прогнозування з використанням многочлена Лагранжа давало похибку, на одну-шість одиниць більшу або меншу, ніж прогнозування з використанням тригонометричного многочлена. Тому, хоч многочлен Лагранжа і можна використовувати для значень, що знаходяться на однаковій відстані h один від одного (періодичні дані), але застосування конкретно періодичних многочленів, як в даному випадку, дає більш правдиві результати.

Висновки

Розглянуто проблеми комплексної автоматизації готелів. Метою статті було визначити основні напрямки в розвитку готельного бізнесу та можливі підходи до вирішення означених проблем. Було запропоновано новий підхід до вирішення проблеми планування заселеності готелів, який полягає в використанні чисолових методів для прогнозування відвідуваності готелю. Автор сподівається, що запропонований підхід знайде практичне застосування в готельному бізнесі.

Безперечно, стаття не охоплює всі можливі проблеми, з якими стикаються власники готелів, але було зроблено перші кроки для вирішення найбільш нагальних проблем.

1. *“Система Едельвейс”, http://www.el.sochi.biz/docs/Edelweiss_offer_full.doc. 2. “Система управління готелем – CENIUM”, http://www.torob.info/soft/hotel/cenium.htm. 3. Fidelio Front Office – "Фіделіо" система портьє, http://www.osp.ru/cw/2002/18/023_1_print.htm. 4. “Готельний бізнес сьогодні” , http://www.econ.pu.ru/edu/courses/socmn/socmn240.pdf. 5. "Гостиница и ресторан: бизнес и управление". – 2003. – № 3. 6. Волков Ю.Ф. Технология гостиничного обслуживания. – Ростов-на-Дону: "Феникс", 2003. 7. Монастырный П.И., Крылов В.И. Начала теории вычислительных метеодов. – Минск: Наука и техника, 1983.*

УДК 681.3

Т.І. Завалій
Національний університет “Львівська політехніка”,
кафедра інформаційних систем та мереж

ОЦІНКА ЯКОСТІ КЛАСИФІКАТОРІВ, ПОБУДОВАНИХ НА ОСНОВІ ПРАВИЛ, ВИВЕДЕНИХ З ТАБЛИЦІ МЕДИЧНИХ ДАНИХ ЗА ДОПОМОГОЮ ТЕХНОЛОГІЇ НЕТОЧНИХ МНОЖИН

Розглянуто методологію неточних множин для пошуку правил у таблицях даних та побудови класифікаторів нових об’єктів. Оцінюванюються такі класифікатори, наводяться результати аналізу таблиці медичних даних та застосування різних алгоритмів дискретизації.

This paper is devoted to application of rough sets for mining rules from data tables and construction of classifiers that can perform classification of new objects. The quality of such classifiers is evaluated. The results of medical data table analysis and application of different discretization algorythms are presented.

Постановка проблеми в загальному вигляді

Задача, результати розв’язання якої наведені у цій статті, належить до широкого класу задач видобування знань з баз даних (Knowledge discovery in databases, KDD) [1, 2], які останнім часом здобули широке визнання і практичне застосування. Дуже часто поряд з поняттям «видобування знань з баз даних» зустрічаються також такі терміни, як «машинне навчання» (Machine Learning) [3] та «інтелектуальний аналіз даних» (Data Mining) [4, 1, 2], який є складовою частиною KDD.

Поява такої галузі, як KDD та її подальший розвиток частково були спричинені вдосконаленням технологій зберігання даних, внаслідок чого люди зіткнулись з величезними потоками інформації у різних галузях своєї діяльності. Будь-яка установа (комерційна, медична, наукова тощо) тепер використовує програмні засоби, за допомогою яких реєструє всі дані про свою діяльність. Зрозуміло, що такі великі обсяги даних без відповідної обробки є здебільшого лише «мертвим вантажем» і не приносять користі. Для такої обробки спочатку використовувалися методи математичної статистики, однак вони виявилися корисними в основному лише для перевірки попередньо сформульованих гіпотез.

Натомість інтелектуальний аналіз дозволяє виявляти в даних нетривіальні шаблони та приховані закономірності, що використовуються для формування баз знань. Це досягається шляхом використання спеціальних методів, в яких не застовуються апріорно задані схеми розв'язування чи шаблони для шуканих результатів. За їх допомогою вирішуються задачі прогнозування, класифікації, розпізнання образів, сегментації баз даних, виявлення в даних «прихованих» знань, інтерпретації даних, встановлення асоціацій в базі даних тощо.

Інтелектуальний аналіз даних є складовою частиною процесу видобування знань з баз даних, яке полягає в побудові моделей знань на основі даних. Всі дані з джерела даних або лише їх частина надходять на обробку. Ці первинні дані подаються найчастіше у вигляді таблиці даних і попередньо обробляються. Лише після цього вони передаються безпосередньо алгоритму дослідження даних. Результатом роботи алгоритму є моделі, які подають приховані знання. Ці моделі надалі обробляються, інтерпретуються та оцінюються.

