

О.М. Верес, В.Л. Мельник, Л.Б. Чирун
Національний університет “Львівська політехніка”,
кафедра інформаційних систем та мереж

ЗАСТОСУВАННЯ MS SQL SERVER 2005 ДЛЯ ПОБУДОВИ ІНТЕЛЕКТУАЛЬНОЇ СКЛАДОВОЇ ІНФОРМАЦІЙНОЇ СИСТЕМИ

© Верес О.М., Мельник В.Л., Чирун Л.Б., 2008

Розглянуто принципи інтелектуального аналізу даних за допомогою методів Data Mining, а зокрема використання дерев рішень.

In the given article main principles of intellectual data analysis using Data Mining methods (Decision Trees) are discussed.

Постановка проблеми у загальному вигляді

Системи баз даних досягли великого успіху протягом останніх двох десятиліть. За останні роки реляційні бази даних накопичили величезні обсяги даних у різних галузях людської діяльності. Аналізувати ці дані “вручну” стало надзвичайно важко. З кожним днем все більше даних збираються та накопичуються в базах даних. Пошук корисної інформації став центром уваги багатьох підприємств [1–4]. Все більшу увагу привертає до себе Data Mining як ключовий компонент аналізу інформації. Моделі аналізу призначені для відповіді на складні запитання, які ви можете задавати постійно, при цьому відсутня необхідність побудови складних застосувань для знаходження прихованих залежностей.

Знаходження прихованих закономірностей в даних, взаємозв'язків між різними змінними в базах даних, моделювання і вивчення складних систем на основі історії їх поведінки – ось предмет і завдання Data Mining. Результати Data Mining — емпіричні моделі, класифікаційні правила, виділені кластери і т.д. — можна потім інкорпорувати в існуючі системи підтримки прийняття рішень та використовувати для прогнозу майбутніх ситуацій.

Data Mining – це процес виявлення в “сирих” даних раніше не відомих нетривіальних, практично корисних і доступних для інтерпретації знань, необхідних для прийняття рішень у різних сферах людської діяльності [2, 5, 6]. Data Mining є одним з кроків видобування знань з баз даних.

Алгоритми, які використовуються в Data Mining, вимагають великої кількості обчислень. Раніше це було стримуючим чинником широкого практичного застосування Data Mining, проте сьгоднішнє зростання продуктивності сучасних процесорів зняло гостроту цієї проблеми. Тепер за прийнятний час можна провести якісний аналіз сотень тисяч та мільйонів записів.

Ця технологія застосовується у всіх ділових секціях, зокрема банківській справі, телекомунікаціях, промисловості, маркетингу та електронній комерції.

Завдання, що вирішуються методами Data Mining:

- **Класифікація** – це встановлення приналежності об'єктів (спостережень, подій) до одного із заздалегідь відомих класів.

- **Регресія** (зокрема, завдання прогнозування) – це встановлення залежності від вхідних змінних безперервних вихідних.

- **Кластеризація** – це групування об'єктів (спостережень, подій) на основі даних (властивостей), що описують суть цих об'єктів. Об'єкти усередині кластера повинні бути “схожими” один на одного та відрізнятися від об'єктів, що увійшли до інших кластерів. Чим більше схожі об'єкти усередині кластера і чим більше відмінностей між кластерами, тим точніша кластеризація.

- **Асоціація** – це виявлення закономірностей між зв'язаними подіями. Прикладом такої закономірності є правило, яке вказує, що з події X впливає подія Y. Такі правила називаються асоціативними. Вперше це завдання було запропоноване для знаходження типових шаблонів

покупок, що здійснюються в супермаркетах, тому іноді її ще називають аналізом ринкової корзини (market basket analysis).

- **Послідовні шаблони** – це встановлення закономірностей між зв'язаними в часі подіями, тобто виявлення залежності, що якщо відбудеться подія X, то через заданий час відбудеться подія Y.
- **Аналіз відхилень** – це виявлення найнехарактерніших шаблонів.

Аналіз останніх досліджень і публікацій

Ситуація нагадує ринок СУБД четверть століття тому до офіційного прийняття реляційної парадигми та SQL [8–12]. Свої дослідження в цьому напрямку проводили і проводять такі гіганти ринку баз даних, як Microsoft, Oracle та IBM. Вони розробили і стандартизували свої засоби для аналізу даних. Сьогодні існує два галузевих стандарти – PMML (Predictive Model Markup Language) і CRISP-DM.

Стандарт PMML, визначений Data Mining Group (dmg.org): SAS, SPSS, IBM, Microsoft, Oracle, тощо і є XML-форматом зберігання моделі для найпоширеніших алгоритмів. PMML не є програмним інтерфейсом для Data Mining, а фокусується на описі змісту моделі: словник даних, схема отримання, трансформація полів, статистика тощо.

