАСТОТА, яка є функцією масиву і дозволяє автоматично підрахувати частоти для заданих інтервалів.

Пакет аналізу є надбудовою, яку необхідно активувати. Він містить готові інструменти для виконання різноманітних статистичних процедур. Серед них – інструмент "*Описова статистика*", який генерує звіт, що містить основні числові характеристики вибірки: середнє, медіану, моду, стандартне відхилення, дисперсію, асиметрію, ексцес, а також довірчий інтервал для середнього. Це надзвичайно зручний засіб для швидкого отримання узагальненої інформації про дані.

Інший корисний інструмент – "*Гістограма*", який дозволяє не тільки побудувати звичайну гістограму, але й створити діаграму Парето. Діаграма Парето – це відсортована гістограма, яка візуалізує закон Парето, або принцип 80/20. Цей принцип стверджує, що 20% зусиль забезпечують 80% результату. На діаграмі Парето стовпці розташовані в порядку спадання частот, і додатково будується кумулятивна крива. Провівши горизонтальну лінію на рівні 80%, можна побачити, які нечисленні категорії (ті, що ліворуч) дають основний внесок у загальний результат. Це дозволяє зосередити увагу на найважливіших факторах.

Пакет аналізу також включає інструмент "Генерація випадкових чисел", який дозволяє створювати набори даних, що підпорядковуються різним законам розподілу: нормальному, рівномірному, пуассонівському тощо. Це корисно для імітаційного моделювання, перевірки статистичних методів та навчальних цілей.

Таким чином, первинний статистичний аналіз є невід’ємним етапом будь-якого дослідження даних. Він дає змогу глибоко зрозуміти структуру та властивості даних, обґрунтовано обрати методи подальшого аналізу та коректно інтерпретувати отримані результати. Опанування основних статистичних понять та інструментів є обов’язковою умовою для професійної роботи в галузі інтелектуального аналізу даних.

Питання для обговорення

1. Яке основне призначення середовища MatLab? Що означає його назва і на чому базуються обчислення?
2. Опишіть структуру головного вікна MatLab. Які функції виконують вікна Command Window, Workspace та Current Folder?

3. Як створити змінну в MatLab? Яких правил слід дотримуватися при присвоєнні імен? Яке ім’я отримує змінна, якщо його не вказати?
4. Поясніть призначення файлів з розширеннями .mat, .m та .asv у середовищі MatLab.
5. Як зберегти змінні поточної сесії у файл та як їх відновити при наступному сеансі роботи? Наведіть приклади команд.
6. Як створити матрицю в MatLab? Поясніть, як отримати доступ до окремого елемента, рядка або стовпця матриці.
7. У чому полягає різниця між файлом-сценарієм (Script) та файлом-функцією (Function) у MatLab? Які області дії змінних у кожному з них?
8. Для чого використовуються графічні засоби MatLab? Як побудувати простий графік функції та теплову карту (heatmap)?
9. Яка мета нормалізації та стандартизації даних? У яких випадках застосування цих процедур є необхідним?
10. Поясніть суть min-max нормалізації та z-стандартизації. Як ці перетворення впливають на набір даних?

Лекція 5
Ієрархічний кластерний аналіз

Сутність ієрархічної кластеризації

Кластерний аналіз є одним з основних методів інтелектуального аналізу даних, який відноситься до задач навчання без учителя. Його мета – розбити множину об’єктів на групи, які називаються кластерами, таким чином, щоб об’єкти всередині однієї групи були якомога більш схожими між собою, а об’єкти з різних груп – якомога більш відмінними. Ієрархічний кластерний аналіз займає особливе місце серед методів кластеризації, оскільки він будує не просто одне розбиття, а цілу систему вкладених розбиттів – ієрархію кластерів.

Результатом ієрархічної кластеризації є дерево кластерів, яке називається дендрограмою. Коренем цього дерева є весь набір даних, а листям – окремі об’єкти. На різних рівнях дерева ми бачимо, як об’єкти поступово об’єднуються у все більші групи або, навпаки, як велика група розпадається на менші. Дендрограма дає змогу наочно побачити структуру даних та обрати оптимальний рівень деталізації для подальшого аналізу.

Існує два основних підходи до побудови ієрархії кластерів. Агломеративні алгоритми працюють за принципом "знизу вгору". На початку роботи такого алгоритму кожен об’єкт вважається окремим кластером. Потім на кожному кроці два найближчі кластери об’єднуються в один. Процес триває доти, поки всі об’єкти не опиняться в одному великому кластері. Дивізимні алгоритми діють навпаки – "зверху вниз". Спочатку всі об’єкти належать до одного кластера, який потім поступово розбивається на дедалі дрібніші кластери. На практиці агломеративні алгоритми є значно поширенішими.

Етапи проведення ієрархічного кластерного аналізу

Процес ієрархічної кластеризації складається з кількох послідовних етапів. Перший етап – це вибір змінних, які будуть використані для кластеризації. Важливо включити в аналіз лише ті характеристики об'єктів, які є суттєвими з точки зору поставленої задачі. Додавання неінформативних ознак може зашумлювати дані та погіршувати якість кластеризації.

Другий етап – це підготовка даних. Оскільки більшість алгоритмів кластеризації базуються на обчисленні відстаней між об'єктами, необхідно, щоб усі змінні мали порівнянний масштаб. Якщо одна ознака вимірюється в одиницях, а інша – в тисячах, то друга ознака домінуватиме при розрахунку відстаней, що може призвести до некоректних результатів. Тому перед кластеризацією часто виконують нормалізацію або стандартизацію даних.

Третій, найважливіший етап – це вибір міри близькості між об'єктами. Це правило, за яким ми визначаємо, наскільки два об'єкти схожі або відмінні один від одного. Найпоширенішою мірою для числових даних є відстань Евкліда, яка геометрично інтерпретується як найкоротша відстань між двома точками у просторі ознак. Існують й інші метрики, такі як квадрат евклідової відстані, манхеттенська відстань або відстань Чебишева, кожна з яких має свої особливості та сфери застосування. Для категоріальних даних використовують інші міри, наприклад, коефіцієнти подібності. Результатом цього етапу є побудова матриці близькості – квадратної таблиці, де на перетині і-го рядка та j-го стовпця записана міра близькості між і-м та j-м об'єктами.

Четвертий етап – це вибір методу зв'язку кластерів. Це правило, за яким визначається відстань між двома кластерами після того, як вони вже містять більше ніж один об'єкт. Метод найближчого сусіда, або одиночного зв'язку, визначає відстань між кластерами як мінімальну відстань між будь-якими двома об'єктами з різних кластерів. Цей метод має тенденцію створювати довгі "ланцюжкові" кластери. Метод найдальшого сусіда, або повного зв'язку, навпаки, використовує максимальну відстань між об'єктами. Він добре працює, коли кластери є компактними та добре відокремленими. Метод середнього зв'язку обчислює середнє арифметичне всіх попарних відстаней між об'єктами з різних кластерів. Центроїдний метод визначає відстань між кластерами як відстань між їхніми центрами ваги, або центроїдами. Існує також метод Уорда, який об'єднує ті кластери, що призводять до найменшого зростання внутрішньогрупової дисперсії. Він часто дає дуже гарні результати, утворюючи компактні та інтерпретовані кластери.

П'ятий етап – це безпосереднє виконання алгоритму. Для агломеративного алгоритму це ітеративний процес: на кожному кроці знаходиться пара кластерів з найменшою відстанню, вони об'єднуються, і матриця відстаней перераховується з урахуванням нового кластеру. Процес триває до утворення одного кластеру. Результатом є послідовність об'єднань, яку можна візуалізувати у вигляді дендрограми.

Шостий етап – визначення оптимальної кількості кластерів. Оскільки ієрархічний алгоритм будує повне дерево, нам потрібно вирішити, на якому

рівні його "зрізати", тобто скільки кластерів вважати оптимальним. Ключовим орієнтиром є аналіз відстаней, на яких відбуваються об’єднання. Поки об’єднуються близькі, схожі кластери, ці відстані є відносно невеликими. Але на певному кроці відстань між кластерами, які об’єднуються, різко зростає. Це свідчить про те, що ми починаємо об’єднувати різнорідні групи. Саме цей момент є сигналом до зупинки, а оптимальна кількість кластерів дорівнює різниці між загальною кількістю об’єктів та номером кроку, після якого відбувся стрибок.

Останній, сьомий етап – це інтерпретація результатів. Отримавши кластери, необхідно зрозуміти, що вони означають. Для цього аналізують склад кожного кластеру та обчислюють його профіль – середні значення всіх ознак для об’єктів, що входять до кластеру. Порівнюючи профілі різних кластерів, можна дати їм змістовну назву та зробити висновки про структуру досліджуваної сукупності.

Практична реалізація в MS Excel

Незважаючи на те, що Excel не є спеціалізованим статистичним пакетом, він цілком придатний для виконання ієрархічного кластерного аналізу на невеликих наборах даних в навчальних цілях. Розглянемо процес на простому прикладі з шістьма об’єктами, кожен з яких описаний двома ознаками.

Спочатку будується матриця відстаней. Для цього для кожної пари об’єктів за обраною формулою обчислюється відстань. Отримані значення заповнюють симетричну матрицю, на головній діагоналі якої стоять нулі. Найменше значення в цій матриці вказує на пару найближчих об’єктів. У нашому прикладі це об’єкти 4 та 5. Саме вони об’єднуються в перший кластер на першому кроці.

Далі починаються ітерації. Після об’єднання об’єктів 4 і 5 у кластер ми будуємо нову матрицю відстаней меншого розміру. Її рядки та стовпці відповідають новому кластеру (4,5) та решті об’єктів: 1, 2, 3, 6. Відстані між об’єктами 1, 2, 3, 6 залишаються незмінними, а ось відстані між новим кластером та іншими об’єктами обчислюються за обраним методом зв’язку. Якщо використовується метод найближчого сусіда, то за відстань між кластером (4,5) та об’єктом 6 береться мінімальна з відстаней між об’єктом 6 та об’єктами 4 і 5. Якщо метод найдальшого сусіда – то максимальна.

Процес повторюється: на кожному кроці знаходиться мінімальний елемент у поточній матриці відстаней, відповідні кластери об’єднуються, і матриця знову перераховується. Результатом є послідовність об’єднань та значення відстаней на кожному кроці. Аналізуючи ці відстані, ми бачимо, що на останньому кроці, коли об’єднуються два великі кластери, відстань різко зростає. Це дозволяє визначити оптимальну кількість кластерів – у нашому прикладі їх два. За дендрограмою, яку можна побудувати за цими даними, легко побачити склад кластерів: до першого увійшли об’єкти 1, 2, 3, до другого – 4, 5, 6.

Ієрархічний кластерний аналіз у пакеті SPSS

Для роботи з реальними даними, які можуть містити десятки та сотні об'єктів, використовують спеціалізовані статистичні пакети, такі як SPSS. Розглянемо проведення ієрархічного кластерного аналізу на прикладі даних про зайнятість населення в різних галузях промисловості для 26 європейських країн. Мета – згрупувати країни зі схожою структурою економіки.

Після завантаження даних у SPSS необхідно вибрати в меню пункти Аналіз – Класифікація – Ієрархічна кластеризація. У діалоговому вікні, що відкриється, слід перенести всі числові змінні, що характеризують відсоток зайнятих у різних галузях, у поле змінних, а текстову змінну з назвами країн – у поле для міток спостережень. Це дозволить на дендрограмі бачити не номери, а назви країн.

У налаштуваннях графіків необхідно активувати опцію виведення дендрограми. У налаштуваннях методу важливо обрати спосіб стандартизації даних, наприклад, z-оцінки, щоб привести всі показники до єдиного масштабу. Також обираються метод об'єднання кластерів (наприклад, міжгруповий зв'язок) та міра відстані (наприклад, квадрат евклідової відстані). Після запуску розрахунку SPSS генерує звіт, який містить таблицю порядку агломерації та дендрограму.

Таблиця порядку агломерації показує, які саме об'єкти або кластери об'єднувалися на кожному кроці та на якій відстані. Аналіз коефіцієнтів у цій таблиці дозволяє визначити оптимальну кількість кластерів. У нашому прикладі стрибок відстані відбувається на 21-му кроці, тому оптимальна кількість кластерів дорівнює 26 (загальна кількість країн) мінус 21, тобто 5 кластерів.

Дендрограма дає змогу наочно побачити ієрархію кластерів. Провівши вертикальну лінію на відповідному рівні, можна визначити склад кожного з п'яти кластерів. Наприклад, до першого кластеру увійшли розвинуті капіталістичні країни, до другого – країни з аграрно-орієнтованою економікою, а деякі країни, такі як Туреччина та Югославія, утворили окремі кластери, що свідчить про унікальність їхньої економічної структури.

SPSS також дозволяє зберегти інформацію про належність кожної країни до кластеру у вигляді нової змінної. Далі, використовуючи процедуру порівняння середніх, можна обчислити профілі кластерів – середні значення всіх показників зайнятості для кожної з п'яти груп. Це дає змогу змістовно інтерпретувати отримані кластери. Наприклад, кластер розвинутих країн характеризується низькою зайнятістю в сільському господарстві та високою – у фінансовій та соціальній сферах.

Таким чином, ієрархічний кластерний аналіз є потужним інструментом розвідувального аналізу даних, який дозволяє виявити приховану структуру та згенерувати змістовні гіпотези про досліджувані об'єкти. Поєднання простих для розуміння алгоритмів з наочною візуалізацією робить його незамінним на початкових етапах дослідження.

Питання для обговорення

1. Яке основне призначення середовища MatLab? Що означає його назва і на чому базуються обчислення?
2. Опишіть структуру головного вікна MatLab. Які функції виконують вікна Command Window, Workspace та Current Folder?
3. Як створити змінну в MatLab? Яких правил слід дотримуватися при присвоєнні імен? Яке ім’я отримує змінна, якщо його не вказати?
4. Поясніть призначення файлів з розширеннями .mat, .m та .asv у середовищі MatLab.
5. Як зберегти змінні поточної сесії у файл та як їх відновити при наступному сеансі роботи? Наведіть приклади команд.
6. Як створити матрицю в MatLab? Поясніть, як отримати доступ до окремого елемента, рядка або стовпця матриці.
7. У чому полягає різниця між файлом-сценарієм (Script) та файлом-функцією (Function) у MatLab? Які області дії змінних у кожному з них?
8. Для чого використовуються графічні засоби MatLab? Як побудувати простий графік функції та теплову карту (heatmap)?
9. Яка мета нормалізації та стандартизації даних? У яких випадках застосування цих процедур є необхідним?
10. Поясніть суть min-max нормалізації та z-стандартизації. Як ці перетворення впливають на набір даних?

Лекція 6
Міри близькості між об’єктами набору даних

Вступ до задач класифікації та кластеризації

Перш ніж говорити про міри близькості, необхідно зрозуміти, для чого вони потрібні. В інтелектуальному аналізі даних існують дві фундаментальні задачі, пов’язані з групуванням об’єктів: класифікація та кластеризація.

Класифікація – це задача віднесення об’єкта до одного із заздалегідь відомих, наперед визначених класів. Наприклад, за результатами співбесіди ми можемо віднести кандидата до класу "підходить" або "не підходить" на певну посаду. Ми маємо набір правил або модель, побудовану на основі аналізу попередніх кандидатів, і застосовуємо її до нових об’єктів. Це навчання з учителем.

Кластеризація, навпаки, не потребує попереднього знання про класи. Її мета – виявити природне групування об’єктів у даних, розбити їх на групи (кластери) таким чином, щоб об’єкти всередині однієї групи були максимально схожі, а об’єкти з різних груп – максимально відмінні. Склад та кількість груп заздалегідь невідомі і визначаються в процесі аналізу. Це навчання без учителя, і воно часто є першим кроком у дослідженні нових даних.

В обох випадках – чи то для віднесення об’єкта до відомого класу, чи то для об’єднання їх у нові групи – нам потрібен інструмент для кількісного порівняння об’єктів. Цим інструментом і є міри близькості.