Поряд з традиційними технологіями Data mining, такими як дерева розв'язків [4], алгоритми евристичного пошуку [4], нейронні мережі [5, 6], генетичні алгоритми [7], останнім часом з'явилася технологія *неточних множин* (rough sets), яка базується на теорії, створеній Ж. Павлаком (Zdzislaw Pawlak) [8, 9, 10] на початку 1980-х років. Ця теорія застосовується при аналізі таблиць даних та розв'язанні задач класифікації. Такі таблиці отримуються експериментально та дозволяють будувати моделі з подальшим дослідженням цих моделей або їх застосуванням для побудови систем прийняття рішень.

Сьогодні методології неточних множин застосовуються доволі часто [11]. У першу чергу це аналіз даних в галузі медицини: аналіз медичних даних, аналіз і класифікація зображень, діагностування пацієнтів, підтримка рішень щодо лікування, мінімізація даних, аналіз факторів хвороби тощо. У галузі економіки та фінансів відомі застосуваня неточних множин для оцінки ризиків банкрутства, оцінки компаній, планування кредитної політики, виявлення схем поведінки покупців, аналіз факторів, що впливають на коливання на біржі, виявлення прогнозуючих правил, аналіз комерційних даних тощо. Крім цього, підхід на основі неточних можин застосовується при розробці неточних та розмитих (fuzzy) контролерів, обробці та аналізі зображень і звуку, аналізі даних і прогнозуванні в галузі екології. Відомі також застосування для розробки програмного забезпечення, хімічних і молекулярних досліджень, в галузі соціології.

Цілі статті

Цілями статті є використанню технології неточних множин для виведення правил з таблиці даних та подальшої оцінки якості класифікатора, побудованого на основі цих правил.

Основний матеріал

Відповідно до технології неточних множин, набір даних, що досліджується, подається у вигляді таблиці, кожен рядок якої може характеризувати пацієнта, подію чи будь-який об'єкт. Кожен стовпчик таблиці відповідає певній властивості об'єкта. Позначимо через $U=\{u_1,...,u_m\}$ непорожню скінченну множину об'єктів – рядків таблиці, а через $A=\{a_1,...,a_n\}$ – непорожню скінченну множину атрибутів, таких, що $a : U \to V_a$ для всіх $a \in A$. Множину V_a називають множиною значень атрибута, а саму таблицю називають *інформаційною системою.* Крім цього, до таблиці може входити *атрибут прийняття рішення* ***d***, значення якого відносять об'єкти до того чи

іншого класу. Така таблиця $\boldsymbol{A}$ з класифікуючим атрибутом $\boldsymbol{d}$ називається *системою прийняття рішень* (decision system)

$$\boldsymbol{A} = (\mathrm{U}, \mathrm{A} \vee \{\mathrm{d}\}).$$

У випадку великих наборів реальних даних (які здебільшого подають певну модель, предметну область, результати експериментів тощо) деяка частина цих даних може бути надмірною або суперечливою. Це можливо, зокрема, коли у двох різних об'єктів значення відповідних атрибутів збігаються, а значення атрибута прийняття рішення є різним, тобто ці об'єкти неможливо класифікувати. У такому випадку говорять, що ці елементи належать до так званої *неточної* (rough) *області*.

Наведемо приклад системи прийняття рішень.

Таблиця 1

Система прийняття рішень

	Y1	Y2	D
X1	0	1	1
X2	1	0	0
X3	1	0	1
X4	0	1	1

З табл. 1 видно, що елементи X2 та X3 належать до неточної області, оскільки вони мають однакові значення атрибутів Y1 та Y2 і різні значення класифікуючого атрибута D. Також можна зауважити, що таблиця є надмірною, оскільки об'єкти X1 та X4 є однаковими, і кажуть, що вони належать до одного класу еквівалентності.

У великому наборі даних, як правило, присутня велика кількість різних класів еквівалентності. Оскільки для наведення всього такого класу в таблиці достатньо залишити лише один його елемент, то так можна зменшити обсяг даних для аналізу, і це не вплине на результат. Крім цього, з таблиці видаляються суперечливі дані, які входять в неточну область.

Дискретизація даних застосовується до атрибутів числового типу для розбиття значень цих атрибутів на класи еквівалентності. Наприклад, для набору числових значень віку людини – 18, 20, 35, 21, 40, 50, 60, 65, 67 – в результаті дискретизації можна розбити множину значень віку на підмножини та кожну з них позначити номером класу: 1, 2 або 3. Класу під номером 1 відповідатиме молодий вік у проміжку від 0 до 30 років, класу «2» – вік 30 – 40 років, а класу «3» – старший вік у діапазоні 40 – 100 років. Тобто, ми отримаємо 3 вікові групи. Для того, щоб дискретизація не змінювала результатів класифікації об'єктів, необхідно враховувати значення класифікуючого атрибуту. Існує багато методів дискретизації, але у цьому випадку було використано алгоритм *MD-heuristic* [11] та алгоритми, реалізовані в системі Rosetta: *Boolean reasoning*, *Entropy/MDL*, *Equal frequency binning*, *Naive algorithm* та *Semi-naive algorithm* [12, 14, 15].