Стандарт CRISP-DM є результатом зусиль SPSS (тоді ISL), NCR і Daimler Chrysler, фактично, – це методологія. Він описує життєвий цикл проекту Data Mining (послідовність фаз, складові задач, вхід і вихід кожної), не вдаючись до конкретних методик видобування.

Найпоширеніші стандарти «великої трійки» виробників СУБД, а саме: Microsoft, Oracle, IBM, що є закономірним, якщо враховувати обсяги даних, інсталяції і суттєві інвестиції в Data Mining. «Стандарти» кожного відрізняються один від одного, хоча переважно обслуговують одні й ті самі потреби. ISO SQL/Multimedia (SQL MM) – потокові багатофрагментні розширення SQL на область повнотексту, геопростору, мультимедіа тощо. Секція з Data Mining введена в цей стандарт на вимогу IBM, концепція і синтаксис дуже близькі до DMX. Java Data Mining API (JSR-73) – Java-пакет, що дозволяє Java-застосуванням взаємодіяти з Data Mining-засобом. Стандарт підтримувався і любивався Oracle, програмна модель дуже нагадує C# і AMO. Стандарти OLE DB for Data Mining та XML/A розробляються та впроваджуються Microsoft за підтримки Hyperion, SAS, Angoss, KXEN, Megaputer.

Термін Data Mining останнім часом зустрічається часто. Це пов'язано насамперед із посиленням інтересом до цієї теми з боку підприємств малого та середнього бізнесу, а не тільки вузького кола фахівців, як це було кілька років тому.

Не вирішені раніше частини загальної проблеми. Проте в реаліях російського та українського ринку підприємство часто не має можливості придбати окреме застосування цього типу. По-перше, ціни на такі застосування «кусаються» – вони можуть доходити до декількох тисяч доларів залежно від класу застосування та його функціональних можливостей. По-друге, потрібно також витратити засоби на навчання персоналу для роботи з новим інструментом. Все це у поєднанні з природною недовірою до нових розробок відлякує потенційних клієнтів таких систем.

Зрозуміло, багато хто вважав би за краще використовувати одне застосування, яке містило б всі функції, пов'язані із зберіганням, обробкою і видобуванням даних. Таким універсальним засобом є добре знайомий більшості підприємств пакет Microsoft SQL Server.

Побудова інтелектуальної системи дає можливість виключення суб'єктивного підходу до визначення якісного складу науково-педагогічного персоналу кафедр, постійного доступу керівництва навчального закладу до інформації за кадровим складом, проведення інтелектуального аналізу кадрового складу довільної складності.

Цілі (завдання) статті

Основним завданням статті є визначення множини типів джерел даних та алгоритмів інтелектуального аналізу даних для побудови інтелектуальної складової підсистеми формування та аналізу кадрового забезпечення інформаційної системи кафедри. Метою роботи є дослідження методів та засобів побудови моделі видобування даних в SQL Server.

Основний матеріал дослідження

Ми живемо в століття загальної інформатизації. Важко переоцінити значення даних, які ми безперервно збираємо в процесі нашої діяльності, в управлінні бізнесом або виробництвом, в банківській справі, в розв'язанні наукових, інженерних і медичних завдань. Могутні комп'ютерні

системи, в яких зберігаються величезні бази даних, а також їхні керівники стали невід'ємним атрибутом життєдіяльності як великих корпорацій, так і невеликих компаній. Проте наявність даних ще не є достатньою для покращання показників роботи. Потрібно вміти трансформувати «сирі» дані в корисну для прийняття важливих бізнес-рішень інформацію. У цьому і полягає основне призначення технологій Data Mining [2, 7–9].

Microsoft SQL Server (інструментарій Analysis Services, який входить до його складу) отримав власні засоби видобування даних тільки в 2000 році, у межах корпорації Microsoft стратегії BIA, що реалізовується (Business Internet Analysis – аналітика електронної комерції) [14–16]. Мета стратегії – надання компаніям, що займаються електронною комерцією, можливості збирання й аналізу даних про поведінку клієнтів інтерактивних магазинів.

Зрозуміло, що така вузька спеціалізація обмежує функціональність та області застосування засобів SQL Server як засобів видобування даних.

У Microsoft SQL Server 2005 Data Mining як технологія бізнес-аналізу даних отримала подальший розвиток. Вона дає змогу будувати складне аналітичне рішення у вигляді моделі.

Завдання, яке було поставлене при розробленні моделей аналізу – це створити застосування, яке:

- легке у використанні;
- забезпечує повний набір функціональних можливостей;
- легко вбудовується в застосування;
- щільно інтегрується в SQL Server BI технологію;
- розширює ринок продажів для цих застосувань.

