УДК 004.658: 652.3

М.Л. Дворецький

РЕАЛІЗАЦІЯ ЗАДАЧІ КЛАСИФІКАЦІЇ ЗАСОБАМИ МОВИ TRANSACT-SQL ПРИ ІНТЕЛЕКТУАЛЬНОМУ АНАЛІЗІ ДАНИХ

Загальна постановка проблеми та її зв'язок з науково-практичними задачами

Системи підтримки прийняття рішень – це системи, що використовуються для аналізу інформації. Подібні системи створюються на основі таких теорій, як дослідження операцій, теорія управління, а також методи статистичної обробки даних. Підтримка прийняття рішень є одним із застосувань технології баз даних. Серед прикладів такого застосування є сховища даних, оперативна аналітична обробка, багатовимірні бази даних та інтелектуальний аналіз даних (data mining) [1].

В основу сучасної технології Data mining покладена концепція шаблонів (паттернів), що відображають фрагменти багатоаспектних взаємовідношень в даних. Ці шаблони представляють собою закономірності, що властиві підвибіркам даних, які можуть бути компактно виражені у зрозумілій людині формі. Пошук шаблонів відбувається за допомогою методів, що не обмежені рамками апріорних припущень про структуру вибірки і вигляд розподілення значень аналізуємих показників. Важливе положення Data mining – нетривіальність розшукуваних шаблонів. Це означає, що знайдені шаблони мають відображати неочевидні, несподівані регулярності в даних, що складають так звані „приховані знання" [2].

Одною з класичних задач інтелектуального аналізу даних є задача класифікації, що зводиться до визначення класу об'єкта за його характеристиками. Необхідно відмітити, що множина класів, до яких може бути віднесений об'єкт, зазделегіть відома.

В задачі класифікації потребується визначити значення залежної змінної об'єкта на основі значень інших змінних, що характеризують даний об'єкт. Формально задачу класифікації можна записати наступним чином. Є множина об'єктів:

$$I=\{i_1, i_2, ..., i_j, ..., i_n\},$$

де i_n – досліджуваний об'єкт.

Кожен об'єкт характеризується набором змінних:

$$Ij=\{x1, x2, ..., xh, ..., xm, y\},$$

де x_h – незалежні змінні, значення яких відомі та на основі яких визначається значення залежної змінної y.

В Data mining часто набір залежних змінних позначають у вигляді вектора:

$$X=\{x1, x2, ..., xh, ..., xm \}.$$

Кожна змінна може приймати значення із деякої множини:

$$Ch = \{Ch1, Ch2, ... \}.$$

Якщо множина значень незалежної змінної кінцева, то задача має назву задача класифікації. Якщо змінна y приймає значення на множині дійсних чисел R, то задача називається задачею регресії [3].

Існують наступні основні види представлення залежності змінної y від незалежних змінних:

- Класифікаційні правила;
- Дерева рішень;
- Математичні функції.

Класифікаційні правила складаються з двох частин: умова та висновок:

Якщо (умова) то (висновок).

Умовою є перевірка однієї або декількох незалежних змінних. Перевірки декількох змінних можуть бути об'єднані за допомогою операцій „і", „або" та „ні". Висновком є значення залежної змінної або розподілення її ймовірності по класах.

Основною перевагою правил є легкість їх використання і представлення природною мовою. Друга перевага – їх відносна незалежність. В набір легко додати нове правило без необхідності змінювати вже існуючі [3].

Задача класифікації є актуальною для багатьох сфер діяльності. Прикладами таких задач можуть бути наступні задачі: рішення банківського службовця щодо надання клієнту кредиту; задача виявлення та відсіювання „спам" повідомлень в електронній пошті; розпізнавання образів, тощо [4]. У цій статті у якості прикладу буде розглянуто задачу виявлення потенціальних VIP-клієнтів торгівельного підприємства.

Оскільки однією з основних вимог до системи є велика навчаюча вибірка, бажана її інтеграція із потужними середовищами розробки баз даних, де вищезазначена вибірка може зберігатися. Але існуючі програмні рішення задач класифікації, розроблені у якості окремих модулів. Тому постає актуальною реалізація задачі класифікації засобами мови Transact-SQL (Structured Query Language), що дозволило б вбудувати підсистему інтелектуального аналізу майже у будь-яку реляційну СКБД.