Подальшого скорочення розмірів інформаційної системи можна досягти, залишивши для аналізу лише ті атрибути, від яких залежить класифікація об'єктів таблиці. Тобто, вилучаються такі стовпці даних, наявність чи відсутність яких у таблиці не змінює результат класифікації об'єктів. Множину атрибутів, що залишилися, називають *редуктом* (reduct). Іншими словами, редукт є підмножиною множини атрибутів A і дозволяє отримати такий самий результат класифікації, як і з використанням всієї множини A. При цьому атрибуту відповідає деяка ознака важливості з інтервалу (0–1). Знаходження редукту є NP-складною задачею, однак існують досить ефективні методи, які дозволяють вирішувати цю задачу за прийнятний час. Одним з найпростіших методів, що застосовується, зокрема і у випадку неточних множин, є *Boolean reasoning* (логічне виведення).

Основою методу є побудова функції розрізнення, яка є булевою функцією від m змінних і визначається так:

$$f_A(a_1^*, ..., a_m^*) = \wedge\{\vee c_{ij}^* \mid 1 \le j \le i \le n, c_{ij}^* \ne 0\},$$

де $i, j = \overline{(1, n)}$, m – кількість атрибутів табл. $\boldsymbol{A}$, n – кількість об'єктів, a_m^* – атрибут таблиці, c_{ij}^* – елемент спеціальної *матриці розпізнавання* **M(*A*)**. Цей елемент являє собою множину атрибутів, за якими відрізняються два об'єкти u_i та u_j з U, причому ці об'єкти повинні мати різне значення атрибута прийняття рішення. Якщо об'єкти мають однакові значення атрибута прийняття рішення, то $\boldsymbol{c}_{ij}^* = 0$. Тобто, матриця **M(*A*)** є симетричною матрицею розмірів $n \times n$ з нульовою діагоналлю.

Наступним кроком є приведення функції до вигляду кон'юнктивної нормальної форми. Для цього достаньо реалізувати скорочення за законами ідемпотентності для кон'юнкції $\boldsymbol{a} \wedge \boldsymbol{a} = \boldsymbol{a}$ та спрощення $(\boldsymbol{a} \vee \boldsymbol{b}) \wedge \boldsymbol{a} = \boldsymbol{a}$. Після такого спрощення булева функція містить елементи, які відповідають атрибутам, що входять до редукту.

Аналіз таблиці медичних даних програмою Analize_R

Таблиця даних розміром 3532×15 містить атрибути Age (вік), Gender (стать: 0 – жін., 1 – чол.), інші атрибути, що наводять результати різних тестувань, та класифікуючий атрибут KHKS (0 – захворювання немає, 1 – захворювання є).

Таблиця 2

Фрагмент таблиці з результатами діагностування

№	Age	Gender	PIK	KV	SK	UA	AA	BE	OH	REW	R_AK	R_MK	R_AKMK	GH	KHKS
1	53	1	0	0	0	0	0	0	0	0	0	0	0	1	**1**
2	65	1	1	0	0	0	0	0	0	1	0	0	1	0	**0**
3	63	1	1	0	0	0	0	0	0	0	0	0	0	1	**1**
4	62	1	1	0	0	0	0	0	0	0	0	0	0	1	**1**
5	70	1	0	0	0	0	0	0	0	0	0	0	0	0	**1**
6	51	1	1	0	0	0	0	0	0	0	0	0	0	1	**0**
…															
3530	73	1	0	0	0	0	0	0	0	0	0	0	0	1	**1**
3531	69	1	0	0	0	0	0	0	0	1	0	0	1	1	**0**
3532	44	1	1	0	0	0	0	0	0	0	0	0	0	0	**0**
3533	62	1	1	0	0	0	0	0	0	0	0	0	0	1	**0**

Для аналізу наявної таблиці написано програму, яка виконує таку послідовність кроків:

- зчитування вхідних даних;
- дискретизація значень у заданому стовпці;
- вилучення однакових рядків з таблиці;
- побудова функції розрізнення і знаходження редукту;
- вилучення зайвих атрибутів, суперечливих та однакових об'єктів;
- генерування простих логічних правил.

Редукт знаходять за алгоритмом *Boolean reasoning*. Для дискретизації використано алгоритм *MD-heuristic*. Цей алгоритм є простим в реалізації, однак кількість знайдених інтервалів може бути надто великою, що не завжди є задовільним результатом. У найгіршому випадку кожному значенню, що дискретизується, відповідає окремий інтервал. Алгоритм використовує «жадібну» стратегію Джонсона, яка застосовується до матриці ***A****, побудованої на основі вихідної табл. ***A***. Об'єктами матриці ***A**** є всі пари (u_i, u_j) об'єктів з вихідної таблиці, для яких є різним значення класифікуючого атрибута. Атрибутами є так звані «*зрізи*» (cuts) – середини проміжків між двома впорядкованими за зростанням значеннями x_k та x_{k+1} з домена атрибута ***a***, що дискретизується:

$$C_k = (x_k + x_{k+1})/2.$$

Графічно поняття зрізу можна проілюструвати на числовій осі (рис. 1).

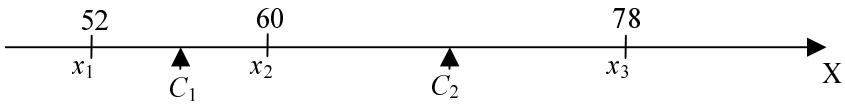

Рис. 1. Ілюстрація поняття зрізу

Показані на рисунку зрізи мають значення C_1=56 і C_2=69 і утворюють три інтервали: (-∞, 56], (56, 69], (69, +∞).