Поняття близькості: подібність та несхожість

Уявімо собі набір даних, представлений у вигляді таблиці "об’єкт-ознака", де рядки відповідають об’єктам, а стовпці – їхнім характеристикам. Кожен об’єкт можна розглядати як точку в багатовимірному просторі ознак. Міри близькості дозволяють визначити, наскільки близько або далеко одна від одної розташовані ці точки.

Існує два протилежних за змістом, але тісно пов’язаних поняття: міри подібності та міри несхожості. Міра подібності показує, наскільки об’єкти схожі. Вона набуває великих значень для дуже схожих об’єктів і малих – для несхожих. Для ідентичних об’єктів міра подібності зазвичай дорівнює одиниці. Міра несхожості, навпаки, показує, наскільки об’єкти відрізняються. Вона мала для схожих об’єктів і велика для несхожих. Для ідентичних об’єктів міра несхожості дорівнює нулю. Найпоширенішою мірою несхожості є відстань. Часто ці дві міри є взаємодоповнювальними: якщо міра несхожості нормована від 0 до 1, то міра подібності може бути обчислена як одиниця мінус міра несхожості.

Результатом попарного порівняння всіх об’єктів набору даних є матриця близькості. Це квадратна таблиця, де на перетині i-го рядка та j-го стовпця записана міра близькості між i-тим та j-тим об’єктами. Якщо в комірках записано відстані, матриця називається матрицею відстаней або матрицею несхожості. Якщо ж записано міри подібності, то це матриця подібності. Обидві матриці є симетричними відносно головної діагоналі, на якій для матриці відстаней стоять нулі, а для матриці подібності – одиниці. Для категоріальних даних для аналізу зв’язку між двома ознаками часто використовують таблиці спряженості, які показують частоту спільного появи різних значень цих ознак.

Міри близькості для різних типів даних

Вибір конкретної міри близькості залежить від типу даних, якими описані об’єкти. Для простих типів даних, коли об’єкти порівнюються лише за однією ознакою, правила є інтуїтивно зрозумілими. Для номінальних ознак, таких як "стать" або "колір", несхожість дорівнює нулю, якщо значення співпадають, і одиниці – якщо відрізняються. Подібність, відповідно, дорівнює одиниці при співпадінні і нулю при неспівпадінні. Для порядкових ознак, наприклад, "оцінка" (задовільно, добре, відмінно), значенням спочатку присвоюються ранги (0, 1, 2), а потім обчислюється нормована різниця між ними. Для числових ознак, таких як "вага" або "вік", найпростішою мірою несхожості є абсолютна різниця значень.

Міри відстаней для числових даних

Для числових даних, коли об’єкти описані кількома ознаками, існує багато різних метрик відстані. Кожна з них має свої особливості та сфери застосування. Найпоширенішою є відстань Евкліда. Це геометрична відстань між двома точками в багатовимірному просторі. Вона інтуїтивно зрозуміла і добре працює, коли ознаки є однорідними та мають однаковий масштаб. Якщо ж ознаки вимірюються в різних одиницях, перед обчисленням евклідової відстані обов’язково потрібно виконати нормалізацію або стандартизацію даних, інакше ознаки з більшими числовими значеннями домінуватимуть у відстані.

Квадрат евклідової відстані використовують тоді, коли хочуть надати більшої ваги об’єктам, що знаходяться далеко один від одного. Манхеттенська відстань, або відстань міських кварталів, обчислюється як сума абсолютних різниць координат. Вона менш чутлива до викидів, ніж евклідова відстань, оскільки різниці не підносяться до квадрату. Відстань Чебишева визначається як максимальна різниця за будь-якою з координат. Вона корисна, коли два об’єкти вважаються різними, якщо вони відрізняються хоча б за однією важливою ознакою.

Існують і складніші метрики, такі як відстань Махаланобіса, яка враховує кореляцію між ознаками. Вона дозволяє виявити, що об’єкти, які в евклідовому просторі здаються далекими, насправді можуть бути близькими, якщо врахувати структуру взаємозв’язків між змінними.

Міри подібності для числових даних

Окрім відстаней, для числових даних використовують і міри подібності. Найвідоміші з них – кореляція Пірсона та косинус подібності. Кореляція Пірсона оцінює тісноту лінійного зв’язку між двома об’єктами, розглядаючи їх як два ряди даних. Вона набуває значень від -1 (повна протилежність) до 1 (повний збіг). Косинус подібності обчислює косинус кута між двома векторами в просторі ознак. Він також змінюється від -1 до 1. Ці міри особливо популярні при аналізі текстових даних, де об’єкти представлені довгими та розрідженими векторами частот слів. Вони дозволяють порівнювати документи, ігноруючи їхню абсолютну довжину, і зосереджуючись на відносному розподілі термінів.

Міри близькості для категоріальних та бінарних даних

Для даних, представлених кількома категоріальними ознаками, найпростішою мірою несхожості є відстань Хеммінга, яка дорівнює кількості ознак, за якими об’єкти відрізняються. Якщо поділити цю кількість на загальну кількість ознак, отримаємо відсоток незгоди.

Окремим і дуже важливим випадком категоріальних даних є бінарні дані, де ознаки можуть набувати лише двох значень, наприклад, "так" або "ні", 0 або 1. Для них будують таблицю спряженості, яка показує, скільки разів обидва об’єкти мали значення 1, обидва мали значення 0, а також скільки разів їхні значення розрізнялися. Тут важливо розрізняти симетричні та асиметричні бінарні ознаки. Для симетричних ознак, де обидва значення рівнозначні

(наприклад, "чоловіча/жіноча" стать), використовують коефіцієнт простої відповідності, який враховує як збіги одиниць, так і збіги нулів. Для асиметричних ознак, де одне значення є важливішим за інше (наприклад, результат медичного тесту: "хворий/здоровий"), збіги нулів (обидва здорові) є менш інформативними. Тому застосовують коефіцієнт Жаккара, який враховує лише збіги одиниць, ігноруючи збіги нулів.

Коли об’єкти описані різнотипними ознаками (наприклад, числовими, номінальними та бінарними), застосовують комбінований підхід. Кожну ознаку нормують до діапазону [0, 1], обчислюють внесок цієї ознаки в несхожість за правилами, що відповідають її типу, а потім усереднюють ці внески. Таким чином можна отримати єдину міру несхожості для об’єктів зі змішаними типами даних.

Особливий випадок: розріджені вектори

Особливу увагу слід приділити розрідженим векторам, які часто зустрічаються при аналізі текстів. Кожен документ описується частотою слів (термінів) з великого словника. Більшість слів у кожному конкретному документі відсутні, тому вектори містять переважно нулі. У цьому випадку використання евклідової відстані або коефіцієнта простої відповідності було б помилковим, оскільки співпадіння нулів (відсутність одного й того ж слова в обох документах) не свідчить про їхню подібність. Для таких даних ідеально підходять косинус подібності, кореляція Пірсона та коефіцієнт Танімото, які ігнорують нульові збіги і зосереджуються на порівнянні ненульових частот.

Таким чином, вибір правильної міри близькості є критичним етапом у вирішенні задач класифікації та кластеризації. Він залежить від типу даних, масштабу ознак, наявності кореляцій та специфіки поставленої задачі. Розуміння цих нюансів дозволяє отримати змістовні та достовірні результати аналізу.

Питання для обговорення

1. Яке основне призначення середовища MatLab? Що означає його назва і на чому базуються обчислення?
2. Опишіть структуру головного вікна MatLab. Які функції виконують вікна Command Window, Workspace та Current Folder?
3. Як створити змінну в MatLab? Яких правил слід дотримуватися при присвоєнні імен? Яке ім’я отримує змінна, якщо його не вказати?
4. Поясніть призначення файлів з розширеннями .mat, .m та .asv у середовищі MatLab.
5. Як зберегти змінні поточної сесії у файл та як їх відновити при наступному сеансі роботи? Наведіть приклади команд.
6. Як створити матрицю в MatLab? Поясніть, як отримати доступ до окремого елемента, рядка або стовпця матриці.
7. У чому полягає різниця між файлом-сценарієм (Script) та файлом-функцією (Function) у MatLab? Які області дії змінних у кожному з них?

8. Для чого використовуються графічні засоби MatLab? Як побудувати простий графік функції та теплову карту (heatmap)?
9. Яка мета нормалізації та стандартизації даних? У яких випадках застосування цих процедур є необхідним?
10. Поясніть суть min-max нормалізації та z-стандартизації. Як ці перетворення впливають на набір даних?

Лекція 7
Алгоритми k-means та c-means. Кластерний аналіз у MatLab

Типи алгоритмів кластерного аналізу

Кластерний аналіз, як ми вже знаємо, призначений для виявлення природних груп об'єктів у даних. Залежно від підходу до побудови цих груп, усі алгоритми кластеризації можна поділити на кілька категорій. За способом обробки даних виділяють ієрархічні та неієрархічні алгоритми. Ієрархічні алгоритми, які ми розглядали раніше, будують дерево вкладених кластерів. Неієрархічні, або плоскі, алгоритми створюють лише одне розбиття об'єктів на кластери, які не перетинаються.

Інший важливий поділ – на чіткі та нечіткі алгоритми. У чітких алгоритмах, таких як k-means, кожен об'єкт може належати лише одному кластеру. Межі між кластерами є жорсткими. У нечітких алгоритмах, таких як c-means, об'єкт може належати до кількох кластерів одночасно з різним ступенем належності. Це дозволяє відображати ситуації, коли об'єкт знаходиться на межі між групами і не може бути однозначно віднесений до жодної з них.

Алгоритми квадратичної похибки. Ідея k-means

Алгоритми квадратичної похибки, до яких належить k-means, є одними з найпоширеніших плоских алгоритмів. Їхня мета – мінімізувати цільову функцію, яка являє собою суму квадратів відстаней від кожного об'єкта до центру того кластера, до якого він належить. Центр кластера, який називають центроїдом, – це уявна точка в просторі ознак, координати якої є середніми арифметичними значень ознак усіх об'єктів даного кластера.

Роботу алгоритму k-means можна описати наступною послідовністю кроків. Спочатку необхідно визначити кількість кластерів k, на які ми хочемо розбити дані. Потім випадковим чином обираються k початкових центроїдів. Далі для кожного об'єкта обчислюється відстань до всіх центроїдів, і об'єкт відноситься до найближчого з них. Після того, як всі об'єкти розподілені, центроїди перераховуються як середнє арифметичне об'єктів, які увійшли до відповідного кластера. Ці два кроки – перерозподіл об'єктів і перерахунок

центроїдів – повторюються доти, доки склад кластерів не перестане змінюватися. Це і є критерієм зупинки алгоритму.

Уявімо собі простий приклад з одновимірними даними – віком відвідувачів вебсайту. Ми хочемо розділити їх на дві вікові групи. На першому кроці ми випадково обираємо два центри, наприклад, 16 та 22 роки. Далі кожного відвідувача відносимо до того центру, різниця з яким за віком є меншою. Після цього перераховуємо центри як середній вік у кожній групі. Наприклад, перша група може складатися з наймолодших відвідувачів, і її новий центр стане, скажімо, 19,5 років. Друга група, відповідно, отримає новий центр, наприклад, 47,9 років. Процес повторюється: ми знову порівнюємо вік кожного відвідувача з новими центрами і перерозподіляємо їх. Це триває доти, доки на якомусь кроці склад груп не стабілізується. В результаті ми отримаємо дві групи: одну – з молодших відвідувачів, іншу – зі старших.

Алгоритм k-means є простим, швидким і добре працює на великих наборах даних. Однак він має й недоліки: потрібно заздалегідь знати кількість кластерів k, він чутливий до випадкового вибору початкових центрів і до викидів, а також не може правильно обробити ситуацію, коли об’єкт знаходиться на однаковій відстані від двох центрів.

Нечітка кластеризація: алгоритм c-means

Алгоритм c-means, або fuzzy c-means, є нечітким узагальненням k-means. Він дозволяє об’єкту належати до кількох кластерів з різним ступенем належності. Цей ступінь належності виражається числом від 0 до 1, і для кожного об’єкта сума ступенів належності до всіх кластерів дорівнює одиниці. Результатом роботи алгоритму є не жорстке розбиття, а матриця нечіткого розбиття, де в рядках – кластери, у стовпцях – об’єкти, а в комірках – коефіцієнти належності.

Етапи алгоритму c-means дещо складніші. Після ініціалізації параметрів (кількості кластерів k, коефіцієнта нечіткості w, параметра зупинки) випадковим чином задається початкова матриця нечіткого розбиття. Потім на її основі обчислюються центроїди кластерів. Далі, виходячи з відстаней до центроїдів, перераховуються коефіцієнти належності. Цей ітераційний процес триває доти, доки зміна цільової функції (або матриці розбиття) між двома послідовними ітераціями не стане меншою за заданий параметр зупинки. На завершальному етапі для прийняття рішення кожен об’єкт відносять до того кластера, для якого його коефіцієнт належності є найбільшим.

Коефіцієнт нечіткості w впливає на "розмитість" розбиття. При значеннях w, близьких до 1, алгоритм поводиться майже як чіткий k-means. Чим більше w, тим більш розмитими стають межі між кластерами.

Модифікації алгоритмів

Як k-means, так і c-means мають численні модифікації, які дозволяють адаптувати їх до різних типів даних та задач. Вони можуть відрізнятися способом визначення відстані між об’єктами (наприклад, використовувати не

евклідову, а манхеттенську відстань), способом обчислення центроїда (наприклад, використовувати медіану замість середнього арифметичного), або видом цільової функції, яка оптимізується (наприклад, максимізувати суму мір подібності, а не мінімізувати суму відстаней). Це робить сімейство алгоритмів квадратичної похибки дуже гнучким інструментом.

Кластерний аналіз у середовищі MatLab

MatLab надає потужний набір функцій для реалізації як ієрархічних, так і неієрархічних алгоритмів кластеризації. Для ієрархічної кластеризації використовуються функції pdist (обчислення попарних відстаней), linkage (побудова ієрархічного дерева) та dendrogram (візуалізація дерева). Функція pdist дозволяє обрати різноманітні метрики відстані, а функція linkage – різні методи зв'язку кластерів (найближчого сусіда, найдальшого сусіда, середнього зв'язку, метод Уорда тощо). Побудована дендрограма дає змогу візуально оцінити структуру даних та визначити оптимальну кількість кластерів за стрибком відстаней. Якість отриманого ієрархічного розбиття можна оцінити за допомогою функції cophenet, яка повертає значення, близьке до коефіцієнта кореляції – чим воно ближче до 1, тим краще дерево відображає структуру даних.

Для чіткої кластеризації за алгоритмом k-means у MatLab існує функція kmeans. Вона приймає матрицю даних та бажану кількість кластерів, а повертає вектор з номерами кластерів для кожного об'єкта та координати центроїдів. Функція kmeans має багато параметрів, які дозволяють керувати процесом кластеризації, наприклад, обирати спосіб ініціалізації центроїдів або максимальну кількість ітерацій.

Для нечіткої кластеризації за алгоритмом c-means використовується функція fcm з пакету Fuzzy Logic Toolbox. Вона повертає матрицю центроїдів, матрицю нечіткого розбиття та історію значень цільової функції. Аналізуючи матрицю нечіткого розбиття, ми бачимо, наприклад, що перший об'єкт з імовірністю 0,94 належить до першого кластера, а п'ятий – з імовірністю 0,95 до другого. На основі цих даних ми можемо прийняти остаточне рішення про належність об'єктів до кластерів.

MatLab також надає чудові можливості для візуалізації результатів кластеризації. За допомогою функцій plot та scatter можна відобразити об'єкти різних кластерів різними кольорами та маркерами, а також показати положення центроїдів.

Таким чином, поєднання ієрархічних методів для попереднього аналізу та визначення кількості кластерів з алгоритмами k-means та c-means для фінального розбиття є потужною стратегією дослідження даних. Середовище MatLab з його багатим набором функцій робить реалізацію цієї стратегії простою та ефективною.