Модель – це основа видобування даних в SQL Server. По суті, модель є сукупністю метаданих, що відображають деякі правила і закономірності у початкових даних. При цьому структура моделі визначає набір ключових атрибутів аналізу, тоді як її зміст несе безпосередньо статистичну інформацію – тут простежується схожість з ідеологією звичайних таблиць. Проте варто мати на увазі, що на основі одного і того самого набору початкових даних можна побудувати декілька різних моделей. У цьому сенсі побудова правильної моделі гарантує нам отримання саме тих «прихованих» залежностей, які ми прагнемо виявити. За те, як виконуватиметься аналіз даних, відповідає алгоритм аналізу.

Всі утиліти аналізу даних, включаючи Microsoft SQL Server 2005 Analysis Services, використовують безліч алгоритмів. Використання готових алгоритмів спрощує роботу із створення застосування, хоча за допомогою аналітичного сервера і мов програмування можна створити і свої власні моделі.

Процес побудови моделі реалізований в Analysis Services у вигляді майстра, що дає змогу крок за кроком задавати параметри моделі і виконувати її обробку, що, на думку розробників, спрощує проведення аналізу.

Перший крок у побудові моделі – вибір джерела даних для аналізу. Підтримуються два типи джерел даних: багатовимірні, що використовуються у межах технології OLAP, і звичайні – реляційні. Наявність першого варіанта дає набагато більшу свободу вибору для аналізу, адже далеко не кожне підприємство має власне багатовимірне сховище даних.

На відміну від традиційних реляційних СУБД, концепція OLAP не так широко відома, хоча загадковий термін «куби OLAP» чули, напевно, майже всі.

OLAP — це не окремо взятий програмний продукт, не мова програмування і навіть не конкретна технологія. Якщо намагатися охопити OLAP у всіх його проявах, то це сукупність концепцій, принципів та вимог, на яких ґрунтуються програмні продукти, що полегшують аналітикам доступ до даних.

Аналітики — це особливі споживачі корпоративної інформації. Завдання аналітика — знаходити закономірності у великих масивах даних. Аналітикові потрібно багато даних, які є вибірковыми та мають характер «набір атрибутів — число». Останнє означає, що аналітик працює з таблицями.

Концепція OLAP з'явилась саме для вирішення подібних проблем. Куби OLAP є, по суті, мета-звітами. Розрізаючи мета-звіти (тобто куби) за вимірюваннями, аналітик отримує «звичайні» двомірні звіти – такі, які його цікавлять (це не обов'язково звіти у звичайному розумінні цього терміна – йдеться про структури даних з такими самими функціями). Переваги кубів очевидні – дані необхідно отримати з реляційної СУБД тільки один раз, при побудові куба. Оскільки

аналітики, як правило, працюють з інформацією, яка не змінюється «на льоту», сформований куб є актуальним протягом достатньо тривалого часу. Завдяки цьому не тільки виключаються перебої в роботі сервера реляційної СУБД (немає запитів з тисячами і мільйонами рядків відповідей), але й різко підвищується швидкість доступу до даних для самого аналітика. Крім того, як вже наголошувалося, продуктивність підвищується і внаслідок підрахунку проміжних сум ієрархій та інших агрегованих значень у момент побудови куба.

Робота з OLAP-системами може ґрунтуватися на двох описаних нижче схемах.

Для «легких» застосувань підійдуть OLAP-засоби, вбудовані в настільні застосування. Такі засоби, як правило, мають безліч обмежень: на кількість вимірювань, на допустимі ієрархії тощо. До подібних засобів, наприклад, належить модуль Pivot Table, що дає змогу працювати з кубами в Microsoft Excel. Pivot Table входить в Microsoft Office і донедавна був єдиним OLAP-продуктом в його складі. У цьому випадку дані видобуваються модулем-клієнтом безпосередньо з реляційної СУБД.

У «важких» випадках застосовують двоступеневу схему «клієнт–сервер». Сервер забезпечує безпосередньо видобування інформації з СУБД і решту всіх дій, необхідних для створення кубів. Спеціалізоване застосування «клієнт» призначене для зручного (а головне — ефективного) перегляду кубів і виявлення тих самих аналітичних закономірностей, з яких ми починали. Серед продуктів Microsoft серверна частина представлена Microsoft Analysis Services, які входять в MS SQL Server.

Основними складовими елементами OLAP є:

- розмірності;
- куби;
- аналітичні моделі.

Всі ці та інші компоненти пов'язуються один з одним за допомогою засобів розроблення.