Матриця ***A**** заповнюється за таким принципом: елементу e_{lk}, що знаходиться на перетині *l*-го рядка і *k*-го стовпця, присвоюється значення 1, якщо для пари об'єктів u_i та u_j з вихідної таблиці ***A***:

$$\min(a(u_i), a(u_j)) < C_k < \max(a(u_i), a(u_j)),$$

де $a(u_i)$, $a(u_j)$ – значення атрибута ***a*** для об'єктів u_i та u_j відповідно; $k = \overline{1, m-1}$, m – розмір домена атрибута ***a***. У протилежному випадку $e_{lk} = 0$.

Неведемо алгоритм *MD-heuristic*:

Крок 1: побудувати матрицю ***A**** на основі табл. ***A***;

Крок 2: вибрати стовпчик з ***A**** з максимальною кількістю одиничок;

Крок 3: видалити з ***A**** стовпчик, вибраний на кроці 2 і всі рядки, позначені в цьому стовпчику одиницями;

Крок 4: якщо матриця ***A**** непорожня, то перехід на крок 2, інакше – зупинка.

Стовпчикам, які вибираються на другому кроці, відповідають зрізи, що утворюють шукані інтервали. Кожному інтервалу присвоюється певна позначка – його порядковий номер. Для завершення дискретизації всі значення, що дискретизувались, замінюються номерами відповідних інтервалів, у які вони потрапляють.

Для досліджуваної таблиці було дискретизовано атрибут Age. У результаті дискретизації за алгоритмом *MD-heuristic* було отримано 51 інтервал:

(0; 32.5)	(48.5; 49.5)	(61.5; 62.5)	(74.5; 75.5)
(32.5; 33.5)	(49.5; 50.5)	(62.5; 63.5)	(75.5; 76.5)
(33.5; 38.5)	(50.5; 51.5)	(63.5; 64.5)	(76.5; 77.5)
(38.5; 39.5)	(51.5; 52.5)	(64.5; 65.5)	(77.5; 78.5)
(39.5; 40.5)	(52.5; 53.5)	(65.5; 66.5)	(78.5; 79.5)
(40.5; 41.5)	(53.5; 54.5)	(66.5; 67.5)	(79.5; 80.5)
(41.5; 42.5)	(54.5; 55.5)	(67.5; 68.5)	(80.5; 81.5)
(42.5; 43.5)	(55.5; 56.5)	(68.5; 69.5)	(81.5; 82.5)
(43.5; 44.5)	(56.5; 57.5)	(69.5; 70.5)	(82.5; 83.5)
(44.5; 45.5)	(57.5; 58.5)	(70.5; 71.5)	(83.5; 84.5)
(45.5; 46.5)	(58.5; 59.5)	(71.5; 72.5)	(84.5; 85.5)
(46.5; 47.5)	(59.5; 60.5)	(72.5; 73.5)	(85.5; 100)
(47.5; 48.5)	(60.5; 61.5)	(73.5; 74.5)	

Кожне значення стовпчика Age було замінено номером інтервалу, у який воно потрапляє. Після цього всі об'єкти таблиці попарно порівнювалися й вилучалися ті з них, що повторювалися. В результаті кількість рядків таблиці зменшилась з 3532 до 854.

Наступним кроком роз'язання задачі є побудова редукту. До початку цієї процедури таблиця містила 14 атрибутів: Age, Gender, PIK, KV, SK, UA, AA, BE, OH, REW, R_AK, R_MK, R_AK_MK, GH та класифікуючий атрибут KHKS.

За методом *Boolean reasoning* побудована матриця розрізнення **M(*A*)** розмірів $n \times n$, де n = 854 – кількість об'єктів, на основі якої побудовано булеву функцію у кон'юктивній формі. Спрощенням цієї функції отримано редукт: {age, PIK, KV, SK, AA, BE, OH, REW, R_MK, GH}.

Це означає, шо лише цих 10-ти атрибутів достатньо для класифікації всіх об'єктів початкової таблиці. Решта атрибутів: Gender, UA, R_AK, R_AK_MK, є зайвими. Ці атрибути вилучаються з таблиці. Тепер вона має розмір 854×11.

Згідно з концепцією неточних множин набір даних може містити об'єкти, які належать до неточної області. Для їх пошуку здійснено перебір і порівняння об'єктів таблиці та відповідних їм значень атрибута прийняття рішення. Якщо для однакових об'єктів класифікуючий атрибут набував різних значень, то ці об'єкти вилучалися як суперечливі. Для досліджуваної таблиці знайдено 172 такі об'єкти. Крім цього, ще раз вилучено об'єкти, що повторювались. У результаті отримано таблицю розміром 521×11.

Отримання логічних правил є кінцевою метою дослідження. Ці правила відображають закономірності у вхідному наборі даних та дозволяють класифікувати нові об'єкти. Для таблиці даних такі правила мають вигляд:

IF $(A_1 = a_1)$ and $(A_2 = a_2)$ and … THEN decision = d,

де A_1, A_2, ... A_n – імена атрибутів; a_1, a_2, ... a_n – значення атрибутів; d – значення класифікуючого атрибута.

У результаті цієї процедури було згенеровано 521 правило. Деякі з них наведено на рис. 2.