Огляд публікацій та аналіз невирішених питань

Аналіз публікацій [2,3], присвячених задачам інтелектуального аналізу та, зокрема, реалізації задачі класифікації показує, що розроблені програмні продукти виконані у вигляді окремих програмних модулів, та передбачають розташування навчаючої вибірки на клієнтському місці та навантаження клієнтського місця обробкою великих об'ємів даних. Прикладами вищезазначених продуктів є такі системи, як See5/C5.0 [5], WizWhy [6], Xelopes [7] та ін.

Зовсім не приділяється увага створенню прикладної реалізації алгоритмів рішення задачі класифікації за допомогою операторів мови SQL-Transact, що б дозволило інтегрувати підсистему класифікації безпосередньо у систему керування баз даних (СКБД).

Мета досліджень

Метою є реалізація задачі класифікації засобами мови SQL-Transact у вигляді запитів та збережених процедур, та розробка засобів приведення задачі регресії до класичної задачі класифікації шляхом розподілення області допустимих значень незалежних та залежної змінних на інтервали.

Результати досліджень

Задача класифікації та регресії вирішуються в два етапи. На першому виділяється навчаюча вибірка. В неї входять об'єкти, для яких відомі значення як незалежних, так і залежної змінних. На основі навчаючої вибірки будується модель визначення значення залежної змінної. Її часто називають функцією класифікації або регресії [8]. Для отримання максимально точної функції до навчаючої функції висуваються наступні основні вимоги:

- Кількість об'єктів, що входять у вибірку, має бути достатньо великою. Чим більше об'єктів, тим побудована на її основі

функція класифікації або регресії буде точнішою;

- У вибірку мають входити об'єкти, що представляють всі можливі класи у випадку задачі класифікації або всю область значень у випадку задачі регресії;
- Для кожного класу в задачі класифікації або кожного інтервалу області значень в задачі регресії вибірка має містити достатню кількість об'єктів.

На другому етапі побудовану модель застосовують до аналізуємих об'єктів (до об'єктів із невизначеним значенням незалежної змінної).

Далі розглянемо реалізацію задачі класифікації засобами мови SQL-Transact на базі середовища розробки баз даних MS SQL Sever 2000.

Будемо вважати, що навчаюча вибірка об'єктів дослідження, представлених множиною $I_j=\{x_1, x_2, ..., x_h, ..., x_m, y\}$ (де x_h – незалежні змінні, значення яких відомі та на основі яких визначається значення залежної змінної y) фізично представлена у базі даних у вигляді окремої таблиці, або матеріалізованого представлення. Позначимо його **Study_Table**.

Крім того, у якості вхідних параметрів буде виступати кількість інтервалів, на які буде розбита область значень тієї чи іншої незалежної або залежної змінної. Вони будуть виражені множиною $K=\{Kx_1, Kx_2, ..., Kx_h, ..., Kx_m, Ky\}$.На рівні бази даних це буде таблиця **Param** із трьома полями: **Name** - назва поля таблиці Study_Table, **Kol** - кількість інтервалів, на яку розбивається область значень цього поля та **In_Out** – флаг, що позначає незалежну змінну.

Поставлена задача може буди розділена на наступні етапи:

- Реалізація розбиття діапазону значень змінних на деяку кількість інтервалів;
- Збереження відомостей про інтервали у таблиці **Param_Interval,** яка матиме наступні поля: Param – ім'я змінної, Num – номер інтервалу, Low_bord – нижня межа інтервалу, Upper_Bord – верхня межа інтервалу;
- Створення нової таблиці **Study_Table_Work,** в яку записати значення із таблиці **Study_Table** із заміною конкретний значень незалежних та залежної змінних на номери інтервалів (зведення задачі регресії до класичної задачі класифікації);

 ISSN 1562-9945

- Отримання набору правил, що визначають поведінку незалежної змінної по відношенню до залежних змінних у вигляді таблиці Prob_Table із наступними полями: Param – назва змінної, Num – номер інтервалу, Y_Num – номер інтервалу незалежної змінної, Prob – ймовірність виникнення ситуації.

Етапи реалізації задачі класифікації схематично зображені на рис. 1.