Кінцевою метою використання OLAP є аналіз даних і подання результатів цього аналізу у вигляді, зручному для сприйняття й ухвалення рішень. Основна ідея OLAP полягає в побудові багатовимірних кубів, доступних для запитів користувача. Проте початкові дані для побудови OLAP-кубів зазвичай зберігаються в реляційних базах даних. Нерідко це спеціалізовані реляційні бази даних, так звані сховища даних (Data Warehouse). На відміну від так званих оперативних баз даних, з якими працюють застосування, що модифікують дані, сховища даних призначені винятково для обробки й аналізу інформації, тому проектується вони так, щоб час виконання запитів до них був мінімальним. Зазвичай дані копіюються в сховищі з оперативних баз відповідно до певного розкладу.

Типова структура сховища даних істотно відрізняється від структури звичайної реляційної СУБД. Як правило, ця структура денормалізована (це дає змогу підвищити швидкість виконання запитів), тому може допускати надмірність даних.

Основними складовими структури сховищ даних є таблиця фактів (fact table) і таблиці вимірювань (dimension tables).

Після вибору джерела можна безпосередньо формувати структуру моделі. Для цього потрібно визначити таблицю (або вимірювання у разі багатовимірного джерела), що містить аналізовані дані, а також вибрати одне з полів таблиці (або показник багатовимірного куба), яке знаходитиметься у фокусі дослідження.

Вибір початкових даних і предмета аналізу – процес творчий, тому, якщо не вдалося отримати необхідні оцінки відразу, необхідно спробувати змінити структуру моделі, ввівши в неї додаткові атрибути. Можливо, це дасть змогу оцінити ситуацію з іншого погляду.

Наступний важливий крок – вибір алгоритму аналізу даних. Як вже зазначалось вище, Analysis Services підтримує декілька алгоритмів [7, 9, 16].

Оскільки галузі застосування і результати роботи кожного з них можуть сильно відрізнятися, є сенс зупинитися докладніше на цьому кроці.

Алгоритм Microsoft Decision Trees ґрунтується на відомому методі побудови дерев рішень. У його межах значення кожного з досліджуваних атрибутів класифікується на основі значень решти атрибутів з використанням правил вигляду «якщо — то». Результат роботи такого алгоритму — деревоподібна структура, кожен вузол якої є якимось питанням. Щоб вирішити, до якого класу віднести деякий об'єкт або ситуацію, потрібно відповісти на питання, що стоять у вузлах цього дерева, починаючи з його кореня.

Алгоритм **Microsoft Decision Trees** будує модель, створюючи різні зрізи даних, які називають також вузлами дерева. Він додаватиме вузли кожного разу, коли вхідний стовпець (input column), тобто стовпець, який аналізують, буде значною мірою пов'язаний із стовпцем, що передбачається (predicable column). Спосіб поділу залежить від статистики та структури даних.

Алгоритм Microsoft Clustering. Цей алгоритм використовує інший, не менш відомий метод пошуку логічних закономірностей — метод «найближчого сусіда». У процесі роботи алгоритму початкові дані об'єднуються в групи (кластери) на основі аналогічних або схожих значень атрибутів. Отримані набори даних аналізуються, що дає змогу виявити приховані закономірності або побудувати ймовірнісний прогноз. За цим алгоритмом проводять глибший аналіз даних, ніж за деревом рішень, але і він має свої обмеження. Його переважно застосовують для наборів даних зі схожими атрибутами, значення яких належать певному інтервалу.

Цей алгоритм має принципову відмінність від Microsoft Decision Tree, яка полягає в тому, що для нього не потрібне значення, що передбачається (predicate column). Алгоритм виконує тренування моделі винятково на основі реляційних залежностей між даними, об'єднуючи їх у кластери однотипних значень. Його дію можна спрощено представити як виконання відносно множини даних реляційного оператора GROUP BY.

Алгоритм працює шляхом виявлення взаємозв'язків у даних і створення серії кластерів, що групуються по атрибутах цих взаємозв'язків. Після визначення першої групи кластерів він намагається виконати інші групування, проводячи послідовні ітерації за всіма групами, відшукуючи сукупність атрибутів, що дають кращий результат.

Microsoft Clustering пропонує два методи групування даних усередині кластера:

- **Expectation Maximization (EM)** – пошук максимального математичного очікування;
- **K-Means** – використання відстані між атрибутами і кластеризація за відстанню.

Для використання алгоритму необхідно задати ключовий стовпець і вхідні стовпці.

Алгоритм Association Rules. Алгоритм шукає у даних закономірності вигляду А і В та породжує закономірність вигляду С. Цей алгоритм особливо корисний при роботі з великими каталогами. Завдяки можливостям прогнозування алгоритм Association Rules особливо добре підходить для застосувань, що аналізують перехресні дані за продажами, зокрема при торгівлі через Web.

В алгоритмі використовуються поняття випадків (cases) і окремих предметів (items). Групування окремих предметів називається набором предметів. Алгоритм групує множини предметів у випадок.