Правило 1: **If** (age = 18) and (PIK = 0) and (KV = 0) and (SK = 0) and (AA = 0) and (BE = 0) and (OH = 0) and (REW = 0) and (R_MK = 0) and (GH = 1) **Then** KHKS = 1

Правило 2: **If** (age = 30) and (PIK = 1) and (KV = 0) and (SK = 0) and (AA = 0) and (BE = 0) and (OH = 0) and (REW = 1) and (R_MK = 0) and (GH = 0) **Then** KHKS = 0

Правило 3: **If** (age = 28) and (PIK = 1) and (KV = 0) and (SK = 0) and (AA = 0) and (BE = 0) and (OH = 0) and (REW = 0) and (R_MK = 0) and (GH = 1) **Then** KHKS = 1

. . .

Правило 520: **If** (age = 23) and (PIK = 1) and (KV = 0) and (SK = 1) and (AA = 0) and (BE = 0) and (OH = 0) and (REW = 0) and (R_MK = 0) and (GH = 1) **Then** KHKS = 0

Правило 521: **If** (age = 50) and (PIK = 0) and (KV = 0) and (SK = 0) and (AA = 0) and (BE = 0) and (OH = 0) and (REW = 0) and (R_MK = 0) and (GH = 0) **Then** KHKS = 0

Рис. 2. Приклад згенерованих правил

Аналіз таблиці медичних даних за допомогою системи ROSETTA

ROSETTA [15, 16] – це відкрита система, призначена для аналізу та виявлення прихованих закономірностей у даних. Дані подаються у вигляді системи прийняття рішень.

Наведемо результати використання *ROSETTA* для дослідження заданої таблиці даних (фрагмент якої подано у табл. 2). Мета цього дослідження полягала не лише в отриманні правил та оцінці якості класифікаці за допомогою нових об'єктів, а й у порівнянні ефективності методів дискретизації, яка здійснювалась на етапі обробки даних.

У заданій таблиці засобами *ROSETTA* здійснено такі дії:

1) для подальшої оцінки якості класифікації задану таблицю за чотирма різними пропорціями розбито на дві частини – навчальну та тестову. Навчальна частина таблиці становила відповідно 95, 75, 50 та 25 відсотків всіх рядків таблиці;

2) дискретизовано значення атрибута Age за п'ятьма алгоритмами: *Boolean reasoning*, *Entropy/MDL*, *Equal frequency binning*, *Naive algorithm* та *Semi-naive algorithm;*

3) знайдено редукти та виведено правила для всіх способів утворення навчальних таблиць та алгоритмів дискретизації;

4) оцінено якість класифікації за знайденими правилами.

Загалом проведено 24 експерименти, які можна розбити на чотири групи.

1. У першій групі експериментів навчальні приклади становили 95 % всієї таблиці. *Прикладом* будемо називати об'єкт таблиці, або точніше – кортеж значень атрибутів, що відповідає об'єкту. Результати цих експериментів наведені у табл. 3.

При дискретизації атрибута Age за алгоритмом *Boolean reasoning* отримано множину з п'яти зрізів, а саме: {32.5, 33.5, 38.5, 84.5, 86} та побудувано такі інтервали: (0; 32.5), (32.5; 33.5), (33.5; 38.5), (38.5; 84.5), (84.5; 86), (86; 100). Крайні значення – 0 і 100 – вибиралися довільно.

За алгоритмом *Equal frequency binding* отримано зрізи 50.5; 63.5 і, відповідно, інтервали: (0; 50.5), (50.5; 63.5), (63.5; 100).

Застосування алгоритму *Entropy/MDL* теж дало малу множину зрізів: 84.5; 86. Наслідком є такі три інтервали: (0; 84.5), (84.5; 86), (86; 100).

Результати роботи *Naive algorithm* збігаються з результатами алгоритму *MD-heuristic,* наведеними раніше, за винятком двох останніх інтервалів. Виявлено 50 зрізів, які дозволили побудувати 51 інтервал:

(0; 32.5)	(48.5; 49.5)	(61.5; 62.5)	(74.5; 75.5)
(32.5; 33.5)	(49.5; 50.5)	(62.5; 63.5)	(75.5; 76.5)
(33.5; 38.5)	(50.5; 51.5)	(63.5; 64.5)	(76.5; 77.5)
(38.5; 39.5)	(51.5; 52.5)	(64.5; 65.5)	(77.5; 78.5)
(39.5; 40.5)	(52.5; 53.5)	(65.5; 66.5)	(78.5; 79.5)
(40.5; 41.5)	(53.5; 54.5)	(66.5; 67.5)	(79.5; 80.5)
(41.5; 42.5)	(54.5; 55.5)	(67.5; 68.5)	(80.5; 81.5)
(42.5; 43.5)	(55.5; 56.5)	(68.5; 69.5)	(81.5; 82.5)
(43.5; 44.5)	(56.5; 57.5)	(69.5; 70.5)	(82.5; 83.5)
(44.5; 45.5)	(57.5; 58.5)	(70.5; 71.5)	(83.5; 84.5)
(45.5; 46.5)	(58.5; 59.5)	(71.5; 72.5)	(84.5; 86)
(46.5; 47.5)	(59.5; 60.5)	(72.5; 73.5)	(86; 100)
(47.5; 48.5)	(60.5; 61.5)	(73.5; 74.5)	

Алгоритм *Semi-Naive algorithm* знайшов зрізи: 75.5; 76.5; 78.5; 79.5; 80.5; 81.5; 86 та відповідні інтервали: (0; 75.5), (75.5; 76.5), (76.5; 78.5), (78.5; 79.5), (79.5; 80.5), (80.5; 81.5), (81.5; 86), (86; 100).