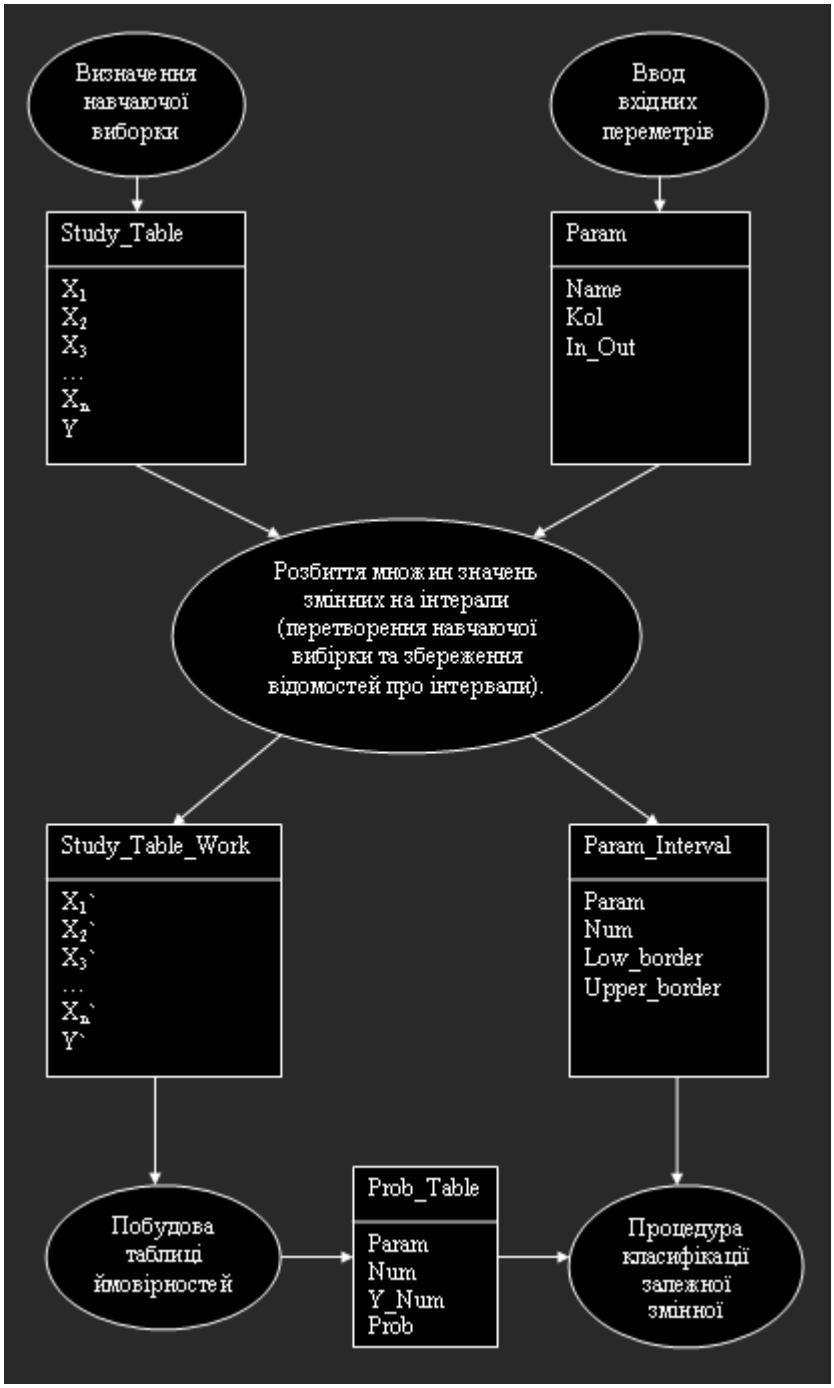

Рисунок 1 – Схема поетапної реалізації задачі класифікації.