Питання для обговорення

1. Яке основне призначення середовища MatLab? Що означає його назва і на чому базуються обчислення?
2. Опишіть структуру головного вікна MatLab. Які функції виконують вікна Command Window, Workspace та Current Folder?
3. Як створити змінну в MatLab? Яких правил слід дотримуватися при присвоєнні імен? Яке ім’я отримує змінна, якщо його не вказати?
4. Поясніть призначення файлів з розширеннями .mat, .m та .asv у середовищі MatLab.
5. Як зберегти змінні поточної сесії у файл та як їх відновити при наступному сеансі роботи? Наведіть приклади команд.
6. Як створити матрицю в MatLab? Поясніть, як отримати доступ до окремого елемента, рядка або стовпця матриці.
7. У чому полягає різниця між файлом-сценарієм (Script) та файлом-функцією (Function) у MatLab? Які області дії змінних у кожному з них?
8. Для чого використовуються графічні засоби MatLab? Як побудувати простий графік функції та теплову карту (heatmap)?
9. Яка мета нормалізації та стандартизації даних? У яких випадках застосування цих процедур є необхідним?
10. Поясніть суть min-max нормалізації та z-стандартизації. Як ці перетворення впливають на набір даних?

Лекція 8
Задача класифікації. Дискримінантний аналіз даних

Вступ до задачі класифікації

Класифікація є однією з найважливіших і найпоширеніших задач інтелектуального аналізу даних. Її суть полягає у віднесенні нового об’єкта до одного із заздалегідь відомих класів на основі аналізу його характеристик. На відміну від кластеризації, де класи формуються в процесі аналізу, у класифікації множина класів є фіксованою і відомою наперед. Саме тому класифікацію відносять до методів навчання з учителем.

Уявімо собі типовий приклад – фільтрацію електронної пошти. Нам потрібно навчити програму відрізняти звичайні листи від спаму. Ми маємо набір листів, для яких заздалегідь відомо, чи є вони спамом, чи ні. Цей набір називається навчальною множиною. Аналізуючи характеристики (ознаки) цих листів, такі як частота появи певних слів, наявність вкладень, адреса відправника, ми будуємо модель – класифікатор. Потім ми перевіряємо якість цієї моделі на іншому наборі листів, для яких ми також знаємо правильну відповідь, але які не використовувалися для навчання. Це тестова множина. Якщо модель добре працює на тестових даних, її можна використовувати для класифікації нових, невідомих листів.

Ознаки, на основі яких приймається рішення, називаються незалежними змінними, або предикторами. Ознака, яку ми прогнозуємо (наприклад, "спам" або "не спам"), називається залежною змінною. У задачі класифікації залежна змінна є категоріальною, тобто вона може набувати лише скінченної кількості значень. Якщо ж залежна змінна є числовою і може набувати будь-яких значень у певному діапазоні, така задача називається задачею регресії.

Проблеми перенавчання та недонавчання

Побудова якісного класифікатора – це мистецтво балансування. Модель не повинна бути ані надто простою, ані надто складною. Якщо модель є надто простою, вона не зможе виявити складні закономірності в даних і даватиме багато помилок навіть на навчальній множині. Це явище називається недонавчанням.

Протилежна ситуація – перенавчання. Воно виникає, коли модель є надто складною і "запам’ятовує" не лише загальні закономірності, але й шум, випадкові викиди та специфічні особливості навчальних даних. Така модель ідеально працює на навчальній множині, але дуже погано – на нових, незнайомих даних. Перенавчена модель не здатна до узагальнення. Завдання аналітика – знайти золоту середину, побудувати модель, яка добре узагальнює дані і правильно класифікує нові об’єкти.

Оцінка ефективності класифікатора. Матриця помилок

Для оцінки якості побудованого класифікатора використовують спеціальний інструмент – матрицю помилок. Це квадратна таблиця, яка показує співвідношення між фактичними значеннями залежної змінної та значеннями, передбаченими класифікатором. Для задачі з двома класами (бінарної класифікації) матриця має розмір 2×2.

Розглянемо приклад з тією ж фільтрацією спаму. Матриця помилок міститиме чотири показники:

TP (True Positive) – істинно позитивні: кількість спам-листів, які класифікатор правильно визначив як спам.

TN (True Negative) – істинно негативні: кількість звичайних листів, які класифікатор правильно визначив як звичайні.

FP (False Positive) – хибно позитивні: кількість звичайних листів, які класифікатор помилково визначив як спам. Це помилка першого роду.

FN (False Negative) – хибно негативні: кількість спам-листів, які класифікатор помилково пропустив, визначивши їх як звичайні. Це помилка другого роду.

На основі цих чотирьох чисел можна розрахувати низку метрик, які всебічно характеризують роботу класифікатора:

Точність (Accuracy): частка правильних відповідей серед усіх об’єктів. Це найпростіша, але не завжди найкраща метрика, особливо якщо класи незбалансовані.

Повнота (Recall, Sensitivity): частка спам-листів, які були виявлені, серед усіх спам-листів. Висока повнота означає, що класифікатор рідко пропускає спам.

Точність позитивних результатів (Precision): частка дійсно спам-листів серед усіх листів, які класифікатор визначив як спам. Висока точність означає, що класифікатор рідко помиляється, приймаючи звичайний лист за спам.

Специфічність (Specificity): частка звичайних листів, які були правильно визначені, серед усіх звичайних листів.

F-оцінка (F-score): гармонійне середнє точності та повноти. Це збалансована метрика, яка дозволяє порівнювати класифікатори, коли важливі і точність, і повнота.

Аналізуючи ці показники, ми можемо зробити висновок про сильні та слабкі сторони нашого класифікатора і, за потреби, покращити його.

Сутність дискримінантного аналізу

Дискримінантний аналіз – це один з класичних статистичних методів класифікації. Його ідея полягає в тому, щоб побудувати за навчальною вибіркою так звану дискримінантну функцію. Це лінійна комбінація незалежних змінних, яка найкращим чином розділяє об’єкти різних класів.

Умова оптимальності такого поділу є інтуїтивно зрозумілою: ми прагнемо, щоб об’єкти всередині одного класу були якомога ближчими один до одного (мінімальна внутрішньогрупова варіація), а центри різних класів були якомога далі один від одного (максимальна міжгрупова варіація). Дискримінантна функція будується таким чином, щоб максимізувати відстань Махаланобіса між центрами класів. Ця відстань враховує не лише положення центрів, але й розкид об’єктів у класах та кореляцію між ознаками.

Етапи дискримінантного аналізу

Розглянемо покроково, як будується дискримінантна функція для випадку двох класів. Нехай ми маємо навчальну вибірку, де для кожного об’єкта відомі значення двох незалежних змінних (x_1 та x_2) і номер класу.

Розрахунок середніх. Для кожного класу окремо обчислюються середні значення кожної змінної. Ці середні є координатами центрів класів.

Центрування даних. Від значень кожної змінної для кожного об’єкта віднімається середнє значення цієї змінної по його класу. Отримані матриці називаються матрицями центрованих значень. Вони показують відхилення кожного об’єкта від "типового" представника свого класу.

Обчислення об’єднаної коваріаційної матриці. Ця матриця характеризує спільний розкид об’єктів в обох класах та кореляцію між змінними. Вона обчислюється на основі матриць центрованих значень.

Обчислення оберненої коваріаційної матриці. Для подальших розрахунків нам знадобиться матриця, обернена до коваріаційної.

Розрахунок коефіцієнтів дискримінантної функції. Вектор коефіцієнтів (a_1, a_2) обчислюється як добуток оберненої коваріаційної матриці на вектор різниці

між центрами класів. Таким чином, дискримінантна функція має вигляд: $f(x) = a_1 \cdot x_1 + a_2 \cdot x_2$.

Обчислення константи дискримінації. Для кожного класу обчислюється середнє значення дискримінантної функції (підставляючи середні значення змінних). Константа дискримінації C – це середнє арифметичне цих двох середніх. Вона є границею, яка розділяє класи.

Класифікація нових об'єктів. Для нового об'єкта з відомими значеннями x_1 та x_2 обчислюється значення дискримінантної функції $f(x)$. Якщо $f(x) > C$, об'єкт відноситься до першого класу, якщо $f(x) < C$ – до другого. Якщо значення дорівнює C, об'єкт знаходиться точно на межі.

Практична реалізація: спільна робота MS Excel та MatLab

Розглянемо застосування дискримінантного аналізу на конкретному прикладі. Нехай ми маємо дані про діяльність 11 промислових підприємств, кожне з яких характеризується двома показниками: фондовіддачею (x_1) та матеріаломісткістю (x_2). За допомогою кластерного аналізу ці підприємства було розділено на два класи: перший – з високою фондовіддачею та низькою матеріаломісткістю, другий – решта. Наше завдання – на основі цих 11 підприємств (навчальна вибірка) побудувати дискримінантну функцію та класифікувати за її допомогою чотири нових підприємства.

Перший етап – підготовка даних. В MS Excel ми створюємо таблиці з даними для кожного класу, обчислюємо середні значення, а потім будуємо матриці центрованих значень.

Другий етап – інтеграція з MatLab. Завдяки спеціальній надбудові, ми можемо легко передавати дані з Excel у MatLab і назад. Ми імпортуємо в MatLab центровані матриці, вектори середніх та інші дані. MatLab надає потужні засоби для матричних обчислень, що значно спрощує подальші розрахунки.

Третій етап – обчислення в MatLab. У командному вікні MatLab ми, використовуючи вбудовані функції, крок за кроком виконуємо всі необхідні операції: знаходимо об'єднану коваріаційну матрицю, обернену до неї матрицю, а потім і вектор коефіцієнтів дискримінантної функції.

Четвертий етап – повернення до Excel та завершення розрахунків. Отримані в MatLab проміжні результати (наприклад, коваріаційну матрицю) можна імпортувати назад в Excel. Тут ми можемо знайти обернену матрицю за допомогою функції МОБР, або ж зробити це безпосередньо в MatLab. В результаті ми отримуємо дискримінантну функцію, наприклад, $f(x) = 37{,}33 \cdot x_1 - 1{,}14 \cdot x_2$, та константу дискримінації $C = 46{,}22$.

П'ятий етап – класифікація. Підставляючи значення показників для кожного з чотирьох нових підприємств у дискримінантну функцію, ми обчислюємо $f(x)$ і порівнюємо з C. Якщо значення більше за C, підприємство відносимо до першого класу (високоефективні), якщо менше – до другого.

Таким чином, дискримінантний аналіз надає нам простий і прозорий інструмент для класифікації. Поєднання зручності введення та візуалізації даних в Excel з обчислювальною потужністю MatLab робить цей метод доступним та ефективним для вирішення практичних задач.

1. **Питання для обговорення**
2. Яке основне призначення середовища MatLab? Що означає його назва і на чому базуються обчислення?
3. Опишіть структуру головного вікна MatLab. Які функції виконують вікна Command Window, Workspace та Current Folder?
4. Як створити змінну в MatLab? Яких правил слід дотримуватися при присвоєнні імен? Яке ім’я отримує змінна, якщо його не вказати?
5. Поясніть призначення файлів з розширеннями .mat, .m та .asv у середовищі MatLab.
6. Як зберегти змінні поточної сесії у файл та як їх відновити при наступному сеансі роботи? Наведіть приклади команд.
7. Як створити матрицю в MatLab? Поясніть, як отримати доступ до окремого елемента, рядка або стовпця матриці.
8. У чому полягає різниця між файлом-сценарієм (Script) та файлом-функцією (Function) у MatLab? Які області дії змінних у кожному з них?
9. Для чого використовуються графічні засоби MatLab? Як побудувати простий графік функції та теплову карту (heatmap)?
10. Яка мета нормалізації та стандартизації даних? У яких випадках застосування цих процедур є необхідним?
11. Поясніть суть min-max нормалізації та z-стандартизації. Як ці перетворення впливають на набір даних?

Лекція 9
Основні методи розв’язання задачі класифікації

Сучасні підходи до класифікації: різноманітність методів

Задача класифікації, з якою ми познайомилися в попередній лекції, може бути розв’язана за допомогою багатьох різних методів. Кожен з них має свої сильні та слабкі сторони, свої припущення про дані та свій спосіб побудови вирішального правила. Умовно всі ці методи можна поділити на дві великі групи: статистичні методи та методи машинного навчання. До статистичних належать, наприклад, дискримінантний аналіз (розглянутий раніше), метод One Rule (1R) та наївний баєсів класифікатор (Naive Bayes). Методи машинного навчання – це, зокрема, метод k-найближчих сусідів (kNN), метод опорних векторів (SVM), дерева рішень та нейронні мережі. Різноманітність підходів дозволяє аналітику обрати найкращий інструмент для конкретної задачі з урахуванням особливостей даних та вимог до результатів.

Незалежно від обраного методу, результат його роботи часто може бути представлений у вигляді класифікаційних правил. Правило класифікації – це просте логічне висловлювання у формі "ЯКЩО (умова) ТО (висновок)". Умова перевіряє значення однієї або кількох незалежних змінних, об’єднаних логічними операціями, а висновок визначає клас, до якого належить об’єкт, що задовольняє цю умову. Наприклад, правило для визначення кредитоспроможності може виглядати так: ЯКЩО (вік > 35) ТА (дохід > 10 000) ТО (кредит надати). Простота та інтерпретованість таких правил є їхньою головною перевагою.

Простота як запорука ефективності: метод One Rule (1R)

Метод One Rule, або 1R, є, мабуть, найпростішим з усіх методів класифікації. Його назва говорить сама за себе: він будує правило класифікації на основі лише однієї, найбільш інформативної незалежної змінної. Попри свою простоту, цей метод часто дає напрочуд добрі результати і слугує чудовим базовим рівнем для порівняння зі складнішими алгоритмами.

Алгоритм роботи 1R складається з кількох кроків. Спочатку, якщо значення змінних є неперервними, їх необхідно дискретизувати, тобто розбити на інтервали. Далі для кожного можливого значення кожної незалежної змінної підраховується, який клас зустрічається найчастіше. На основі цього формується правило: "Якщо змінна X має значення A, то об’єкт належить до класу B". Для кожного такого правила обчислюється помилка – кількість об’єктів, які це правило класифікує неправильно. Після цього для кожної змінної ми маємо набір правил (по одному на кожне значення) і загальну помилку для цієї змінної. Нарешті, ми обираємо ту змінну, набір правил для якої має найменшу сумарну помилку. Саме ця змінна і буде використовуватися для класифікації нових об’єктів.

Розглянемо це на прикладі задачі про гру у футбол залежно від погодних умов. Маючи дані про минулі ігри, ми можемо для кожної погодної характеристики (спостереження, температура, вологість, вітер) підрахувати, як часто при певному її значенні гра відбувалася, а як ні. Наприклад, виявляється, що коли вологість нормальна, гра відбувалася у 6 випадках з 7, а коли висока – лише у 3 з 7. Правило для вологості "Якщо вологість нормальна, то грати" має лише одну помилку, що є найкращим показником серед усіх змінних. Отже, для класифікації нового дня ми дивимося лише на вологість. Якщо вона нормальна – прогнозуємо гру, якщо висока – не прогнозуємо.

Імовірнісний підхід: наївний баєсів класифікатор (Naive Bayes)

Наївний баєсів класифікатор базується на знаменитій теоремі Байєса, яка дозволяє обчислювати ймовірність події за умови, що сталася інша подія. У контексті класифікації ми хочемо знайти ймовірність того, що об’єкт з певними ознаками належить до певного класу. Теорема Байєса дає нам формулу для

обчислення цієї апостеріорної ймовірності на основі апріорної ймовірності класу та умовних ймовірностей значень ознак для цього класу.

Чому класифікатор називається "наївним"? Тому що він робить "наївне" припущення про незалежність усіх ознак. Тобто вважається, що значення, скажімо, температури ніяк не впливає на значення вологості. У реальному житті це, звісно, не завжди так, але, як не дивно, на практиці цей метод працює дуже добре, особливо на великих масивах даних. Він також передбачає, що значення ознак мають певний розподіл, найчастіше нормальний.