Алгоритм використовує метод сканування набору даних, вишукуючи предмети, що з'являються разом і які представляють той або інший випадок. Кількість випадків і предметів, за якими виробляються асоціативні правила, визначається програмно за допомогою спеціальних параметрів. Правила, що виробляються за допомогою моделі, потім можуть використовуватися за призначенням.

Модель повинна містити ключовий стовпець, вхідні стовпці і один стовпець, що передбачається. Вхідні дані для моделі часто є декількома таблицями.

Алгоритм Naïve Bayes. Ця ймовірнісна модель корисна при класифікації і поглибленому дослідженні даних. Засіб перегляду виразно показує відмінності між двома станами вхідної змінної. Алгоритм забезпечує розрахунок умовної вірогідності між вхідними значеннями і значеннями, що передбачаються.

Алгоритм менше завантажує процесор і тому швидше забезпечує знаходження залежностей між вхідними значеннями, що передбачаються. Можна використовувати цей алгоритм для первинного аналізу даних і потім застосувати ці результати до інших моделей аналізу, які інтенсивніше використовують ресурси процесора.

З використанням цього алгоритму обчислюється вірогідність кожного стану для стовпців, що передбачаються. Після проведення розрахунків можна використовувати Microsoft Naïve Bayes Viewer в SQL Server Business Intelligence Development Studio, щоб візуально подати алгоритмом розподіл стану.

Алгоритм Sequence Clustering. Алгоритм Sequence Clustering поєднує прогнозування, що забезпечується алгоритмом кластеризації, з технологією побудови послідовностей. Послідовністю може

бути будь-яка група подій, пов'язаних з користувачем. Алгоритм знаходить найзагальніші послідовності, групуючи події разом, тобто при цьому формується шлях групування.

Цей алгоритм дуже схожий на Microsoft Clustering Algorithm, проте, на відміну від нього, він знаходить кластери випадків, які містять подібні шляхи в послідовності.

За допомогою цього алгоритму можна передбачити наступний кластер і його розташування в кластерній послідовності. Оскільки при роботі використовуються незв'язані значення в стовпцях, можна використовувати цей алгоритм для виявлення прихованих зв'язків між даними.

Алгоритм Time Series. Цей алгоритм, розроблений Microsoft, дає змогу аналізувати і прогнозувати будь-які дані, залежні від часу. За допомогою технології регресивних дерев цей алгоритм здатний виявляти закономірності у декількох послідовностях та бачити, як пов'язані між собою різні події.

Перевага алгоритму полягає в тому, що він забезпечує можливість крос-передбачення, тобто можливість обробки декількох тимчасових серій і на підставі залежностей між ними передбачити поведінку системи.

Алгоритм Neural Networks. Нейронні мережі – це клас аналітичних методів, що побудовані на (гіпотетичних) принципах навчання мислячих істот і функціонування мозку і які дають змогу прогнозувати значення деяких змінних у нових спостереженнях за даними інших спостережень (для цих же або інших змінних) після проходження етапу так званого навчання на наявних даних. Нейронні мережі є одним з методів «видобування» даних.

При застосуванні цих методів насамперед постає питання вибору конкретної архітектури мережі (кількості «шарів» та «нейронів» у кожному з них). Потім побудована мережа піддається процесу так званого «навчання». На цьому етапі нейрони мережі ітеративно обробляють вхідні дані і коригують свої ваги так, щоб мережа якнайкраще прогнозувала дані, на яких виконується «навчання». Після навчання на наявних даних мережа готова до роботи і може використовуватися для побудови прогнозів.

Це технологія штучного інтелекту, яка є найкращою для пошуку складних взаємозв'язків між даними, які не можна виявити іншими алгоритмами. Хоча закономірності, які він виявляє, можуть виявитися складними для пояснення. Neural Networks – найкращий на даний момент алгоритм пошуку нелінійних залежностей у даних. Оскільки цей алгоритм ретельно аналізує дані, то він працює повільніше за інші.

В аналітичному сервері корпорації Microsoft використовується алгоритм, що створює класифікації і виконує регресійний аналіз даних, створюючи перцептрон (perceptron), що є кібернетичним аналогом нейрона.

Перцептрон – це програмований елемент, що розпізнає, здатний налаштувати логіку своєї роботи під вхідні дані. Перцептрони об'єднані один з одним, але, крім того, кожен з них має зворотний зв'язок, завдяки якому він налагоджується під особливості даних, що надходять на його вхід.

Алгоритм Microsoft Neural Network дуже схожий на цей алгоритм, але має одну істотну відмінність. Алгоритм підраховує вірогідність для кожного вхідного стану, обробляючи в кожен момент часу один випадок і порівнюючи його зі значенням, що передбачається. Помилки, що виникають при порівнянні першого випадку, не відкидаються, а через петлю зворотного зв'язку надходять в нейронну мережу і використовуються на наступному кроці. Ще однією важливою відмінністю від Microsoft Decision Trees є те, що алгоритм може використовувати тривимірні вузли розгалуження усередині моделі і у зв'язку з цим ефективніше аналізувати вхідний потік даних.