Далі наведемо реалізацію запропонованого підходу рішення задачі класифікації у вигляді SQL-коду збережуваних процедур MS SQL Server 2000. Процедура вводу вхідних параметрів матиме таку структуру:

```
CREATE PROCEDURE In_Out_Param(@Param char(20), @In_Out char(1), @kol smallint)
AS
/*якщо змінна незалежна і вже є незалежна змінна у таблиці, змінюємо її на незалежну*/
if exists(select * from param where in_out='O') and @In_Out='O'
        update param set in_out='I' where in_out='O'
/*якщо вже є така, змінна, змінюємо значення кількості інтервалів*/
if exists(select * from param where ltrim(rtrim(name))=ltrim(rtrim(@param)))
        update param set kol=@kol, in_out=@in_out where ltrim(rtrim(name))=ltrim(rtrim(@param))
/*якщо такої змінної немає, вносимо відомості про неї*/
if not exists(select * from param where ltrim(rtrim(name))=ltrim(rtrim(@param)))
            insert into param(name, In_Out, kol) values(@Param, @In_Out, @kol)
GO
```

Далі наведемо код процедури, що відповідає за розподілення множини значень на інтервали із подальшим створенням таблиць, що містять відомості про інтервали та перетвореної навчаючої вибірки.

```
CREATE PROCEDURE Make_New_Tables
AS
declare @param as char(20)
declare @kol as smallint
declare @i as smallint
declare @Max_val as float
declare @Min_val as float
declare @low_bord as float
declare @upper_bord as float
declare @sql1 as varchar(1000)
declare @sql2 as varchar(1000)
/*видаляємо з бази даних таблицю Study_Table_Work  якщо вона вже існує*/
if exists (select * from dbo.sysobjects where id = object_id(N'[dbo].[Study_Table_Work]') and
OBJECTPROPERTY(id, N'IsUserTable') = 1)
drop table [dbo].[Study_Table_Work]
/*очищуємо таблицю Param_Interval */
delete from Param_Interval
/*заповнюємо таблицю Param_Interval */
declare ParamCursor cursor for select name, kol from param
open ParamCursor
fetch first from ParamCursor into @param, @kol
/*обходимо по черзі всі змінні*/
while @@fetch_status = 0
begin
    /*визначаємо область значень (верхню та нижню межу) змінної*/
    delete from tmp_table
```

```
        exec('insert into tmp_table select max('+@param+')+1 from study_table')
        select @max_val=tmp_val from tmp_table
        delete from tmp_table
        exec('insert into tmp_table select min('+@param+') from study_table')
        select @min_val=tmp_val from tmp_table

        set @i=1
        /*для кожного інтервала змінної*/
        while @i<=@kol
        begin
                /*визначаємо верхню та нижню межу інтарвала*/
                set @low_bord=@min_val+(@i-1)*(@max_val-@min_val)/@kol
                set @upper_bord=@min_val+@i*(@max_val-@min_val)/@kol
                insert into Param_Interval(param, num, low_bord, upper_bord) values(@param, @i,
@low_bord, @upper_bord)
                set @i=@i+1
        end
        fetch next from ParamCursor into @param, @kol
    end
    close ParamCursor
    /*заповнюємо структуру таблиці study_table */
    select * into study_table_work from study_table where 1=0
    /*формуємо запит для заповнення таблиці study_table даними*/
    set @sql1='select '
    set @sql2='into study_table_work from study_table s '
    set @i=1
    declare IntervalCursor cursor for select name from param
    open IntervalCursor
    fetch first from IntervalCursor into @param
    /*доповнюємо запит для кожної змінної та інтервалу*/
    while @@fetch_status = 0
    begin
        set @sql1=@sql1+' p'+@i+'.num as ' +@param +', '
        set @sql1=@sql1+ ' inner join Param_Interval p'+@i+'on s.'+@param+'=>p'+@i+'.low_bord
and s.'+@param+'<p'+@i+'.upper_bord'
        set @i=@i+1
        fetch next from IntervalCursor into @param
    end
    close IntervalCursor
    /*виконуємо запит на заповнення таблиці study_table даними*/
    exec(@sql1+@sql2)
    GO
```

Процедура, що формує таблицю правил, які описують функцію залежності залежної змінної від незалежних та обраховує ймовірність того чи іншого висновку на основі навчаючої вибірки.

Для обрахування ймовірності використаний алгоритм Naпve Bayes, що має в основі формулу Байєса. Назва Naпve (наївний) походить від наївного припущення, що всі розглянуті незалежні змінні не залежать одна від одної. У дійсності це не так, але на практиці алгоритм знаходить своє застосування і дає непогані результати.