Розглянемо той самий приклад з футболом. Ми маємо новий день з погодними умовами: сонячно, спекотно, нормальна вологість, немає вітру. Нам потрібно визначити, чи відбудеться гра. Спочатку ми обчислюємо апріорні ймовірності: гра була у 9 випадках з 14 (ймовірність 0,64), не була – у 5 з 14 (0,36). Потім, використовуючи дані, ми обчислюємо умовні ймовірності. Наприклад, ймовірність того, що буде сонячно, за умови, що гра була, дорівнює 2/9 (0,22), а за умови, що гри не було – 3/5 (0,6). Аналогічно для інших ознак. Далі, використовуючи "наївне" припущення про незалежність, ми перемножуємо ці умовні ймовірності для кожного класу. Для класу "гра буде" отримуємо 0,22 * 0,22 * 0,67 * 0,67 = 0,022. Для класу "гри не буде" – 0,6 * 0,4 * 0,2 * 0,4 = 0,019. Потім ми множимо ці значення на апріорні ймовірності класів і ділимо на загальну ймовірність події. В результаті отримуємо апостеріорну ймовірність гри 0,67 і не-гри 0,33. Оскільки 0,67 > 0,33, ми відносимо новий день до класу "гра відбудеться".

Класифікація на основі подібності: метод k-найближчих сусідів (kNN)

Метод k-найближчих сусідів (kNN) має зовсім іншу логіку. Він не будує жодної моделі чи правил у явному вигляді. Натомість він запам’ятовує всю навчальну вибірку і класифікує новий об’єкт, знаходячи серед навчальних об’єктів k найбільш схожих на нього. Схожість визначається за допомогою мір близькості, про які ми говорили в одній з попередніх лекцій. Клас нового об’єкта визначається як клас, який є найпоширенішим серед цих k сусідів. Це називається "простим незваженим голосуванням". Існує також "зважене голосування", де голос кожного сусіда враховується з вагою, обернено пропорційною до відстані до нього: ближчі сусіди мають більший вплив на рішення.

Метод kNN має кілька важливих особливостей. По-перше, необхідно обрати міру близькості. Для числових даних найчастіше використовують евклідову відстань. По-друге, критичним є вибір параметра k – кількості сусідів. Якщо k занадто мале (наприклад, 1), модель буде надто чутливою до шуму та викидів. Якщо k занадто велике, модель може стати надто грубою і втратити здатність виявляти локальні особливості. Оптимальне значення k часто підбирають експериментально, наприклад, методом ковзного контролю. По-третє, оскільки всі ознаки мають різний масштаб, перед застосуванням методу обов’язково потрібно виконати нормалізацію або стандартизацію даних.

Повернемося до нашої задачі з ірисами. Візьмемо нову квітку з довжиною чашолистка 6,2 см та довжиною пелюстки 4,9 см. Ми хочемо визначити її вид. Встановимо кількість сусідів k = 10. За допомогою функції MatLab ми знаходимо 10 найближчих сусідів з навчальної вибірки. Виявляється, що 6 з них належать до класу Virginica, а 4 – до класу Versicolor. За простим голосуванням ми відносимо нову квітку до Virginica. Зважене голосування, яке враховує відстані, дає ще більшу перевагу для Virginica (390 проти 160), підтверджуючи це рішення.

Розділяй та володарюй: дерева рішень

Дерева рішень – це потужний та інтуїтивно зрозумілий метод класифікації. Він будує модель у вигляді ієрархічної деревоподібної структури, яка складається з вузлів та листків. Кожен внутрішній вузол містить перевірку певної ознаки, кожне ребро відповідає результату цієї перевірки, а кожен листок – це кінцевий класифікаційний висновок. Процес класифікації нового об’єкта починається з кореня дерева. На кожному вузлі перевіряється відповідна ознака, і залежно від її значення об’єкт "спрямовується" по одній з гілок. Цей процес триває, поки об’єкт не досягне листка, який і визначить його клас.

Побудова дерева відбувається зверху вниз. На кожному кроці алгоритм обирає ту ознаку, яка найкраще розділяє навчальну вибірку на класи. Критерієм якості розбиття може бути, наприклад, зменшення невизначеності (ентропії). Процес триває доти, доки не виконається певне правило зупинки (наприклад, досягнуто максимальної глибини дерева, або у вузлі залишилося занадто мало об’єктів). Після побудови дерево може бути занадто складним і перенавченим. Для покращення узагальнюючої здатності застосовують процедуру відсікання гілок (pruning), видаляючи найменш важливі частини дерева.

Розглянемо дерево рішень, побудоване для задачі з ірисами. У корені дерева знаходиться перевірка третьої ознаки – довжини пелюстки. Якщо вона менша за певне значення, об’єкт одразу потрапляє до листка і відноситься до класу Setosa. Якщо більша – рухається далі по дереву, де перевіряються інші ознаки. Візуалізація дерева (наприклад, за допомогою функції view в MatLab) дозволяє легко простежити цей шлях. Класифікація нової квітки зводиться до проходження цих перевірок.

Одним з найпотужніших вдосконалень дерев рішень є алгоритм "випадкового лісу" (Random Forest). Він будує не одне дерево, а цілий "ліс" з багатьох дерев, кожне з яких навчається на випадковій підмножині даних та ознак. Кожне окреме дерево може давати не дуже точні прогнози, але коли вони "голосують" разом, помилки окремих дерев компенсуються, і загальний результат є дуже точним і стійким до перенавчання. У MatLab для створення випадкового лісу використовується функція TreeBagger. Побудувавши ліс з, скажімо, 50 дерев, ми можемо класифікувати новий об’єкт, і результат буде визначено на основі "голосування" всіх дерев лісу.

Класифікація за допомогою гіперплощини: метод опорних векторів (SVM)

Метод опорних векторів (SVM) має геометричну інтерпретацію. Він намагається знайти в просторі ознак гіперплощину, яка найкращим чином розділяє об’єкти різних класів. "Найкращим чином" означає, що відстань (зазор) від цієї гіперплощини до найближчих об’єктів кожного класу є максимальною. Ці найближчі об’єкти, які фактично "підтримують" гіперплощину, називаються опорними векторами. Саме вони визначають положення розділяючої межі.

Класичний SVM будує лінійну гіперплощину і призначений для бінарної класифікації. Якщо класи не є лінійно роздільними в поточному просторі ознак, застосовують так званий "ядерний трюк". За допомогою спеціальних функцій (ядер) дані проєктуються в простір вищої розмірності, де вони стають лінійно роздільними. Після цього будується лінійна гіперплощина, яка при зворотній проєкції відповідає нелінійній межі в початковому просторі.

У задачі з ірисами ми можемо застосувати SVM для розділення двох класів, наприклад, Versicolor та Virginica, використовуючи дві ознаки. У MatLab це робиться за допомогою функції fitcsvm. Вона будує модель, знаходить опорні вектори (на графіку вони обводяться кружечками) і проводить розділяючу лінію. Для нового об’єкта функція predict визначає, з якого боку від цієї лінії він знаходиться, і відносить його до відповідного класу. Оцінити якість побудованої моделі можна за допомогою перехресної перевірки, яка покаже очікувану помилку класифікації.

Таким чином, ми розглянули п’ять різних методів класифікації: від гранично простого 1R до складних ансамблевих методів типу випадкового лісу. Кожен з них має своє теоретичне підґрунтя, свої переваги та недоліки. Практична реалізація цих методів у середовищі MatLab за допомогою вбудованих функцій робить їх доступними для вирішення найрізноманітніших задач. Вміння обирати та застосовувати відповідний метод залежно від специфіки даних є ключовою компетенцією фахівця з інтелектуального аналізу даних.

Питання для обговорення

1. Яке основне призначення середовища MatLab? Що означає його назва і на чому базуються обчислення?
2. Опишіть структуру головного вікна MatLab. Які функції виконують вікна Command Window, Workspace та Current Folder?
3. Як створити змінну в MatLab? Яких правил слід дотримуватися при присвоєнні імен? Яке ім’я отримує змінна, якщо його не вказати?
4. Поясніть призначення файлів з розширеннями .mat, .m та .asv у середовищі MatLab.
5. Як зберегти змінні поточної сесії у файл та як їх відновити при наступному сеансі роботи? Наведіть приклади команд.
6. Як створити матрицю в MatLab? Поясніть, як отримати доступ до окремого елемента, рядка або стовпця матриці.

7. У чому полягає різниця між файлом-сценарієм (Script) та файлом-функцією (Function) у MatLab? Які області дії змінних у кожному з них?
8. Для чого використовуються графічні засоби MatLab? Як побудувати простий графік функції та теплову карту (heatmap)?
9. Яка мета нормалізації та стандартизації даних? У яких випадках застосування цих процедур є необхідним?
10. Поясніть суть min-max нормалізації та z-стандартизації. Як ці перетворення впливають на набір даних?

Лекція 10
Пошук асоціативних правил. Алгоритм Apriori

Що таке асоціативні правила?

Уявіть, що ви аналізуєте дані про покупки у великому супермаркеті. Ви помічаєте, що дуже часто разом з хлібом купують молоко, а разом з чипсами – пиво. Якщо ці закономірності є стійкими і статистично значущими, вони можуть бути описані у вигляді правил: "Якщо покупець купує хліб, то з великою ймовірністю він також купить молоко". Саме такі правила називаються асоціативними. Вони дозволяють виявляти приховані, неочевидні зв’язки між елементами даних.

Історично ця задача виникла саме для аналізу "ринкового кошика" – набору товарів, які один покупець кладе до свого кошика за одну покупку. Аналізуючи тисячі таких кошиків, можна знайти типові шаблони покупок. Сьогодні сфера застосування асоціативних правил значно ширша. Їх використовують в рекомендаційних системах для пропозиції товарів або контенту, в медицині для виявлення поєднань симптомів, в інформаційній безпеці для пошуку аномальної поведінки, в біоінформатиці та багатьох інших галузях.

Для формального опису задачі вводяться кілька ключових понять. Елемент – це об’єкт нашого аналізу (товар, подія, симптом). Транзакція – це набір елементів, які з’явилися разом (один чек, одна сесія користувача, один медичний випадок). База даних транзакцій – це сукупність всіх транзакцій, доступних для аналізу. Предметний набір – це будь-яка непорожня множина елементів, що зустрічаються в транзакціях.

Асоціативне правило – це імплікація виду $X \rightarrow Y$, де X та Y є предметними наборами, які не перетинаються. X називають умовою, або антецедентом, а Y – наслідком, або консеквентом. Правило інтерпретується так: "якщо в транзакції присутній набір X, то в ній, швидше за все, присутній і набір Y". Довжина правила – це загальна кількість елементів в X та Y.

Як оцінити якість асоціативного правила?

Коли ми знаходимо тисячі потенційних правил, нам потрібні об’єктивні критерії, щоб відсіяти випадкові збіги та залишити лише найцікавіші та найдостовірніші. Основних таких критеріїв два: підтримка та достовірність. Існують також додаткові, більш тонкі метрики: ліфт, леверідж та поліпшення.

Підтримка (Support) правила X → Y показує, яка частка всіх транзакцій містить одночасно і набір X, і набір Y. Іншими словами, це ймовірність того, що випадково обрана транзакція буде містити обидва набори. Підтримка є мірою статистичної значущості правила. Якщо підтримка дуже мала, правило базується на надто рідкісних подіях, і йому не можна довіряти. Зазвичай аналітик задає мінімальний поріг підтримки, і до подальшого розгляду беруться лише правила, що його перевищують.

Достовірність (Confidence) правила X → Y показує, яка частка транзакцій, що містять набір X, також містить і набір Y. Іншими словами, це умовна ймовірність появи Y за умови, що з’явився X. Достовірність є мірою точності, або сили правила. Якщо достовірність дорівнює 1, то правило виконується завжди. Як і для підтримки, задають мінімальний поріг достовірності.

Повернемося до нашого прикладу з продуктовим магазином. Нехай ми маємо 10 транзакцій. Правило "салат → помідори" зустрічається в 4 транзакціях. Його підтримка дорівнює 4/10 = 0,4. Салат купували в 4 транзакціях, і в усіх цих 4 випадках разом з ним купували помідори. Отже, достовірність цього правила дорівнює 4/4 = 1. Правило "цукерки → помідори" також зустрічається в 4 транзакціях, тому його підтримка теж 0,4. Але цукерки купували в 6 транзакціях, а помідори разом з ними – лише в 4. Тому достовірність цього правила нижча: 4/6 ≈ 0,67.

Підтримка та достовірність – це найважливіші, але не єдині метрики. Розглянемо два правила: "помідори → салат" і "помідори → цукерки". Вони мають однакову підтримку (0,4) і однакову достовірність (0,57). Здавалося б, вони рівноцінні. Але чи це так? Тут нам допомагає ліфт (Lift). Ліфт показує, у скільки разів ймовірність зустріти Y за умови X вища за ймовірність зустріти Y просто так, випадково. Ліфт обчислюється як відношення достовірності правила до підтримки наслідку. Якщо ліфт більший за 1, зв’язок позитивний і значущий. Якщо ліфт дорівнює 1, зв’язку немає. Якщо ліфт менший за 1, зв’язок негативний (наявність X зменшує ймовірність Y). У нашому прикладі підтримка салату (ймовірність того, що його куплять) дорівнює 0,4. Ліфт для правила "помідори → салат" дорівнює 0,57/0,4 = 1,425. Це набагато більше за 1, що свідчить про сильний позитивний зв’язок. Підтримка цукерок дорівнює 0,6. Ліфт для правила "помідори → цукерки" дорівнює 0,57/0,6 = 0,95. Це майже 1, тому зв’язок між помідорами та цукерками випадковий. Отже, ліфт дозволив нам побачити те, що було приховане за однаковими підтримкою та достовірністю.

Іноді виникає ситуація, коли два правила мають однакові достовірність і ліфт. Тоді для їх порівняння можна використати леверідж (Leverage). Він показує різницю між частотою спільної появи X та Y і частотою, яку можна було б очікувати, якби вони були незалежними. Чим більший леверідж, тим

частіше зустрічається правило, тим воно значиміше для більшої кількості випадків. Нарешті, поліпшення (Improvement) показує, наскільки правило краще за просте випадкове вгадування. Якщо поліпшення більше за 1, правило має сенс.

Як знаходити асоціативні правила? Алгоритм Apriori

Найпростіший спосіб знайти всі асоціативні правила – це згенерувати всі можливі комбінації умов та наслідків і перевірити їх на відповідність порогам підтримки та достовірності. Однак кількість таких комбінацій зростає експоненційно зі збільшенням кількості елементів. Для реальних баз даних цей підхід є абсолютно неприйнятним через величезну обчислювальну складність.

Тому було розроблено ефективніші алгоритми, найвідомішим з яких є Apriori. Він базується на простій, але геніальній ідеї: якщо набір елементів є частим (тобто має підтримку вищу за заданий поріг), то всі його підмножини також є частими. І навпаки, якщо набір є нечастим, то всі його надмножини також будуть нечастими. Це дозволяє відсікати цілі "гілки" простору пошуку, не розглядаючи їх.

Алгоритм Apriori працює ітеративно, крок за кроком знаходячи часті набори дедалі більшої довжини.

Перший крок. Знаходимо всі часті одноелементні набори. Для цього просто підраховуємо, скільки разів зустрічається кожен елемент у всіх транзакціях. Ті елементи, чия підтримка нижча за заданий мінімальний поріг, відсіваються.

Другий крок. З частих одноелементних наборів формуємо кандидатів у двоелементні набори. Це всі можливі пари з елементів, що залишилися. Підраховуємо підтримку для кожної пари і знову залишаємо лише ті, що перевищують поріг.

Третій та наступні кроки. Процес повторюється: на основі частих (k-1)-елементних наборів формуємо кандидатів у k-елементні набори. Важливо, що кандидати утворюються лише з тих наборів, у яких всі підмножини є частими (це і є ключова ідея Apriori для відсікання). Потім підраховуємо їх підтримку і залишаємо часті.

Зупинка. Алгоритм завершується, коли на черговому кроці не вдається сформувати жодного кандидата.

Результатом роботи першої фази алгоритму є множина всіх частих наборів різної довжини.

Друга фаза – генерація правил. Для кожного знайденого часто набору ми перебираємо всі можливі способи розбити його на умову X та наслідок Y. Для кожного такого правила обчислюємо достовірність. Якщо вона вища за заданий мінімальний поріг, правило вважається знайденим і додається до підсумкової множини.

Розглянемо роботу алгоритму на простому прикладі з 4 транзакціями та товарами, що мають номери від 0 до 5. Нехай мінімальна підтримка дорівнює

0,5 (тобто набір має зустрічатися принаймні у 2 з 4 транзакцій), а мінімальна достовірність – 0,75.