Алгоритм Microsoft Neural Network не підтримує можливості занурення в дані (drill-through), точно так, як і не підтримує розмірностей, які створені на основі моделей аналізу. Пов'язане це з тим, що структура вузлів моделі не має однозначної залежності від вхідних даних і може бути нелінійною залежністю з дуже малим рівнем кореляції.

В основу нейронної мережі, що використовується в алгоритмі, покладено багаторівневу перцептронну мережу (Multilayer Perceptron network), яку також іноді називають мережею зі зворотним дельта-зв'язком за помилками навчання (Back-Propagated Delta Rule network).

Вибір алгоритму. Вибір правильного алгоритму залежить від класу завдання, яке потрібно вирішити, а також від складу початкових даних. Завдання класифікації неоднорідних даних краще вирішувати за допомогою алгоритму дерев рішень, а завдання прогнозування або виявлення

невних закономірностей — за допомогою методу кластеризації. Який би алгоритм ви не обрали, на цьому побудова моделі закінчена, і можна переходити до наступного процесу — тренування моделі. Також на вибір моделі аналізу може впливати об'єм даних, які опрацьовуються, оскільки деякі алгоритми (Neural Networks) дуже вимогливі до обчислювальних ресурсів.

Тренування побудованої моделі — це не що інше, як процес обробки початкових даних згідно з обраним алгоритмом. Цей процес може зайняти тривалий час, особливо при великих обсягах даних. Після закінчення тренування початкові дані більше вам не знадобляться. В результаті тренування модель буде заповнена статистичними даними, які можуть бути представлені як в графічному, так і в цифровому вигляді.

Дерева рішень в SQL Server 2005

Розглянемо застосування засобів MS SQL Server 2005 для інтелектуального аналізу складу науково-педагогічних працівників кафедри університету [16].

Одним з найпростіших і разом з тим ефективних методів Data Mining є дерева рішень. Вони є простими для сприйняття, оскільки описують знайдені знання у вигляді правил “якщо ... то ...”. Сфера застосування дерева рішень сьогодні широка, але всі завдання, що вирішуються цим апаратом, можуть бути об'єднані у три класи.

- **Опис даних.** Дерева рішень дають змогу зберегти дані у компактній формі, замість них ми можемо зберегти дерево рішень, яке містить точний опис об'єктів.

- **Класифікація.** Дерева рішень вирішують завдання класифікації, тобто віднесення об'єктів до одного із заздалегідь відомих класів. Цільова змінна повинна мати дискретні значення.

- **Регресія.** Якщо цільова змінна має безперервні значення, дерева рішень дають змогу встановити залежність цільової змінної від незалежних(вхідних) змінних. Наприклад, до цього класу належать завдання числового прогнозування(прогнози значень цільової змінної).

Сьогодні існує значна кількість алгоритмів, що реалізують дерева рішень CART, C4.5, Newid, Itrule, CHAID, Cn2 тощо. Але найпоширенішими є два:

- **CART (Classification and Regression Tree)** – це алгоритм побудови бінарного дерева рішень. Кожен вузол дерева при розбитті має тільки двох нащадків. Як видно з назви алгоритму, він вирішує завдання класифікації і регресії;

- **C4.5** – алгоритм побудови дерева рішень, причому кількість нащадків вузла не обмежена. Не вміє працювати з безперервним цільовим полем, тому він вирішує тільки завдання класифікації.

У SQL Server використовується гібридний алгоритм дерев рішень, який є розробкою Microsoft Research. За цим алгоритмом розв'язують задачі класифікації, регресії та асоціації.

Принцип роботи цього алгоритму найкраще продемонструвати за допомогою прикладу. Вхідний набір даних містить інформацію про 28 викладачів кафедри вищого навчального закладу (ВНЗ). Зокрема тут містяться дані про стать, науковий ступінь, стаж роботи, наявність публікації з грифом Міністерства освіти і науки, а також займану посаду.

Властивістю для передбачення є посада працівника.

Першим кроком алгоритму є побудова таблиці кореляції значень. Кожна колонка в цій таблиці – пара атрибут/значення вхідних властивостей. Кожен рядок – стан значення передбачуваної властивості. Комірки таблиці – це величина кореляції значень вхідної властивості та передбачених станів.