Таблиця 3

Результати досліджень у випадку використання для виведення правил 95 % прикладів із заданої таблиці

Метод дискретизації атрибута Age	К-сть інтервалів	Розмір редукту	К-сть знайдених правил	Якість класифікації
Boolean reasoning	6	10	64	0.791
Equal frequency binding	3	10	92	0.864
Entropy/MDL	3	10	42	0.79
Naive algorithm	51	10	555	0.8
Semi-Naive algorithm	8	10	83	0.796
Без дискретизації	72	10	601	0.797

Редукт шукався за алгоритмом Джонсона та за генетичим алгоритмом. Оскільки алгоритми показали однаковий результат – {AGE, PIK, KV, SK, UA, AA, BE, OH, REW, GH}, то надалі для кожного випадку розбиття таблиці прикладів розглядатиметься лише один редукт.

Далі наведемо зведені результати решти експериментів, які виконувались за тією ж схемою, що й перша група.

2. У другій групі експериментів навчальні приклади становили 75 % заданої таблиці. Результати досліджень наведені у табл. 4.

Таблиця 4

Результати досліджень у випадку використання для виведення правил 75 % прикладів з заданої таблиці

Метод дискретизації атрибута Age	К-сть інтервалів	Розмір редукту	К-сть знайдених правил	Якість класифікації
Boolean reasoning	6	9	48	0.835
Equal frequency binding	3	9	86	0.887
Entropy/MDL	3	10	29	0.838
Naive algorithm	51	10	496	0.8
Semi-Naive algorithm	8	9	50	0.829
Без дискретизації	70	10	537	0.794

У цьому випадку редукт з 10-ти атрибутів не змінився, а редукт з 9-ти атрибутів виглядає так: {AGE, PIK, KV, SK, AA, BE, OH, REW, GH}

3. У третій групі експериментів навчальні приклади становили 50 % всієї таблиці. Результати наведені у табл. 5.

Таблиця 5

Результати досліджень у випадку використання для виведення правил 50 % прикладів із заданої таблиці

Метод дискретизації атрибута Age	К-сть інтервалів	Розмір редукту	К-сть знайдених правил	Якість класифікації
Boolean reasoning	10	10	77	0.844
Equal frequency binding	3	10	78	0.88
Entropy/MDL	3	10	36	0.85
Naive algorithm	51	10	424	0.773
Semi-Naive algorithm	8	10	65	0.855
Без дискретизації	71	10	459	0.765

4. В останній групі експериментів навчальні приклади становили 25 % всієї таблиці. Результати для цієї групи наведено у табл. 6.

Таблиця 6

Результати досліджень у випадку використання для виведення правил 50 % прикладів із заданої таблиці

Метод дискретизації атрибута Age	К-сть інтервалів	Розмір редукту	К-сть знайдених правил	Якість класифікації
Boolean reasoning	10	10	56	0.826
Equal frequency binding	3	10	56	0.867
Entropy/MDL	3	10	29	0.844
Naive algorithm	45	10	301	0.714
Semi-Naive algorithm	9	10	61	0.82
Без дискретизації	67	10	335	0.699

Таблиця 7

Зведені результати оцінювання якості класифікації при різній кількості навчальних прикладів та різних алгоритмів дискретизації

Алгоритм дискретизації	Кількість навчальних прикладів із заданої таблиці, %			
	25	***50***	***75***	***95***
Boolean reasoning	0,826	0,844	0,835	0,791
Equal frequency binding	0,867	0,88	0,887	0,864
Entropy/MDL	0,844	0,85	0,838	0,79
Naive algorithm	0,714	0,773	0,8	0,8
Semi-Naive algorithm	0,82	0,855	0,829	0,796
Без дискретизації	0,699	0,765	0,794	0,797

У табл. 7 подано зведені результати дослідження якості побудованих класифікаторів. На основі цієї таблиці побудовано графік, зображений на рис. 3.