На першому кроці ми знаходимо підтримку для кожного товару. Товари 1, 2, 3 та 5 мають підтримку 0,5 або більше, товари 0 та 4 – менше. Отже, часті одноелементні набори: {1}, {2}, {3}, {5}.

На другому кроці формуємо всі можливі пари з цих товарів: {1,2}, {1,3}, {1,5}, {2,3}, {2,5}, {3,5}. Підраховуємо їх підтримку. Виявляється, що пари {1,3}, {2,3}, {2,5}, {3,5} мають підтримку 0,5 і залишаються. Пари {1,2} та {1,5} – відсікаються.

На третьому кроці формуємо триелементні набори. З чотирьох частих пар ми можемо утворити лише один кандидата – {2,3,5}. Його підтримка дорівнює 0,5, тому він також є частим. Чотириелементні набори утворити вже неможливо, алгоритм зупиняється.

Тепер ми маємо всі часті набори. З них можна згенерувати правила. Наприклад, з набору {1,3} можна утворити правило {1} → {3} (якщо товар 1, то товар 3) та {3} → {1}. Обчислюємо їх достовірність. Правило {1} → {3} має достовірність 1, оскільки товар 1 зустрічається лише разом з товаром 3. Правило {3} → {1} має достовірність 2/3 ≈ 0,67, оскільки товар 3 зустрічається в трьох транзакціях, але тільки в двох з них разом з товаром 1. Аналогічно аналізуємо всі інші набори. Наприкінці залишаємо лише ті правила, чия достовірність не нижча за 0,75. У нашому прикладі це будуть, зокрема, правила: "якщо товар 1, то товар 3", "якщо товар 2, то товар 5", "якщо товар 5, то товар 2".

Таким чином, алгоритм Apriori дозволяє ефективно знаходити приховані та практично корисні закономірності у великих масивах даних, що робить його незамінним інструментом сучасної аналітики.

Питання для обговорення

1. Яке основне призначення середовища MatLab? Що означає його назва і на чому базуються обчислення?
2. Опишіть структуру головного вікна MatLab. Які функції виконують вікна Command Window, Workspace та Current Folder?
3. Як створити змінну в MatLab? Яких правил слід дотримуватися при присвоєнні імен? Яке ім’я отримує змінна, якщо його не вказати?
4. Поясніть призначення файлів з розширеннями .mat, .m та .asv у середовищі MatLab.
5. Як зберегти змінні поточної сесії у файл та як їх відновити при наступному сеансі роботи? Наведіть приклади команд.
6. Як створити матрицю в MatLab? Поясніть, як отримати доступ до окремого елемента, рядка або стовпця матриці.
7. У чому полягає різниця між файлом-сценарієм (Script) та файлом-функцією (Function) у MatLab? Які області дії змінних у кожному з них?
8. Для чого використовуються графічні засоби MatLab? Як побудувати простий графік функції та теплову карту (heatmap)?

9. Яка мета нормалізації та стандартизації даних? У яких випадках застосування цих процедур є необхідним?
10. Поясніть суть min-max нормалізації та z-стандартизації. Як ці перетворення впливають на набір даних?

Лекція 11
Задача прогнозування. Аналіз часових рядів

Вступ до задачі прогнозування

Прогнозування є однією з найважливіших і, водночас, найскладніших задач інтелектуального аналізу даних. Його мета – передбачити майбутні значення певних показників на основі аналізу минулих даних. Прогноз дозволяє зменшити невизначеність і ризики при прийнятті рішень у найрізноманітніших сферах: в економіці (прогнозування курсу валют, попиту на товари), у бізнесі (планування продажів), у медицині (прогнозування поширення захворювань), у web-аналітиці (прогнозування трафіку) та багатьох інших. Складність прогнозування полягає в необхідності обробки великих обсягів даних, виявленні прихованих закономірностей та врахуванні впливу численних факторів.

Тісно пов'язаними з прогнозуванням є задачі класифікації та регресії, де також відбувається передбачення значення залежної змінної. Однак ключовою особливістю прогнозування є наявність часової складової: дані впорядковані в часі, і майбутнє значення залежить від минулих. Основою для прогнозування є часові ряди. Методи прогнозування поділяються на статистичні (регресійний аналіз, методи аналізу часових рядів) та кібернетичні (нейронні мережі, дерева рішень, метод опорних векторів).

Часовий ряд: основні поняття та характеристики

Часовий ряд – це послідовність значень деякої ознаки, впорядкована в хронологічному порядку. Кожне окреме значення називається рівнем ряду, а кількість рівнів – довжиною ряду. Часовий ряд завжди складається з двох елементів: періоду (або моменту) часу та відповідного йому числового значення.

Залежно від характеру рівнів розрізняють моментні, інтервальні та похідні часові ряди. Моментний ряд фіксує стан явища на певний момент часу (наприклад, курс валют на конкретну дату). Інтервальний ряд показує результат за певний період (наприклад, кількість продажів за місяць). Похідний ряд утворений середніми або відносними величинами (наприклад, середня заробітна плата за місяць).

Рівні часового ряду не є статистично незалежними. Значення наступного рівня часто залежить від попередніх. Для кількісної оцінки цього зв'язку використовують автоковаріацію та автокореляцію. Автокореляція – це кореляційна залежність між послідовними рівнями одного й того ж ряду, зсунутими на певну кількість періодів (лаг). Коефіцієнт автокореляції показує

тісноту цього зв’язку. Наприклад, високе значення автокореляції першого порядку свідчить про те, що кожне наступне значення ряду тісно пов’язане з попереднім.

Важливою характеристикою є стаціонарність ряду. Стаціонарний ряд коливається навколо постійного середнього рівня з постійною дисперсією. Нестаціонарний ряд має тренд (тенденцію до зростання або спадання), сезонні або циклічні коливання. Більшість реальних економічних та соціальних процесів є нестаціонарними, що ускладнює їх аналіз.

Структура часового ряду та її виявлення

Аналіз часових рядів спрямований на виявлення його структури, тобто розкладання на складові компоненти, які потім можна моделювати окремо. Виділяють такі основні компоненти:

Тренд (T) – довгострокова тенденція зміни показника (зростання, спадання, стабільність).

Циклічна компонента (C) – коливання з тривалим періодом (більше року), пов’язані, наприклад, з економічними циклами.

Сезонна компонента (S) – коливання з фіксованим і відомим періодом (день, тиждень, місяць, квартал), зумовлені, наприклад, сезонними факторами.

Випадкова компонента (E) – залишок після виділення регулярних компонент, який є випадковим шумом.

Процес виділення компонент називається декомпозицією часового ряду. Залежно від характеру зв’язку між компонентами будують різні моделі. Якщо коливання мають приблизно сталу амплітуду, використовують адитивну модель:

$Y=T+C+S+E$. Якщо амплітуда коливань зростає або спадає разом із трендом, використовують мультиплікативну модель:

$Y=T\cdot C\cdot S\cdot E$.

Перш ніж будувати модель, необхідно виявити наявність тренду та інших компонент. Для цього використовують:

Графічний аналіз – візуальна оцінка наявності тенденції та коливань.

Перевірку наявності тренду за допомогою статистичних критеріїв, наприклад, методу перевірки різниць середніх рівнів (з використанням t-критерію Стьюдента) або критерію серій.

Автокореляційний аналіз – побудову та аналіз корелограми (графіка автокореляційної функції). Наявність значущої автокореляції першого порядку вказує на тренд. Якщо найбільшим є коефіцієнт автокореляції порядку k, це свідчить про наявність сезонних або циклічних коливань з періодом k.

Також на початковому етапі важливо виявити аномальні рівні (викиди). Для цього використовують, наприклад, критерій Ірвіна. Якщо викид спричинений технічною помилкою, його замінюють розрахунковим значенням. Якщо ж він відображає реальний, хоч і рідкісний, вплив факторів, його залишають.

Побудова моделі часового ряду
(на прикладі адитивної тренд-сезонної моделі)

Розглянемо процес побудови моделі на прикладі адитивної моделі, яка є сумою трендової, сезонної та випадкової компонент. Нехай ми маємо щоквартальні дані за 5 років. Графічний аналіз та автокореляційна функція підказують нам, що ряд має зростаючий тренд і сезонні коливання з періодом 4 квартали приблизно однакової амплітуди. Отже, обираємо адитивну модель.

Перший етап – вирівнювання ряду та виділення сезонної компоненти. Для цього використовують метод ковзної середньої. Спочатку розраховують ковзну середню з періодом, що дорівнює періоду сезонності (у нашому випадку 4). Потім, щоб "прив'язати" ці середні до конкретних моментів часу, виконують центрування – беруть середнє двох послідовних ковзних середніх. Отримана центрована ковзна середня містить лише тренд (і, можливо, циклічну компоненту), але вже без сезонності та шуму. Далі, віднімаючи центровану ковзну середню від фактичних значень ряду, ми отримуємо попередні оцінки сезонної компоненти.

Другий етап – розрахунок скоригованих значень сезонної компоненти. Для цього знаходять середні оцінки сезонної компоненти для кожного кварталу за всі роки. Оскільки сума сезонних компонент за всі квартали в адитивній моделі має дорівнювати нулю, обчислюють коригуючий коефіцієнт (як середнє відхилення від нуля) і віднімають його від середніх оцінок. Це дає остаточні, скориговані значення сезонної компоненти для кожного кварталу.

Третій етап – усунення сезонності та побудова тренду. Віднімаючи значення сезонної компоненти від кожного рівня вихідного ряду, ми отримуємо дані, що містять лише тренд і випадкову компоненту. На основі цих даних будується трендова модель. Для цього можна підібрати різні функції (лінійну, поліноміальну, експоненційну) і обрати ту, що має найвищий коефіцієнт детермінації R^2. У нашому прикладі найкращою виявилася лінійна модель, яка описує тренд рівнянням прямої.

Четвертий етап – розрахунок теоретичних рівнів за моделлю та оцінка випадкової компоненти. Додаючи значення тренду (розраховані за лінійним рівнянням) та відповідні значення сезонної компоненти, ми отримуємо теоретичні значення ряду за нашою моделлю. Різниця між фактичними значеннями та теоретичними і є випадковою компонентою (залишками).

Оцінка якості моделі та прогнозування

Останній, але надзвичайно важливий етап – оцінка адекватності та точності побудованої моделі.

Адекватність моделі перевіряється за допомогою аналізу випадкової компоненти. Якщо модель побудовано правильно, випадкова компонента має бути стаціонарним рядом, тобто коливатися навколо нуля без тренду, і не мати значущої автокореляції. Для цього будують корелограму залишків. Якщо всі коефіцієнти автокореляції залишків не виходять за межі довірчого інтервалу і

згасають, модель вважається адекватною. У нашому прикладі це підтверджується.

Точність моделі оцінюється за допомогою різних метрик, які показують, наскільки теоретичні значення відхиляються від фактичних. Найпоширеніші з них:

MAE (Mean Absolute Error) – середнє абсолютне відхилення.

MSE (Mean Squared Error) – середньоквадратична похибка.

MAPE (Mean Absolute Percentage Error) – середня абсолютна похибка у відсотках. Вважається, що при MAPE менше 10% точність прогнозу дуже висока, 10-20% – висока, 20-50% – задовільна, більше 50% – незадовільна.

У нашому прикладі ми отримали дуже низькі значення похибок, що свідчить про високу якість моделі.

Після підтвердження адекватності та точності модель можна використовувати для прогнозування. Для отримання прогнозу на майбутні періоди (наприклад, на наступні 4 квартали) ми просто підставляємо в рівняння тренду відповідні номери періодів (21, 22, 23, 24) і додаємо значення сезонної компоненти для цих кварталів. Графік, що поєднує фактичні дані та прогноз, наочно демонструє результат.

Інструменти прогнозування в MS Excel

MS Excel надає кілька зручних інструментів для прогнозування. По-перше, можна додати лінію тренду до графіка часового ряду, візуально оцінити її тип (лінійна, експоненційна тощо), вивести рівняння та коефіцієнт детермінації, а також продовжити лінію на потрібну кількість періодів уперед. По-друге, існують функції для прогнозування: ПЕРЕДСКАЗ (для точкового прогнозу на основі лінійної залежності), ТЕНДЕНЦІЯ (для прогнозу декількох значень за лінійним трендом), РІСТ (для прогнозу за експоненційним трендом). Ці функції є функціями масиву і потребують спеціального введення.

Найсучаснішим і найпотужнішим інструментом є Лист прогнозу. Він дозволяє створити прогноз на основі часового ряду в кілька кліків. Користувач може задати дату завершення прогнозу, довірчий інтервал, спосіб обробки сезонності (автоматично або вручну), а також способи заповнення пропусків. Результатом є новий робочий аркуш з детальною таблицею прогнозних значень та інтерактивним графіком, що показує історичні дані, прогноз та довірчі інтервали. Це робить складний процес прогнозування доступним для широкого кола користувачів.

Таким чином, аналіз часових рядів – це потужний, багатоетапний процес, який дозволяє не лише зрозуміти минулі тенденції, але й з певною точністю зазирнути в майбутнє, озброївши особу, що приймає рішення, цінною інформацією.

Питання для обговорення

1. Яке основне призначення середовища MatLab? Що означає його назва і на чому базуються обчислення?

2. Опишіть структуру головного вікна MatLab. Які функції виконують вікна Command Window, Workspace та Current Folder?
3. Як створити змінну в MatLab? Яких правил слід дотримуватися при присвоєнні імен? Яке ім’я отримує змінна, якщо його не вказати?
4. Поясніть призначення файлів з розширеннями .mat, .m та .asv у середовищі MatLab.
5. Як зберегти змінні поточної сесії у файл та як їх відновити при наступному сеансі роботи? Наведіть приклади команд.
6. Як створити матрицю в MatLab? Поясніть, як отримати доступ до окремого елемента, рядка або стовпця матриці.
7. У чому полягає різниця між файлом-сценарієм (Script) та файлом-функцією (Function) у MatLab? Які області дії змінних у кожному з них?
8. Для чого використовуються графічні засоби MatLab? Як побудувати простий графік функції та теплову карту (heatmap)?
9. Яка мета нормалізації та стандартизації даних? У яких випадках застосування цих процедур є необхідним?
10. Поясніть суть min-max нормалізації та z-стандартизації. Як ці перетворення впливають на набір даних?

Лекція 12
Робота з нейронними мережами в MatLab: класифікація та прогнозування

Що таке штучні нейронні мережі?

Штучні нейронні мережі (ШНМ) – це один з найпотужніших і найуніверсальніших інструментів сучасного інтелектуального аналізу даних. Вони являють собою обчислювальні системи, натхненні будовою та принципами роботи біологічних нейронних мереж, зокрема, мозку людини. ШНМ здатні до навчання, узагальнення та виявлення складних нелінійних закономірностей у даних, що робить їх надзвичайно ефективними для розв’язання задач класифікації, кластеризації, прогнозування, розпізнавання образів та багатьох інших.

Основою будь-якої нейронної мережі є штучний нейрон – спрощена математична модель біологічного нейрона. Нейрон отримує кілька вхідних сигналів, які імітують сигнали від дендритів. Кожен вхід має свою вагу, яка відповідає "силі" синаптичного зв’язку. Ваги можуть бути позитивними (збуджувальними) або негативними (гальмівними). Внутрішній стан нейрона (сигнал активації) обчислюється як зважена сума вхідних сигналів (часто з додаванням зміщення). Ця сума потім передається через функцію активації, яка перетворює її у вихідний сигнал нейрона. Вихідний сигнал може бути переданий далі іншим нейронам. Функція активації вносить нелінійність у роботу мережі, що дозволяє їй моделювати складні залежності. Найпоширенішими функціями активації є порогова (ступінчаста), лінійна та

сигмоїдна (логістична). Сигмоїдна функція особливо популярна завдяки своїй гладкості та простоті обчислення похідної, що важливо для алгоритмів навчання.