Нижче наведемо реалізацію процедури мовою SQL.

```
CREATE PROCEDURE Make_Rules AS
declare @param as char(20)
declare @num as smallint
declare @sql1 as varchar(1000)
declare @num_y as smallint
declare @i as smallint
/*визначаємо кількість інтервалів для незалежної змінної*/
select @num_y=i.num from param p inner join param_interval i on p.name=i.param where p.in_out='O'
declare ParamCursor cursor for select distinct param, num from param_Interval
/*очищуємо таблицю правил*/
delete from prob_Table
open ParamCursor
/*цикл для кожної змінної та інтервала*/
fetch first from ParamCursor into @param, @num
while @@fetch_status = 0
begin
      set @i=1
      while @i<=@num_y
      begin
            /*формуємо запит для визначення ймовірності правила*/
            set @sql1='insert into prob_Table select @param,  '+@param+', @i, count(y) from
study_table_work where y=@i  group by' +@param
            /*записуємо правило у таблицю prob_Table */
            exec(@sql1)
      end
      fetch next from ParamCursor into @param, @num
end
GO
```

Ймовірність належності об'єкта до класу C_r при умові рівності його змінної x_h деякому значенню c_d^h визначаються за формулою:

$$P(x_h = c_d^h \mid y = c_r) = P(x_h = c_d^h \text{ і } y = c_r) / P(y = c_r),$$

тобто дорівнює кількості об'єктів у навчаючій вибірці, у яких $x_h = c_d^h$ і $y = c_r$ до кількості об'єктів, що відносяться до класу c_r.

Зараз розроблений метод проходить етап випробування для вирішення задачі класифікації клієнтів супермаркету. Задача полягає у визначенні, чи є клієнт супермаркету потенціальним VIP-клієнтом.

VIP-клієнтами вважалися ті 20% клієнтів, що приносять 80% прибутку торгівельного підприємства. Незалежними змінними виступали частота відвідувань, середня, максимальна та мінімальна сума покупок, час, який минув від першої покупки у супермаркеті та відношення суми безготівкових розрахунків до загальної суми покупок. Після отримання статистичних результатів планується подальше доопрацювання та вдосконалення системи.

Висновки та перспективи подальших досліджень

Реалізація методів розв'язання задачі класифікації засобами мови SQL-Transact у вигляді запитів та збережених процедур шляхом інтеграції із системами керування баз даних (СКБД) максимально спрощує механізм доступу до великих навчаючих вибірок, що є однією з основних вимог до системи класифікації. Запропонована реалізація задачі класифікації дозволяє шляхом незначної зміни SQL-запиту (враховуючи особливості мови SQL конкретної СКБД та особливості структури бази даних) інмплентувати їх майже у будь-яку реляційну СКБД. Отримані правила будуть поповнювати базу знань, для представлення якої у подальшому будуть розроблені відповідні засоби в рамках СКБД.

Сферою застосування підсистеми може бути будь яка система підтримки прийняття рішення, що стикається із необхідністю класифікації, починаючи із медицини і закінчуючи прогнозуванням результатів виборів.

Перспектива досліджень передбачається у реалізації мовою SQL розподілення множини значень змінних на інтервали залежно від закону розподілу таким чином, щоб у різні проміжки потрапляла пропорційна кількість елементів навчаючої вибірки. Це дозволило б уникнути нульової ймовірності для правил, що не увійшли в навчаючу вибірку. Крім того, планується реалізація засобами мови SQL інших класів задач інтелектуального аналізу даних.

ЛИТЕРАТУРА

1. Дейт К. Дж. Введение в системы баз данных. - Киев*Москва : Диалектика, 1998. -787с.
2. Дюк В., Самойленко А. Data Mining: учебный курс. –Спб: Питер, 2—1.- 368с.

3. Барсегян А.А., Куприянов М.С., Степаненко В.В., Холод И.И. Методы и модели анализа данных: OLAP и Data Mining. – Питербург, 2004. – 336 с.
4. Конноли Т., Бегг К., Страчан А. Базы данных : проектирование, реализация и сопровождение. Теория и практика, 2-е издание. – “Вильямс»” : Москва*Санкт-Петербург*Киев, 2000. – 1112с.
5. http://www.rulequest.com
6. http://www.wizsoft.com
7. http://www.prudsys.com/Produkte/Algorithmen/Xelopes/
8. Гарсиа-Молина Г., Ульман Дж.Д., Уидом Дж. Системы баз данных. Полный курс. – М.: Изд. дом «Вильямс», 2003. – 1088 с.

 ISSN 1562-9945