Таблиця 1

Таблиця кореляції значень

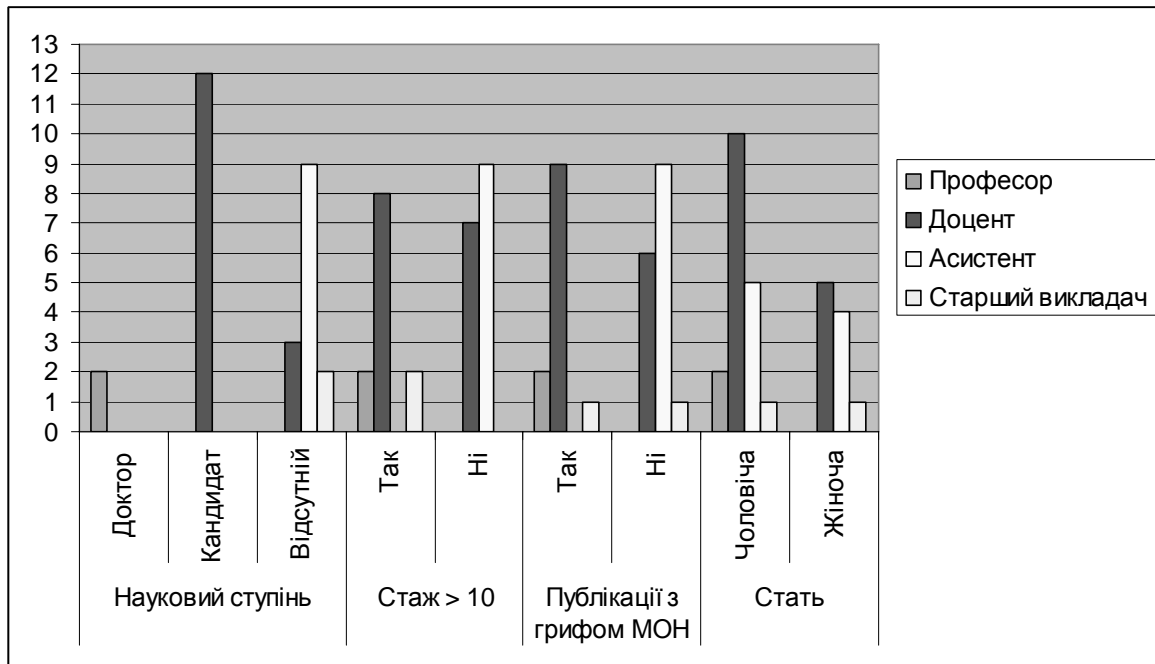
№ з/п	Посада	Науковий ступінь			Стаж > 10		Публікації з грифом МОН		Стать	
		доктор	кандидат	відсутній	так	ні	так	ні	чол.	жін.
1	2	3	4	5	6	7	8	9	10	11
1.	Професор	2	0	0	2	0	2	0	2	0
2.	Доцент	0	12	3	8	7	9	6	10	5
3.	Асистент	0	0	9	0	9	0	9	5	4
4.	Старший викладач	0	0	2	2	0	1	1	1	1

Алгоритм дерев рішень спочатку вибирає атрибут для розподілу на кореневі елементи. Критерій поділу вибирається так, щоб підмножини після розділення були дуже різні в розумінні критерію передбачення. Як видно з діаграми, таким критерієм буде Науковий ступінь (див. рисунок). Але для комп'ютера простіше працювати з числами, а не зображеннями. Для вимірювання цих величин ми можемо застосувати деякий формальний критерій. Таким критерієм, наприклад, може бути ентропія.

Для обрахунку застосовується відповідна математична формула

$$E(p_1, p_2, \dots, p_n) = \sum -p_i \log_2 p_i,$$

де p_i – ймовірність i -го стану передбачуваної величини.



Таблиця кореляції значень у вигляді діаграми

Найменше значення ентропії має атрибут Науковий ступінь, який і є найзначущим атрибутом, за яким буде здійснено розподіл на кореневі елементи. Цей процес повторюється для кожного листка дерева.

Таблиця 2

Таблиця кореляції значень з обчисленою ентропією

№ з/п	Посада	Науковий ступінь			Стаж > 10		Публікації з грифом МОН		Стать	
		доктор	кандидат	відсутній	так	ні	так	ні	чол.	жін.
1	2	3	4	5	6	7	8	9	10	11
1.	Професор	2	0	0	2	0	2	0	2	0
2.	Доцент	0	12	3	8	7	9	6	10	5
3.	Асистент	0	0	9	0	9	0	9	5	4
4.	Старший викладач	0	0	2	2	0	1	1	1	1
5.	Нормалізовані кількості	1,00	0,00	0,00	0,17	0,00	0,17	0,00	0,11	0,00
		0,00	1,00	0,21	0,66	0,44	0,75	0,38	0,55	0,50
		0,00	0,00	0,65	0,00	0,56	0,00	0,56	0,28	0,40
		0,00	0,00	0,14	0,17	0,00	0,08	0,06	0,06	0,10
		0,00	0,00	1,27	1,26	0,99	1,04	1,24	1,58	1,36
6.	Ентропія	0,00	0,00	1,27	1,26	0,99	1,04	1,24	1,58	1,36
7.	Сумарна ентропія	1,27			2,25		2,28		2,94	

У цьому прикладі було розглянуто тільки задачу класифікації з використанням атрибутів з малою кількістю станів.