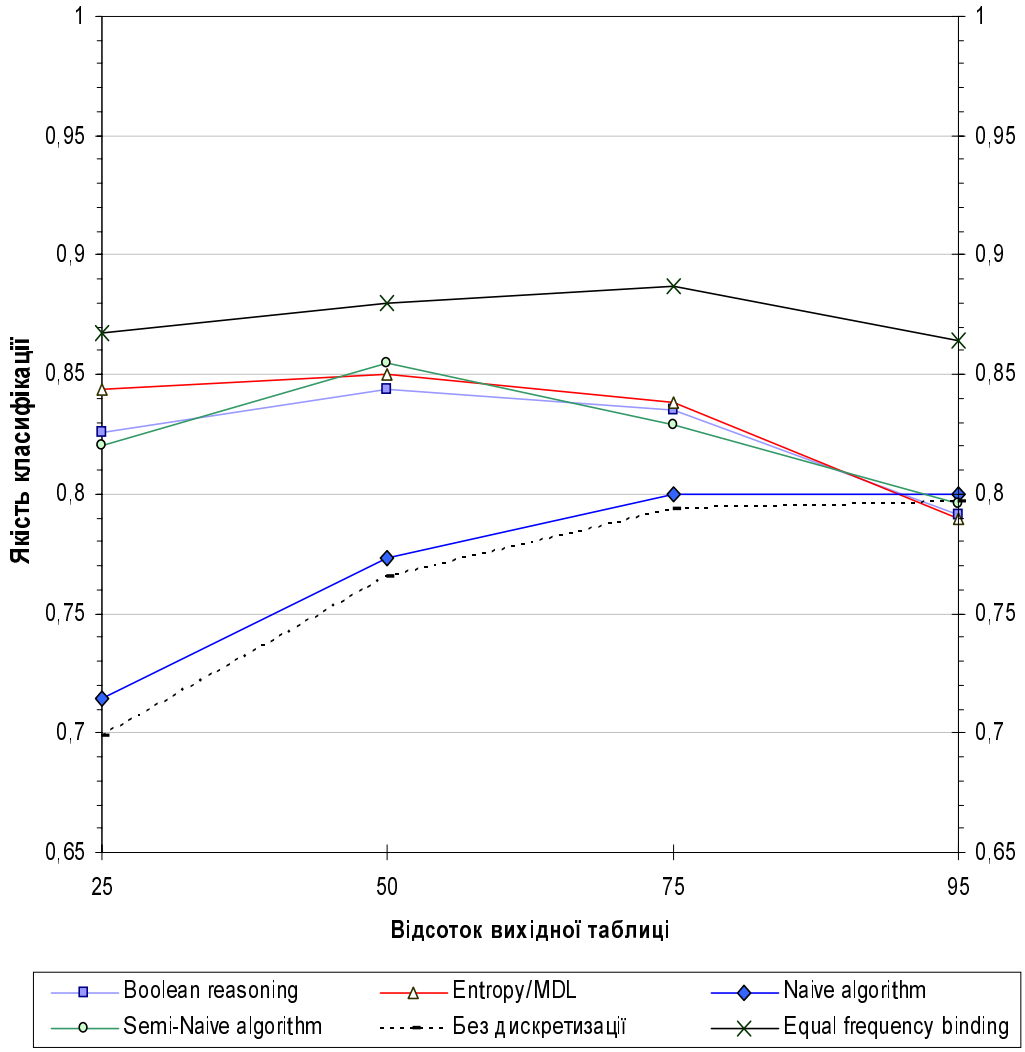

Рис. 3. Графік наведення результатів класифікації

На рис. 3 подано кінцеві результати проведеного дослідження таблиці медичних даних. Можна бачити, як змінюється якість класифікації залежно від застосованих алгоритмів дискретизації та різних варіантів розбиття досліджуваної таблиці на навчальну та тестову множини.

У першу чергу слід звернути увагу на два випадки – дискретизація за алгоритмом *Naive* та випадок без дискретизації. Якість класифікації для цих випадків є найнижчою, але ці класифікатори поводяться найбільш «природно» – зі збільшенням розмірів навчальної таблиці і, відповідно, кількості знайдених правил, покращується якість класифікації. Алгоритм *Naive* дає низький ступінь дискретизації і, як наслідок, ми досягаємо незначного зменшення розміру вихідної таблиці та отримуємо велику кількість правил (див. табл. 3–6).

Використання решти алгоритмів дискретизації дає набагато меншу кількість правил. У випадку, коли для побудови класифікатора взяли 75 % прикладів з таблиці, використання алгоритму *Equal frequency binding* дозволило досягнути найкращої якості класифікації – **0.887**. Для цього класифікатора спостерігається зростання якості класифікації зі збільшенням розміру навчальної таблиці від 25 % до 75 %.

Алгоритми *Boolean reasoning*, *Entropy/MDL* та *Semi-Naive* дозволили досягти якості класифікації у межах 0.79 – 0.855, причому максимум спостерігається, коли навчальна таблиця становить 50 % від початкової. Зі збільшенням навчальної множини до 95 % якість класифікації погіршується до 0.79 – 0.796. Очевидно, модель, що складається з великої кількості правил, зокрема, як у випадках використання алгоритму *Naive* та без дискретизації, є більш стійкою до особливостей наявних даних. Натомість класифікатори з малою кількістю правил показують кращу якість, причому найкращі результати отримуються при розбитті таблиці на навчальну і тестову у пропорції 50 % на 50 %, яка є стандартною при дослідженні алгоритмів машинного навчання.

Висновки

У статті наведено результати дослідження таблиці медичних даних великих розмірів, яка містила результати діагностування 3532 пацієнтів.

Аналіз даних показав, що для постановки діагнозу достатньо 10 параметрів, які відповідають атрибутам таблиці. Шляхом дискретизації віку виявлено певні вікові групи пацієнтів, знання про які може бути корисним при постановці діагнозів. Результатом дослідження стала побудова класифікатора на основі виявлених у таблиці правил, який дозволяє правильно класифікувати 88,7 % нових об'єктів. Однак не виключено, що використовуючи всі можливості системи ROSETTA можна добитися дещо кращих результатів.