Архітектура нейронних мереж

Нейрони в мережі з'єднані між собою, утворюючи певну архітектуру або топологію. За способом організації нейронів розрізняють два основні типи архітектур: повнозв'язні (де кожен нейрон з'єднаний з усіма іншими) та шаруваті. Найпоширенішими є шаруваті мережі прямого поширення, де нейрони об'єднані в послідовні шари. Сигнал у таких мережах поширюється лише в одному напрямку: від вхідного шару через один або кілька прихованих шарів до вихідного шару. Приховані шари відіграють ключову роль у виявленні складних абстрактних ознак.

Найпростішим прикладом нейронної мережі є перцептрон. Він може бути одношаровим (всього один нейрон) або багатошаровим. Одношаровий перцептрон з пороговою функцією активації здатен класифікувати лише лінійно роздільні множини об'єктів. Багатошарові перцептрони, завдяки наявності прихованих шарів та нелінійних функцій активації, можуть розв'язувати набагато складніші задачі, включаючи класифікацію нелінійно роздільних даних.

Існують також інші, складніші архітектури. Рекурентні нейронні мережі мають зворотні зв'язки, що дозволяє їм зберігати інформацію про попередні стани. Це робить їх особливо ефективними для роботи з послідовними даними, такими як часові ряди, текст або мова. Згорткові нейронні мережі спеціалізуються на обробці даних з сітчастою топологією, наприклад, зображень, і широко використовуються в задачах комп'ютерного зору.

Як навчаються нейронні мережі?

Навчання нейронної мережі – це процес налаштування її внутрішніх параметрів, насамперед вагових коефіцієнтів, для успішного розв'язання поставленої задачі. Розрізняють три основні парадигми навчання: з учителем, без учителя та з підкріпленням.

Навчання з учителем використовується для задач класифікації та регресії. Мережі пред'являються приклади вхідних даних разом з правильними відповідями (цільовими значеннями). Мережа порівнює свій вихід з цільовим значенням, обчислює помилку і коригує ваги так, щоб зменшити цю помилку. Цей процес повторюється для багатьох прикладів багаторазово.

Навчання без учителя застосовується для кластеризації. Мережі подаються лише вхідні дані, і вона самостійно повинна виявити в них структуру, згрупувавши схожі об'єкти.

Навчання з підкріпленням є проміжним варіантом, де мережа навчається на основі сигналу "винагороди" за правильні дії.

Класичним алгоритмом навчання багатошарових мереж є алгоритм зворотного поширення помилки. Його суть полягає в наступному: після того, як

сигнал пройшов через мережу в прямому напрямку і був обчислений вихід, обчислюється помилка на виході. Потім ця помилка поширюється назад через мережу, від вихідного шару до вхідного, і для кожного нейрона обчислюється його внесок у загальну помилку. На основі цього внеску коригуються ваги зв'язків. Метою є мінімізація загальної помилки мережі, яка часто обчислюється як сума квадратів відхилень. Цей процес називається градієнтним спуском: ваги змінюються в напрямку, протилежному градієнту функції помилки, що веде до її зменшення.

Процес навчання організований у вигляді епох. Одна епоха – це одноразове пред'явлення мережі всіх прикладів з навчальної множини. Для великих наборів даних епоха може розбиватися на ітерації, коли приклади подаються не по одному, а групами (батчами). Навчання триває доти, доки помилка на навчальній або контрольній (тестовій) вибірці не перестане зменшуватися або не буде виконано іншу умову зупинки. Дуже важливо уникати перенавчання, коли мережа надто добре запам'ятовує навчальні дані, включаючи шум, і втрачає здатність до узагальнення на нових даних.

Класифікація за допомогою нейронних мереж у MatLab

MatLab надає потужні та зручні засоби для роботи з нейронними мережами. Розглянемо процес створення класифікатора на прикладі одношарового перцептрона для розділення об'єктів на два класи. Спочатку ми задаємо навчальну множину – матрицю вхідних даних X (де кожен стовпець – це один об'єкт з двома ознаками) та цільовий вектор T, який містить номери класів (0 або 1) для кожного об'єкта. За допомогою функції perceptron ми створюємо нейронну мережу, а функція configure налаштовує її входи та виходи відповідно до наших даних. Потім ми використовуємо функцію adapt для навчання мережі, пред'являючи їй дані. Результат навчання візуалізується: на графіку з'являються об'єкти двох класів (позначені різними символами) та розділяюча пряма, яку знайшов перцептрон. Ми можемо переглянути налаштовані ваги та зміщення, а також схему мережі за допомогою функції view. Навчену мережу можна зберегти і використати для класифікації нових об'єктів, просто подавши їх на вхід мережі як аргумент.

Для складніших задач, наприклад, класифікації об'єктів на чотири класи, нам знадобиться багатошарова мережа. У MatLab ми створюємо її за допомогою функції feedforwardnet, вказуючи кількість нейронів у кожному шарі, наприклад, [10, 2, 1]. Це тришарова мережа з 10 нейронами в першому (прихованому) шарі, 2 – у другому і 1 – у вихідному. Ми можемо задати функції активації для різних шарів (наприклад, logsig – сигмоїдна). Навчання здійснюється функцією train. Під час навчання відкривається вікно Neural Network Training, де в реальному часі можна спостерігати за прогресом: значенням функції помилки, градієнтом, кількістю епох. Після завершення навчання ми можемо переглянути графік продуктивності, гістограму помилок, регресійний аналіз, які підтверджують високу якість класифікації. Перевірка

мережі на нових, не бачених раніше об’єктах, показує, що вона успішно відносить їх до відповідних класів.

Графічний інтерфейс NNTool та майстер прогнозування

Окрім роботи в командному режимі, MatLab пропонує зручний графічний інтерфейс користувача – NNTool. Він запускається командою nntool. Це візуальне середовище дозволяє створювати, налаштовувати та навчати нейронні мережі без написання коду. Ми можемо імпортувати вхідні дані та цільові вектори, вибрати тип мережі (наприклад, feed-forward backprop), задати кількість шарів та нейронів, функції активації та метод навчання. Після створення мережі її можна навчити, натиснувши кнопку Train Network. Усі процеси та результати візуалізуються так само, як і в командному режимі. Навчену мережу можна експортувати в робочу область і використовувати для класифікації нових об’єктів.

Для прогнозування часових рядів у MatLab існує спеціальний майстер, який запускається командою nnstart, а потім вибором пункту Time Series app. Цей майстер крок за кроком проводить користувача через весь процес: вибір типу мережі (наприклад, нелінійна авторегресійна NAR), завантаження даних (можна скористатися вбудованими прикладами, як-от дані про сонячні плями), розбиття даних на навчальну, тестову та перевірочну вибірки, вибір архітектури мережі та алгоритму навчання. Після навчання майстер надає інструменти для оцінки якості моделі: графік автокореляції помилок, графік відгуку часового ряду, де порівнюються фактичні та прогнозовані значення, а також значення середньоквадратичної помилки (MSE) та коефіцієнта детермінації R. На завершення майстер пропонує згенерувати код функції, який можна зберегти та використовувати для отримання прогнозів на нових даних. Це робить складний процес побудови нейромережевих прогностичних моделей доступним і зрозумілим.

Таким чином, нейронні мережі є надзвичайно гнучким і потужним інструментом, а середовище MatLab надає всі необхідні засоби для їх ефективного застосування – від простих перцептронів до складних багатошарових архітектур, як у командному режимі, так і за допомогою зручних графічних інтерфейсів.

Питання для обговорення

1. Яке основне призначення середовища MatLab? Що означає його назва і на чому базуються обчислення?
2. Опишіть структуру головного вікна MatLab. Які функції виконують вікна Command Window, Workspace та Current Folder?
3. Як створити змінну в MatLab? Яких правил слід дотримуватися при присвоєнні імен? Яке ім’я отримує змінна, якщо його не вказати?
4. Поясніть призначення файлів з розширеннями .mat, .m та .asv у середовищі MatLab.

5. Як зберегти змінні поточної сесії у файл та як їх відновити при наступному сеансі роботи? Наведіть приклади команд.
6. Як створити матрицю в MatLab? Поясніть, як отримати доступ до окремого елемента, рядка або стовпця матриці.
7. У чому полягає різниця між файлом-сценарієм (Script) та файлом-функцією (Function) у MatLab? Які області дії змінних у кожному з них?
8. Для чого використовуються графічні засоби MatLab? Як побудувати простий графік функції та теплову карту (heatmap)?
9. Яка мета нормалізації та стандартизації даних? У яких випадках застосування цих процедур є необхідним?
10. Поясніть суть min-max нормалізації та z-стандартизації. Як ці перетворення впливають на набір даних?

Лекція 13
Виявлення зв'язків і закономірностей. Кореляційний та дисперсійний аналізи даних

Вступ до аналізу зв'язків

Аналіз даних рідко обмежується простим описом окремих ознак. Найцікавіші висновки народжуються з вивчення взаємозв'язків між ними. Чи впливає рівень освіти на заробітну плату? Чи пов'язаний вік людини з її політичними вподобаннями? Чи залежить успішність складання іспиту від кількості пропущених занять? Відповіді на ці та подібні питання дають методи статистичного аналізу зв'язків. Залежно від типів змінних, зв'язок між якими ми досліджуємо, застосовуються різні підходи.

Ми розрізняємо незалежні (впливаючі) та залежні (результативні) змінні. Аналіз зв'язку передбачає з'ясування кількох ключових моментів: чи існує зв'язок взагалі, наскільки він сильний (тіснота зв'язку), який його характер (прямий чи обернений) та якою є його форма (лінійна, нелінійна). Важливо пам'ятати, що статистичний зв'язок, виявлений між змінними, не завжди є причинно-наслідковим. Він лише вказує на узгоджену зміну їхніх значень, яка може бути зумовлена впливом третього, неврахованого фактора.

Виявлення зв'язків за допомогою таблиць спряженості

Коли обидві змінні є категоріальними (наприклад, стать та політичні переваги), найпростішим і найнаочнішим інструментом є таблиці спряженості. Така таблиця показує, як часто зустрічаються різні комбінації значень двох ознак. Наприклад, ми можемо побудувати таблицю, де рядки відповідають статі (чоловіча, жіноча), стовпці – підтримці одного з кандидатів (А або Б), а в комірках буде зазначено кількість виборців з відповідними характеристиками.

Візуальний аналіз такої таблиці вже може дати певне уявлення. Якщо, скажімо, серед чоловіків явно переважає підтримка кандидата А, а серед жінок – кандидата Б, це свідчить про наявність зв'язку. Однак для статистично обґрунтованого висновку використовують критерій згоди Пірсона (хі-квадрат). Він порівнює фактичні частоти, які ми спостерігаємо в таблиці, з теоретичними частотами, які мали б місце, якби зв'язку між змінними не було. Чим більша розбіжність між спостережуваними та очікуваними частотами, тим більше підстав відхилити гіпотезу про відсутність зв'язку. Розраховане значення критерію порівнюється з критичним, і на основі цього порівняння робиться висновок про статистичну значущість зв'язку.

Кореляційний аналіз: вимірювання зв'язку між кількісними змінними

Кореляційний аналіз застосовується, коли обидві змінні є кількісними (наприклад, вік і дохід, кількість пропущених занять і рейтинг). Його мета – визначити, чи існує між ними лінійна залежність, оцінити її силу та напрямок. Основною мірою є коефіцієнт кореляції, який змінюється в межах від -1 до 1.

Знак коефіцієнта вказує на напрямок зв'язку: додатне значення означає прямий зв'язок (зі зростанням однієї змінної зростає й інша), від'ємне – обернений (зі зростанням однієї, інша спадає).

Абсолютна величина коефіцієнта вказує на силу (тісноту) зв'язку. Чим ближче значення до 1 або -1, тим сильніший зв'язок. Значення, близькі до 0, свідчать про відсутність лінійної залежності.

Найпоширенішим є коефіцієнт кореляції Пірсона, який є параметричним методом. Це означає, що для його коректного застосування розподіл обох змінних має бути близьким до нормального. Він вимірює саме лінійний зв'язок. Якщо залежність є нелінійною, коефіцієнт Пірсона може бути близьким до нуля, навіть якщо зв'язок існує.

У випадках, коли розподіл змінних відрізняється від нормального, або коли дані представлені в порядковій шкалі, використовують непараметричні аналоги, зокрема коефіцієнт рангової кореляції Спірмена. Для його обчислення значення змінних замінюються рангами (порядковими номерами у впорядкованому ряду), і кореляція обчислюється між цими рангами.

Обчисленого значення коефіцієнта недостатньо для остаточного висновку. Необхідно перевірити його статистичну значущість, тобто переконатися, що він не є випадковим для даної вибірки. Для цього формулюють гіпотезу про рівність коефіцієнта нулю в генеральній сукупності та перевіряють її за допомогою t-критерію Стьюдента. Якщо розраховане значення критерію перевищує критичне, коефіцієнт вважається значущим.

Потужним інструментом візуалізації кореляційного зв'язку є діаграма розсіювання. Кожна точка на ній відповідає одному об'єкту з координатами, що дорівнюють значенням двох досліджуваних змінних. Форма "хмари" точок дозволяє візуально оцінити наявність, силу та напрямок зв'язку. Витягнутий еліпс точок свідчить про наявність кореляції, а його орієнтація – про її напрямок.

Розглянемо класичний приклад: залежність між кількістю пропущених занять та підсумковим рейтингом студента. Діаграма розсіювання покаже "хмару" точок, витягнуту зліва направо вниз, що свідчить про обернений зв’язок. Розрахунок коефіцієнта Пірсона дасть значення, близьке до -0,94, що є дуже сильним зв’язком. Коефіцієнт Спірмена буде близьким до -0,96, підтверджуючи результат. Перевірка за критерієм Стьюдента покаже, що обидва коефіцієнти є статистично значущими. Таким чином, ми можемо з високою впевненістю стверджувати, що існує сильний обернений зв’язок між відвідуванням занять та успішністю.

Дисперсійний аналіз (ANOVA): виявлення впливу факторів

Дисперсійний аналіз (ANOVA) використовується в ситуаціях, коли ми маємо одну кількісну залежну змінну (наприклад, швидкість розв’язання задачі) та одну або кілька категоріальних незалежних змінних, які називаються факторами (наприклад, методика навчання). Мета ANOVA – визначити, чи впливає фактор на залежну змінну, тобто чи є статистично значущою відмінність між середніми значеннями залежної змінної в групах, сформованих різними рівнями фактора.

Основна ідея методу полягає в розкладанні загальної дисперсії залежної змінної на дві частини: дисперсію, зумовлену впливом фактора (міжгрупова), та дисперсію, зумовлену випадковими причинами (внутрішньогрупова). Якщо міжгрупова дисперсія значно перевищує внутрішньогрупову, це свідчить про те, що фактор дійсно впливає на результат. Для порівняння цих дисперсій використовується F-критерій Фішера. Чим більше розраховане значення F-критерію, тим більша ймовірність того, що вплив фактора є значущим. Сучасні програмні засоби, такі як Excel та MatLab, окрім самого значення F, обчислюють ще й p-значення (ймовірність помилки). Якщо p-значення менше за обраний рівень значущості (зазвичай 0,05), то вплив фактора вважається статистично значущим.

Розглянемо приклад: трьом групам студентів пред’являли слова з різною швидкістю (низькою, середньою, високою), а потім вимірювали показник їх відтворення. Середні значення в групах відрізняються. Але чи є ця відмінність випадковою, чи вона зумовлена саме швидкістю пред’явлення? Проведемо однофакторний дисперсійний аналіз. У MS Excel це можна зробити як "вручну", розраховуючи всі суми квадратів і дисперсії, так і за допомогою інструменту "Однофакторний дисперсійний аналіз" з пакету "Аналіз даних". Обидва способи дають однаковий результат: F-критерій значно перевищує критичне значення, а p-значення є дуже малим. Отже, ми з високою ймовірністю (понад 99%) можемо стверджувати, що швидкість пред’явлення слів суттєво впливає на показник їх відтворення.

У MatLab для проведення однофакторного дисперсійного аналізу використовується функція anova1(). Вона не тільки повертає p-значення, але й будує діаграму розмаху (boxplot), яка наочно демонструє медіани та розкид значень у кожній групі. Результати аналізу виводяться у вигляді стандартної

таблиці ANOVA, де містяться всі необхідні компоненти: суми квадратів, ступені свободи, середні квадрати, F-статистика та p-значення. Якщо p-значення перевищує 0,05, ми не маємо підстав відхилити нульову гіпотезу, і вплив фактора вважається статистично незначущим.