Цей алгоритм також вміє будувати дерева з використанням атрибутів з великою кількістю станів, але рекомендована кількість – менша за 10. Якщо станів більше, можна застосувати групування станів.

Висновки

Результати аналізу показали, що SQL Server 2005 містить стандартні алгоритми математики і статистики, відкриті в різні часи та відшліфовані людством, починаючи з середини XVII ст., а саме:

- наївний Байєс;
- дерева рішень;
- часові ряди;
- асоціативні правила;
- послідовності та кластеризація;
- нейронні мережі;
- нечіткий пошук тексту.

Використання SQL Server 2005 Data Mining дає змогу після опрацювання великих обсягів даних знайти в них закономірності та на цій підставі створити прогноз поведінки системи при виникненні таких умов. При цьому немає потреби писати складні математичні системи, оскільки про це вже поклопоталася корпорація Microsoft, яка реалізувала так звані моделі аналізу даних у готовому вигляді.

Подальші дослідження будуть спрямовані на розроблення та опис множини алгоритмів аналізу даних і побудові на їхньому ґрунті моделей для інтелектуального аналізу даних із застосуванням різних типів джерел даних підсистеми формування та аналізу кадрового складу кафедри.

1. Inmon W.H. *Building the Data Warehouse*. John Wiley, 1996. 2. Fayyad U.M., Piatetsky-Shapiro G., Smyth P., Uthurusamy R. *Advances in Knowledge Discovery and Data Mining*. AAAI/MIT Press, 1996. 3. Kimball R. *The Data Warehouse Toolkit*. John Wiley & Sons, New York, 1996. 4. Андрейчиков А.В. *Интеллектуальные информационные системы* / А.В. Андрейчиков, О. Н. Андрейчикова. – М.: Финансы и статистика, 2004. – 424 с. 5. Баймаков А.И. *Интеллектуальные информационные технологии: Учеб. пособие* / А.И. Баймаков, И. А. Баймаков. – М.: МГТУ им. Н.Э. Баумана, 2005. – 304 с. 6. Гаврилова Т. *Базы знаний интеллектуальных систем: Учебник для вузов* / Т. Гаврилова, В. Хорошевский. – СПб.: Питер, 2000. – 384 с. 7. Вороновский Г.А. *Генетические алгоритмы, искусственные нейронные сети и проблемы виртуальной реальности* / Вороновский Г.А., Махотило К.В., Петрашев С.Н., Сергеев С.А.. – Харьков: Основа, 1997. – 306 с. 8. Девятков В.В. *Системы искусственного интеллекта. Серия: Информатика в техническом университете* / Девятков В.В. – М. : Изд-во МГТУ им. Н.Э. Баумана, 2001. – 352 с. 9. Дюк В. *Data mining: Учебный курс* / В. Дюк, А. Самойленко. – СПб.: Питер, 2001. – 368 с. 10. Гарсиа-Молина Г. *Системы баз данных: Полный курс; Пер. с англ.* / Г. Гарсиа-Молина, Дж. Ульман, Дж. Уидом. — М.: Издательский дом „Вильямс”, 2003. — 1088 с. 11. Дейт К. Дж. *Введение в системы баз данных* / Дейт К. Дж. — 8-е изд. Ппер. с англ. — М.: Издательский дом „Вильямс”, 2005. — 1328 с. 12. Конноли Т. *Базы данных : проектирование, реализация и сопровождение. Теория и практика: Учеб. пособие* / Т. Конноли, К. Бегг. — 3-е изд. Пер. с англ. — М. : Издательский дом „Вильямс”, 2003. — 1440 с. 13. Ревунков Г.И. *Базы и банки данных и знаний : учеб. для вузов* / Г.И. Ревунков, Э.Н. Самохвалов, В.В. Чистов; Под ред. В.Н. Четверикова. — М.: Высшая школа, 1992. — 367 с. 14. Тихомиров Ю. *Microsoft SQL Server 7.0 . Серия "В подлиннике"* / Ю. Тихомиров. – СПб. : БХВ-Петербург, 2001. – 720 с. 15. Оутей М. *Эффективная работа: SQL Server 2000* / М. Оутей, П. Конте. – СПб.: Питер, 2002. – 992 с. 16. Каленик А.И. *Использование новых возможностей Microsoft SQL Server 2005*. – М.: Русская Редакция; СПб.: Питер, 2006. – 334 с.