Одним із шляхів покращання результатів є контроль над якістю вхідних даних, а в процесі аналізу – використання так званих динамічних чи наближених редуктів. Динамічні редукти будуються для окремих частин основної таблиці, внаслідок чого виявлені за їх допомогою правила є більш стійкими до «шумів» та аномалій в даних. У випадку наближених редуктів кожному атрибуту, який входить у такий редукт, присвоюється коефіцієнт важливості, що теж підвищує якість правил.

Крім цього, в системі ROSETTA є можливість зміни параметра «RNG seed», який відповідає за випадковість вибірки прикладів при розбитті таблиці на навчальну і тестову множини. Попередні експерименти показали, що у випадку, коли навчальна множина становить 95 % досліджуваної таблиці, збільшення значення цього параметра призводить до покращання якості класифікації на 4–6 %.

У перспективі можна говорити про дедалі більшу інтеграцію методології неточних множин з іншими методами Data mining, наприклад, з методологією розмитих множин. Це дозволить якісніше моделювати складні предметні області та знаходити нові галузі для практичного застосування таких гібридних підходів.

1. Usama Fayyad, Gregory Piatetsky-Shapiro, Padhraic Smyth, From Data Mining to Knowledge Discovery in Databases. AAAI'97, Providence, Rhode Island, July 27–31, 1997 (http://www.aaai.org/Conferences/National/1997/aaai97.html). 2. Арсеньев С. Извлечение знаний из медицинских баз данных, 2000 (http://neural.narod.ru/Arsen.htm). 3. Mitchell T. Machine learning. The McGraw-Hill Companies, Inc. 1997. 4. Дюк В., Самойленко А. Data mining: Учебный курс. – СПб: Питер, 2001. 5. Necht-Nielsen R. Neurocomputing. Addison-Vesley, 1990. 6. Уоссерман Ф. Нейрокомпьютерная техника. – М.: Мир, 1992. 7. Goldberg D. E. Genetic Algorythms in Search, Optimization and Mashine

Learning. Addison-Vesley, 1989. 8. Pawlak Z., Rough Sets – Theoretical Aspects of Reasoning about Data. Kluwer Academic Publishers, Dordrecht, 1991. 9. Pawlak Z., Information systems – theoretical foundations. Information Systems 6, 1981. – C. 205–218. 10. Pawlak Z., Rough sets. International Journal of Computer and Information Sciences 11, 1982. – P. 341–356. 11. Komorowski J., Pawlak P., Polkowski L., Skowron A. Rough Sets: A Tutorial. //Eds. S.K.Pal and A. Skowron, Rough Fuzzy Hybridization: A New Trend in Decision-Making, Spriner-Verlag, Singapore, 1998. – P.3–98. 12. Hung Son Nguyen and Sinh Hoa Nguyen. Some efficient algorithms for rough set methods. In Proc. Fifth Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems (IPMU'96), pages 1451–1456, Granada, Spain, July, 1996. 13. Frank Markham Brown. Boolean Reasoning: The Logic of Boolean Equations. Kluwer Academic Publishers, Dordrecht, The Netherlands, 1990. 14. H. S. Nguyen and S. H. Nguyen. Some efficient algorithms for rough set methods. In Proc. Fifth Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems (IPMU'96), July 1996. 15. Aleksander Øhrn, ROSETTA Technical Reference Manual, 2001 (http://www.idi.ntnu.no/~aleks/). 16. Aleksander Øhrn, Discernibility and Rough Sets in Medicine: Tools and Applications, PhD thesis, Norwegian University of Science and Technology, Department of Computer and Information Science, 1999.

УДК 621.372

В.М. Заяць
Національний університет "Львівська політехніка",
кафедра програмного забезпечення автоматизованих систем

АНАЛІЗ ДИНАМІКИ ТА УМОВ СТІЙКОСТІ ДИСКРЕТНИХ МОДЕЛЕЙ КОЛИВНИХ СИСТЕМ

Запропоновано підхід до побудови універсальної моделі дискретної коливної системи, яка має широкий спектр динамічних режимів. Отримані аналітичні оцінки амплітуди та частоти гармонічних та квазігармонічних коливань, які апробовані для широкого класу функцій, що використовуються при побудові моделі. Встановлені необхідні та достатні умови стійкості виявлених режимів.

Approach to construction of universal model of the discrete oscillation system which own the wide spectrum of the dynamic modes is offered. Analytical estimations of amplitude and frequency of vibrations harmonic and quasi-harmonic, and which are approved for the wide class of functions, that are used for construction of model, are got. Set necessary and sufficient terms of stability of the exposed modes.

Постановка проблеми в загальному вигляді

При розробленні реальних коливних приладів чи дослідженні фізичних явищ, що володіють бажаними характеристиками інформаційного сигналу за амплітудою, частотою і формою, доцільно провести їх аналіз та комп'ютерне моделювання шляхом створення математичної моделі об'єкта, що розробляється. Такий підхід вимагає значно менших часових і технічних засобів порівняно з фізичним експериментом, особливо на попередній стадії розробки, коли пристрій, що розробляється, відсутній.

Останнім часом в нелінійній динаміці широке застосування знаходять дискретні моделі коливних систем [5–9], для яких дискретність закладена в природі самого об'єкта досліджень, а не є наслідком дискретизації неперервної системи. Доцільність використання дискретних за своєю природою моделей пояснюється такими їх особливостями:

– простотою математичного опису порівняно з неперервними моделями;

– наявністю широкого спектра динамічних режимів;