Таким чином, таблиці спряженості, кореляційний та дисперсійний аналізи є потужним арсеналом методів, які дозволяють дослідникові перейти від простого опису даних до виявлення глибинних зв'язків і закономірностей, що лежать в їх основі. Кожен з цих методів має свою сферу застосування, і правильний вибір інструменту є запорукою отримання достовірних та змістовних результатів.

Питання для обговорення

1. Яке основне призначення середовища MatLab? Що означає його назва і на чому базуються обчислення?
2. Опишіть структуру головного вікна MatLab. Які функції виконують вікна Command Window, Workspace та Current Folder?
3. Як створити змінну в MatLab? Яких правил слід дотримуватися при присвоєнні імен? Яке ім'я отримує змінна, якщо його не вказати?
4. Поясніть призначення файлів з розширеннями .mat, .m та .asv у середовищі MatLab.
5. Як зберегти змінні поточної сесії у файл та як їх відновити при наступному сеансі роботи? Наведіть приклади команд.
6. Як створити матрицю в MatLab? Поясніть, як отримати доступ до окремого елемента, рядка або стовпця матриці.
7. У чому полягає різниця між файлом-сценарієм (Script) та файлом-функцією (Function) у MatLab? Які області дії змінних у кожному з них?
8. Для чого використовуються графічні засоби MatLab? Як побудувати простий графік функції та теплову карту (heatmap)?
9. Яка мета нормалізації та стандартизації даних? У яких випадках застосування цих процедур є необхідним?
10. Поясніть суть min-max нормалізації та z-стандартизації. Як ці перетворення впливають на набір даних?

Лекція 14
Регресійний аналіз даних. Лінійна регресія

Вступ до регресійного аналізу

Регресійний аналіз є одним із фундаментальних методів статистики та інтелектуального аналізу даних. Його основна мета – вивчення залежності між однією або кількома незалежними змінними, які називають факторами, та однією залежною змінною, яку називають відгуком. Якщо кореляційний аналіз відповідає на питання "чи пов'язані змінні між собою?", то регресійний аналіз дозволяє відповісти на питання "як саме вони пов'язані?", тобто описати цей зв'язок у вигляді математичного рівняння. Це рівняння, яке називається рівнянням регресії, дає змогу прогнозувати значення залежної змінної для нових, невідомих раніше значень факторів.

Важливо розуміти різницю між регресією та класифікацією. В обох задачах ми прогнозуємо значення залежної змінної. Однак у класифікації залежна змінна є категоріальною, тобто вона може набувати лише кількох фіксованих значень, наприклад, "так" або "ні". У регресії ж залежна змінна є числовою і може набувати будь-яких значень у певному діапазоні, як-от ціна товару, обсяг продажів або температура повітря. Залежно від кількості факторів розрізняють просту, або парну, регресію, де вивчається вплив лише одного фактора, та множинну регресію, де досліджується вплив двох і більше факторів. За формою зв'язку виділяють лінійну та нелінійну регресію. Лінійна регресія, яка описує залежність за допомогою рівняння прямої лінії, є найпростішою, найбільш дослідженою та найширше вживаною, тому ми розпочнемо саме з неї.

Етапи побудови регресійної моделі

Процес регресійного аналізу, як і будь-яке серйозне дослідження, складається з кількох логічних та послідовних етапів. Перший етап – це постановка задачі та висунення гіпотез. На цьому етапі ми, спираючись на знання про предметну область, формулюємо припущення про те, які саме фактори можуть впливати на досліджуваний показник.

Другий етап – визначення та підготовка змінних. На цьому етапі важливо не лише чітко визначити, що буде залежною, а що – незалежними змінними, але й перевірити незалежні змінні на мультиколінеарність. Мультиколінеарність – це наявність сильного лінійного зв'язку між самими незалежними змінними. Якщо дві незалежні змінні дуже сильно корелюють між собою, вони несуть майже однакову інформацію, і включення обох у модель може призвести до спотворення результатів та ускладнення інтерпретації. Тому одну з таких змінних, як правило, вилучають з аналізу.

Третій етап – вибір форми моделі. На основі візуального аналізу діаграми розсіювання, теоретичних міркувань або простого перебору кількох варіантів обирається тип функції, яка, ймовірно, найкраще описуватиме залежність. Це

може бути лінійна, логарифмічна, поліноміальна, експоненціальна та інші функції.

Четвертий, ключовий етап – оцінка параметрів обраної моделі. Найпоширенішим методом для цього є метод найменших квадратів. Його суть полягає у знаходженні таких значень коефіцієнтів рівняння, за яких сума квадратів відхилень фактичних значень залежної змінної від значень, розрахованих за моделлю, є мінімальною. Іншими словами, ми шукаємо таку лінію (або криву), яка проходить максимально близько до всіх точок на діаграмі розсіювання.

П’ятий етап – оцінка якості побудованої моделі. Цей етап є надзвичайно важливим і включає в себе кілька процедур. По-перше, це перевірка правильності моделі, яка підтверджує, що обчислення проведено без помилок. По-друге, це оцінка точності апроксимації, тобто того, наскільки добре модель описує наявні дані. По-третє, це перевірка адекватності, або статистичної значущості моделі в цілому. І по-четверте, це перевірка значущості окремих коефіцієнтів моделі. Якщо якість моделі визнана незадовільною, ми повертаємося до попередніх етапів і, можливо, обираємо іншу форму моделі або інший набір факторів.

Шостий етап – інтерпретація отриманих результатів. Ми повинні не просто отримати числа, але й зрозуміти їхній зміст у контексті досліджуваної задачі. Нарешті, сьомий етап – застосування побудованої моделі для прогнозування нових значень залежної змінної.

Проста лінійна регресія: рівняння та метод найменших квадратів

Найпростішим випадком є проста лінійна регресія, яка описує зв’язок між двома змінними. Рівняння такої регресії має вигляд рівняння прямої лінії. У цьому рівнянні є два важливі параметри. Перший – це вільний член, який показує, яким буде значення залежної змінної, коли незалежна змінна дорівнює нулю. Другий, і найголовніший – це коефіцієнт регресії. Він показує, на скільки одиниць у середньому зміниться залежна змінна при зміні незалежної змінної на одну одиницю. Якщо цей коефіцієнт додатний, зв’язок є прямим (із зростанням фактора зростає і відгук), якщо від’ємний – оберненим.

Як уже зазначалося, для знаходження цих коефіцієнтів використовується метод найменших квадратів. Він мінімізує суму квадратів вертикальних відстаней від точок даних до побудованої прямої. Це гарантує, що знайдена пряма буде найкращою з точки зору цього критерію.

Оцінка якості регресійної моделі

Після того, як ми отримали рівняння регресії та обчислили його коефіцієнти, необхідно ретельно оцінити, наскільки добре воно описує дані. Для цього існує кілька ключових показників.

Перший крок – це перевірка правильності побудови моделі. Вона базується на тому, що загальну мінливість (варіацію) залежної змінної можна розкласти на дві частини: частину, яка пояснюється впливом включених у модель

факторів, і частину, яка залишається непоясненою і зумовлена впливом випадкових, неврахованих факторів. Суми квадратів відхилень, що відповідають цим двом частинам, пов'язані простим співвідношенням. Його виконання є ознакою того, що всі проміжні обчислення проведені правильно.

Другий, надзвичайно важливий показник – це коефіцієнт детермінації. Він показує, яку частку загальної варіації залежної змінної вдалося пояснити за допомогою побудованої моделі. Значення коефіцієнта детермінації завжди знаходиться в межах від 0 до 1. Чим ближче це значення до 1, тим точніше модель описує дані. Наприклад, якщо коефіцієнт детермінації дорівнює 0,96, це означає, що 96% змін залежної змінної зумовлені впливом включених у модель факторів, і лише 4% – іншими, неврахованими причинами. Це дуже високий показник.

Третій етап – перевірка адекватності моделі, тобто її статистичної значущості в цілому. Для цього використовують F-критерій. Він перевіряє гіпотезу про те, що насправді ніякого зв'язку між факторами та відгуком немає, і знайдений нами коефіцієнт детермінації є випадковим. Якщо розраховане значення F-критерію перевищує певне критичне значення, ця гіпотеза відхиляється, і модель визнається статистично значущою, тобто адекватною.

Четвертий етап – перевірка значущості окремих коефіцієнтів регресії. Для цього використовують t-критерій. Він перевіряє гіпотезу про те, що певний коефіцієнт, скажімо, коефіцієнт при першому факторі, насправді дорівнює нулю, тобто що цей фактор насправді не впливає на результат. Якщо t-критерій є значущим (його значення перевищує критичне), ми можемо зробити висновок, що цей фактор дійсно має вплив і повинен бути залишений у моделі.

І нарешті, тісноту лінійного зв'язку між фактором та відгуком оцінює коефіцієнт кореляції Пірсона. Його квадрат для випадку простої лінійної регресії дорівнює коефіцієнту детермінації. Значення коефіцієнта кореляції, близьке до 1 за модулем, свідчить про дуже сильний зв'язок.

Практична реалізація в MS Excel

Розглянемо застосування цих теоретичних знань на конкретному прикладі. Нехай ми маємо дані про обсяг виробництва (це наш фактор) та сумарні виробничі витрати (це залежна змінна) для десяти підприємств. Наше завдання – побудувати лінійну регресійну модель залежності витрат від обсягу виробництва.

Перший крок – візуалізація. Будуємо діаграму розсіювання. Точки на графіку розташовуються вздовж певної лінії, що дає підстави припустити наявність лінійного зв'язку.

Другий крок – обчислення параметрів моделі. В Excel це можна зробити кількома способами. Найпростіший – скористатися функцією ЛИНЕЙН. Вона повертає масив, який містить коефіцієнти рівняння регресії. В результаті ми отримуємо конкретне рівняння, яке пов'язує витрати з обсягом виробництва.

Третій крок – оцінка якості. Ми обчислюємо за отриманим рівнянням теоретичні значення витрат для кожного підприємства і порівнюємо їх з

фактичними. На основі цих порівнянь знаходимо суми квадратів відхилень. Переконуємося, що основне варіаційне рівняння виконується. Далі обчислюємо коефіцієнт детермінації. Він виявляється дуже високим, близьким до 0,96. Це означає, що майже 96% варіації витрат пояснюється обсягом виробництва, що свідчить про високу точність моделі. Коефіцієнт кореляції також є дуже високим, підтверджуючи тісний зв'язок.

Для перевірки адекватності обчислюємо F-критерій. Його значення виявляється набагато більшим за критичне, що дозволяє зробити висновок про статистичну значущість, тобто адекватність моделі. Перевірка коефіцієнтів за t-критерієм також показує, що обидва параметри рівняння є значущими.

В Excel існує й інший, ще зручніший спосіб – спеціалізований інструмент "Регресія", який входить до пакету "Аналіз даних". Він автоматично виконує всі необхідні обчислення і видає результат у вигляді готової, добре структурованої таблиці. Ця таблиця містить усі показники якості, які ми щойно розглянули: множинний R, R-квадрат (коефіцієнт детермінації), скоригований R-квадрат, стандартну помилку, результати дисперсійного аналізу з F-статистикою та її значущістю, а також таблицю з коефіцієнтами рівняння, їх стандартними помилками, t-статистиками та p-значеннями. Це надзвичайно зручний та потужний інструмент для швидкого та всебічного аналізу.

Крім лінійної, Excel дозволяє легко будувати й інші регресійні моделі, наприклад, експоненціальну, логарифмічну, поліноміальну або степеневу. Це робиться за допомогою додавання лінії тренду на графік. Додавши на діаграму лінію тренду, можна одразу побачити рівняння обраної моделі та відповідний коефіцієнт детермінації. Це дозволяє легко порівняти різні моделі між собою та обрати ту, яка має найвищий коефіцієнт детермінації, тобто найкраще описує дані. У нашому прикладі поліноміальна модель другого порядку має коефіцієнт детермінації 0,99, що свідчить про її дещо вищу точність порівняно з лінійною.

Регресійний аналіз у SPSS

Статистичний пакет SPSS також надає широкі можливості для регресійного аналізу. Для побудови лінійної моделі в меню обираємо пункти "Аналіз", потім "Регресія" і далі "Лінійна". У діалоговому вікні, що відкривається, ми вказуємо, яка змінна є залежною, а які – незалежними. У налаштуваннях статистик можна вибрати додаткові параметри, такі як довірчі інтервали для коефіцієнтів, описові статистики, матрицю коваріацій тощо. Результати, як і в Excel, містять таблицю з коефіцієнтом детермінації, результатами дисперсійного аналізу (F-критерій) та таблицю коефіцієнтів з їхніми t-статистиками та p-значеннями.

Для порівняння різних типів моделей у SPSS існує спеціальний інструмент "Підгонка кривих" (в меню "Аналіз" -> "Регресія" -> "Підгонка кривих"). Він дозволяє одночасно побудувати кілька моделей (лінійну, логарифмічну, квадратичну, експоненціальну тощо) на одному графіку. Для кожної моделі програма надає таблицю з результатами дисперсійного аналізу, значеннями

коефіцієнтів та, найголовніше, коефіцієнтом детермінації. Це дає змогу легко та наочно порівняти якість різних моделей і обрати найкращу.

Таким чином, регресійний аналіз є надзвичайно потужним інструментом для моделювання залежностей та прогнозування. Розуміння логіки його проведення, основних показників якості та вміння застосовувати його за допомогою сучасних програмних засобів є невід'ємною складовою професійної компетенції фахівця з інтелектуального аналізу даних.

Питання для обговорення

1. Яке основне призначення середовища MatLab? Що означає його назва і на чому базуються обчислення?
2. Опишіть структуру головного вікна MatLab. Які функції виконують вікна Command Window, Workspace та Current Folder?
3. Як створити змінну в MatLab? Яких правил слід дотримуватися при присвоєнні імен? Яке ім'я отримує змінна, якщо його не вказати?
4. Поясніть призначення файлів з розширеннями .mat, .m та .asv у середовищі MatLab.
5. Як зберегти змінні поточної сесії у файл та як їх відновити при наступному сеансі роботи? Наведіть приклади команд.
6. Як створити матрицю в MatLab? Поясніть, як отримати доступ до окремого елемента, рядка або стовпця матриці.
7. У чому полягає різниця між файлом-сценарієм (Script) та файлом-функцією (Function) у MatLab? Які області дії змінних у кожному з них?
8. Для чого використовуються графічні засоби MatLab? Як побудувати простий графік функції та теплову карту (heatmap)?
9. Яка мета нормалізації та стандартизації даних? У яких випадках застосування цих процедур є необхідним?
10. Поясніть суть min-max нормалізації та z-стандартизації. Як ці перетворення впливають на набір даних?

Лекція 15
Факторний аналіз даних

Що таке факторний аналіз і для чого він потрібен?

Уявіть, що ми маємо набір даних, який описує студентів за допомогою семи різних характеристик: швидкість виконання завдань, поведінка, активність, кількість відсутностей, уважність, впевненість та мотивація. Аналізувати всі ці сім змінних окремо може бути складно. Крім того, деякі з них, ймовірно, тісно пов'язані між собою і насправді вимірюють якусь одну, глибшу характеристику. Наприклад, активність, швидкість і впевненість можуть бути різними проявами такого прихованого фактора, як "впевненість у собі".

Саме для вирішення таких задач і призначений факторний аналіз. Це сукупність методів, які дозволяють описати зв'язки між великою кількістю спостережуваних змінних за допомогою меншої кількості прихованих, безпосередньо не вимірюваних змінних, які називаються факторами. Головна мета факторного аналізу – "стиснути" інформацію, зменшити розмірність простору ознак, об'єднавши тісно корелюючі змінні в один узагальнений фактор. Це робить подальший аналіз простішим, а структуру даних – більш зрозумілою.

Фактори, які ми отримуємо в результаті, є кількісним вираженням цих прихованих, латентних змінних. Вони інтерпретуються як внутрішні, суттєві характеристики об'єктів дослідження, що є причиною взаємопов'язаних значень вихідних, вимірюваних змінних.

Розрізняють два основні типи факторного аналізу. Дослідницький факторний аналіз застосовується тоді, коли ми не маємо попередніх гіпотез про структуру даних і хочемо її виявити. Підтверджуючий факторний аналіз, навпаки, використовується для перевірки вже існуючої гіпотези про те, що дані мають певну факторну структуру.

Основні поняття: навантаження, ваги, спільність

Результати факторного аналізу подаються у вигляді двох ключових матриць. Перша – це матриця факторних навантажень. Її елементи, які називаються факторними навантаженнями, показують тісноту зв'язку між кожною вихідною змінною та кожним виявленим фактором. По суті, це коефіцієнти кореляції. Чим більше за модулем значення навантаження, тим сильніший зв'язок. Додатне значення вказує на прямий зв'язок (зі зростанням фактора зростає і значення змінної), від'ємне – на обернений. Аналізуючи цю матрицю, ми для кожного фактора визначаємо ті змінні, які з ним найтісніше пов'язані, і на основі цього даємо фактору змістовну назву.

Друга важлива матриця – це матриця факторних ваг. Її елементи, факторні ваги, – це значення виявлених факторів для кожного окремого об'єкта дослідження. Вони показують, наскільки сильно в кожному об'єкті проявляються властивості, що відповідають даному фактору. Ці нові змінні (фактори) можна зберегти і використовувати в подальшому аналізі замість великої кількості вихідних.

У факторному аналізі також важливо розрізняти загальні фактори, які впливають на всі або багато змінних, та характерні фактори, які є специфічними лише для однієї змінної. Частина дисперсії змінної, яка пояснюється загальними факторами, називається спільністю. Наша мета – максимізувати спільність, тобто досягти того, щоб виявлені загальні фактори пояснювали якомога більшу частину варіації кожної змінної.

Метод головних компонент

Найпоширенішим методом факторного аналізу є метод головних компонент. Його основна ідея полягає в лінійному перетворенні вихідного простору ознак у новий простір меншої розмірності. Нові осі (головні

компоненти) обираються таким чином, щоб вони були взаємно перпендикулярними (незалежними) і щоб перша компонента пояснювала максимально можливу частку загальної дисперсії даних, друга – максимально можливу частку з тієї, що залишилася, і так далі.

Процес починається зі стандартизації вихідних змінних, щоб усі вони мали однаковий масштаб. Потім будується кореляційна матриця, яка показує зв'язки між усіма парами змінних. На основі цієї матриці знаходяться її характеристичні числа (власні значення) та відповідні їм власні вектори. Характеристичне число для кожної компоненти показує, яку частку загальної дисперсії вона пояснює. Чим більше число, тим важливіша компонента.

Далі ми обираємо оптимальну кількість компонент. Для цього використовують кілька критеріїв. По-перше, залишають лише ті компоненти, чиє характеристичне число більше за одиницю (критерій Кайзера). По-друге, аналізують графік "кам'янистого осипу", де характеристичні числа зображені в порядку спадання. Оптимальна кількість компонент відповідає точці перелому на цьому графіку, після якої спадання сповільнюється.

Обертання факторів та інтерпретація

Початкова матриця факторних навантажень часто буває важкою для інтерпретації, оскільки багато змінних можуть мати середні навантаження за кількома факторами. Для полегшення інтерпретації застосовують процедуру обертання факторів. Геометрично це означає поворот системи координат у просторі факторів. Мета обертання – зробити матрицю навантажень більш контрастною: збільшити навантаження за одними факторами і зменшити за іншими, наближаючись до так званої "простої структури", де кожна змінна має високе навантаження лише за одним фактором.

Найпоширенішим методом ортогонального обертання є метод Варімакс. Він максимізує дисперсію квадратів навантажень, роблячи великі навантаження ще більшими, а малі – ще меншими. Після обертання ми отримуємо повернуту матрицю компонент, яку вже можна інтерпретувати.

Інтерпретація полягає в тому, щоб для кожного фактора визначити ті змінні, з якими він має найбільші за модулем навантаження, і на основі їхнього змісту дати фактору назву. Наприклад, якщо фактор має високі навантаження за змінними "швидкість", "активні відповіді" та "впевненість", ми можемо назвати його "впевненість у собі".

Практична реалізація факторного аналізу в SPSS

Розглянемо, як застосувати ці теоретичні знання на практиці в пакеті SPSS. Нехай ми маємо дані про поведінку 20 студентів, оцінену за сьома характеристиками (швидкість, поведінка, активність, відсутність, уважність, впевненість, мотивація). Наше завдання – виявити фактори, що лежать в основі цих спостережуваних змінних.

У меню SPSS обираємо "Аналіз" -> "Зменшення розмірності" -> "Факторний аналіз". У діалоговому вікні переносимо всі сім змінних у поле "Змінні". Далі ми маємо налаштувати параметри аналізу.

Описові статистики: Натискаємо кнопку "Описові". Тут ми відмічаємо "Початкове рішення", а в розділі "Кореляційна матриця" – "Коефіцієнти", а також "КМО і критерій сферичності Бартлетта". Ці критерії дозволяють оцінити, чи придатні наші дані для факторного аналізу.

Критерій КМО (Кайзера-Мейєра-Олкіна) оцінює, наскільки повно кореляції між змінними можна пояснити іншими змінними. Значення має бути більшим за 0,5. У нашому випадку воно становить 0,584, що є прийнятним.

Критерій сферичності Бартлетта перевіряє гіпотезу про те, що кореляційна матриця є одиничною (тобто зв'язку між змінними немає). Значущість цього критерію (p-значення) має бути меншою за 0,05. У нашому випадку воно дорівнює 0,000, що підтверджує наявність кореляцій і доцільність факторного аналізу.

Виділення факторів: Натискаємо кнопку "Виділення". Ми залишаємо метод "Головні компоненти". У полі "Виділити" обираємо "На основі власних значень більше 1". Це доручить програмі залишити лише ті компоненти, чиє характеристичне число перевищує 1. Також відмічаємо опцію виведення "Графік кам'янистого осипу".

Обертання: Натискаємо кнопку "Обертання" і обираємо метод "Варимакс". Це забезпечить нам ортогональне обертання для отримання більш інтерпретованої матриці. Відмічаємо "Повернуте рішення" та "Графіки навантажень".

Значення факторів: Натискаємо кнопку "Значення" і відмічаємо "Зберегти як змінні". Це додасть до нашого файлу даних нові змінні, які будуть містити факторні ваги для кожного студента.

Після всіх налаштувань натискаємо "ОК" для запуску аналізу.

Аналіз результатів факторного аналізу в SPSS

Результати виводяться у вікні переглядача. Першою ми бачимо таблицю з критеріями КМО та Бартлетта, яка підтверджує придатність даних. Далі йде таблиця "Пояснена сукупна дисперсія". У ній ми бачимо, що лише дві перші компоненти мають характеристичні числа більші за 1 (перша – 2,964, друга – 2,837). Разом вони пояснюють майже 83% загальної дисперсії вихідних змінних. Це чудовий результат: ми зменшили кількість змінних із 7 до 2, втративши лише 17% інформації. Графік кам'янистого осипу також чітко показує точку перелому після другої компоненти, підтверджуючи, що оптимальною є двофакторна модель.

Найважливішою для інтерпретації є таблиця "Повернута матриця компонент". Вона показує факторні навантаження після обертання Варимакс. Аналізуючи її, ми бачимо, що з першим фактором найтісніше пов'язані змінні "Поведінка", "Уважність" та "Мотивація". Причому зв'язок із "Поведінкою" – обернений (від'ємне значення), що може означати, що студенти, які ретельніше

розбиралися із завданнями, отримали нижчі бали за цією шкалою. Цей фактор можна назвати "Старанність" або "Зосередженість на завданні".

З другим фактором найбільші навантаження мають змінні "Швидкість", "Активність" та "Впевненість". Цей фактор явно відображає "Впевненість у собі" або "Швидкість реагування". Змінна "Відсутність" має слабкі навантаження за обома факторами, що свідчить про її специфічність; можливо, її варто було б виключити з аналізу.

Графік навантажень у повернутому просторі наочно ілюструє ці зв'язки, розміщуючи змінні близько до осей відповідних факторів.

Нарешті, у вікні даних з'явилися дві нові змінні, які містять факторні ваги для кожного студента. Тепер ми можемо використовувати ці два фактори замість семи вихідних у подальшому аналізі, наприклад, у кластеризації або регресії.

Таким чином, факторний аналіз є потужним інструментом, який дозволяє не тільки зменшити розмірність даних, але й глибше зрозуміти їхню внутрішню структуру, виявивши приховані закономірності та взаємозв'язки.

Питання для обговорення

1. Яке основне призначення середовища MatLab? Що означає його назва і на чому базуються обчислення?
2. Опишіть структуру головного вікна MatLab. Які функції виконують вікна Command Window, Workspace та Current Folder?
3. Як створити змінну в MatLab? Яких правил слід дотримуватися при присвоєнні імен? Яке ім'я отримує змінна, якщо його не вказати?
4. Поясніть призначення файлів з розширеннями .mat, .m та .asv у середовищі MatLab.
5. Як зберегти змінні поточної сесії у файл та як їх відновити при наступному сеансі роботи? Наведіть приклади команд.
6. Як створити матрицю в MatLab? Поясніть, як отримати доступ до окремого елемента, рядка або стовпця матриці.
7. У чому полягає різниця між файлом-сценарієм (Script) та файлом-функцією (Function) у MatLab? Які області дії змінних у кожному з них?
8. Для чого використовуються графічні засоби MatLab? Як побудувати простий графік функції та теплову карту (heatmap)?
9. Яка мета нормалізації та стандартизації даних? У яких випадках застосування цих процедур є необхідним?
10. Поясніть суть min-max нормалізації та z-стандартизації. Як ці перетворення впливають на набір даних?

ПІСЛЯМОВА

Ось ми й дісталися фінальної сторінки нашого посібника. Попереду залишилося п’ятнадцять лекцій, які провели нас довгим і, сподіваємося, захопливим шляхом – від першого знайомства з поняттям «шаблон» до створення власних нейронних мереж та факторних моделей. Цей шлях не був простим, адже інтелектуальний аналіз даних є однією з найскладніших, але водночас і найцікавіших дисциплін сучасної комп’ютерної науки.

Коли ми розпочинали цю роботу, нашою головною метою було створити не просто черговий підручник із сухою теорією, а практично орієнтований посібник, який стане для вас надійним провідником у реальному світі аналізу даних. Ми прагнули, щоб кожна лекція давала вам не лише знання, але й розуміння того, як застосувати ці знання на практиці – в MatLab, SPSS чи MS Excel. Судити про те, наскільки нам це вдалося, звісно, вам.

Протягом нашої подорожі ми розібрали фундаментальні поняття та найважливіші методи Data Mining. Ви навчилися готувати дані до аналізу, розуміти природу різних шкал вимірювання та долати проблеми пропусків і викидів. Ви опанували мистецтво кластеризації, навчившись знаходити приховані групи там, де їх не видно неозброєним оком. Ви занурилися у світ класифікації, озброївшись цілим арсеналом методів – від простого правила 1R до складних ансамблів дерев рішень. Ви навчилися прогнозувати майбутнє, аналізуючи часові ряди, та виявляти приховані зв’язки між подіями за допомогою асоціативних правил. І, нарешті, ви доторкнулися до найсучасніших технологій – нейронних мереж та факторного аналізу, які відкривають безмежні можливості для дослідження складних систем.

Однак важливо пам’ятати, що будь-який посібник, навіть найкращий, – це лише відправна точка. Справжнє опанування професії аналітика даних відбувається не в аудиторії, а в процесі розв’язання реальних задач, коли ви стикаєтеся з неструктурованими даними, нечіткими вимогами та обмеженнями в часі. Саме тоді вам знадобляться не лише знання алгоритмів, але й розвинене критичне мислення, інтуїція та, найголовніше, наполегливість. Не бійтеся помилятися – кожна помилка в аналізі даних, кожен неправильно побудований прогноз – це безцінний досвід, який робить вас кращими фахівцями.

Світ даних невпинно змінюється: з’являються нові алгоритми, зростають обчислювальні потужності, виникають нові виклики. Те, що сьогодні здається передовим краєм науки, завтра стане рутинним інструментом. Тому найважливіше, що ви маєте винести з цього курсу, – це не запам’ятовування конкретних формул чи налаштувань програм, а розуміння фундаментальних принципів, логіки мислення аналітика та, найголовніше, здатність навчатися самостійно.

Сподіваємося, що цей посібник став для вас не просто підручником, а вірним супутником, який допоміг зробити перші, найважчі кроки в професії. Ми щиро бажаємо вам цікавих задач, несподіваних відкриттів, чистих даних та високої точності прогнозів.

Бажаємо успіхів, натхнення та нових звершень!

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Болюбаш Н. М. Інтелектуальний аналіз даних : навч. посіб. Миколаїв : Вид-во ЧНУ ім. Петра Могили, 2023. 320 с.
2. Гороховатський В. О., Творошенко І. С. Методи інтелектуального аналізу та оброблення даних : навч. посіб. / М-во освіти і науки України ; Харків. нац. ун-т радіоелектроніки. Харків : ХНУРЕ, 2021. 92 с. URL: https://openarchive.nure.ua/handle/document/15868
3. Іванчук Я. В., Месюра В. І., Яровий А. А., Манжілевський О. Д. Інтелектуальний аналіз даних та машинне навчання. Частина 1. Базові методи та засоби аналізу даних : навчальний посібник. Вінниця : ВНТУ, 2021. 69 с.
4. Ланде Д. В., Субач І. Ю., Бояринова Ю. Є. Основи теорії і практики інтелектуального аналізу даних у сфері кібербезпеки : навчальний посібник. Київ : ІСЗЗІ КПІ ім. Ігоря Сікорського», 2018. 300 с. URL: https://ela.kpi.ua/server/api/core/bitstreams/5eb4cb50-3ef0-44d1-b35f-a80eade1554b/content
5. Литвин В. В., Пасічник В. В., Нікольський Ю. В. Аналіз даних та знань. Львів : Магнолія 2006, 2015. 276 с.
6. Лупан І. В. Інтелектуальний аналіз даних Data Mining : навчально-методичний посібник. Кропивницький : ФОП Піскова М. А., 2022. 112 с. URL: https://dspace.cusu.edu.ua/server/api/core/bitstreams/f8bbe456-1b5e-47b6-a80f-0a20b5d17264/content
7. Мельников О. С. Інтелектуальний аналіз даних : навч.-метод. посібник / Нац. техн. ун-т "Харків. політехн. ін-т". Харків : Impress, 2023. 196 с. URL: https://repository.kpi.kharkov.ua/handle/KhPI-Press/72877
8. Мельников О. С. Інтелектуальний аналіз даних : навч.-метод. посібник / Нац. техн. ун-т "Харків. політехн. ін-т". Харків : Impress, 2023. 196 с. URL: https://repository.kpi.kharkov.ua/handle/KhPI-Press/72877
9. Сергеєв-Горчинський О. О., Іщенко Г. В. Інтелектуальний аналіз даних: Комп’ютерний практикум : навч. посіб. для студ. спеціальності 122 «Комп’ютерні науки та інформаційні технології», спеціалізацій «Інформаційні системи та технології проектування», «Системне проектування сервісів» / КПІ ім. Ігоря Сікорського. Київ : КПІ ім. Ігоря Сікорського, 2018. 73 с.
10. Талах М. В., Дворжак В. В. Інтелектуальний аналіз даних. Частина 1. Чернівці : Технодрук, 2022. 367 с.

Навчальне видання

ІНТЕЛЕКТУАЛЬНИЙ АНАЛІЗ ДАНИХ

Конспект лекцій

Укладачі: **Пархоменко** Олександр Юрійович

Тищенко Світлана Іванівна

Ємельянов Святослав Ігорович

Жебко Олександр Олегович

Богатєнкова Олександра Євгенівна

Формат 60х84 1/16. Ум. друк. арк. 4,0.
Тираж 20 прим. Зам. № ____

Надруковано у видавничому відділі
Миколаївського національного аграрного університету
54008, м. Миколаїв, вул. Георгія Гонгадзе, 9

Свідоцтво суб'єкта видавничої справи ДК № 4490 від 20.02.2013 р.