

УДК 519.71:612.17

В. Марценюк, докт. техн. наук; З. Майхрук

Тернопільський державний медичний університет ім. І.Я. Горбачевського

ЯКІСНИЙ АНАЛІЗ СИСТЕМИ РІВНЯНЬ ХОДЖКІНА–ХАКСЛІ ЕЛЕКТРИЧНОЇ АКТИВНОСТІ АКСОНА НА ОСНОВІ КЛАСИФІКАЦІЙНИХ ПРАВИЛ

***Резюме.** Наведено новий підхід до якісного аналізу системи рівнянь Ходжкіна–Хакслі мультіваріативним методом. Раніше такий аналіз був розроблений шляхом застосування багатовимірного підходу при дослідженні залежності форми траєкторії розв'язку рівнянь від початкових умов. Тепер забезпечено можливість урахування констант швидкостей через застосування алгоритму послідовного покриття. Дано оцінку складності. Підхід реалізовано у вигляді програмного забезпечення в пакеті Java-класів.*

***Ключові слова:** якісний аналіз, рівняння Ходжкіна–Хакслі, класифікаційні правила.*

V. Martsenyuk, Z. Mayhruk

QUALITATIVE ANALYSIS OF HODGKIN-HUXLEY AXON ELECTRICAL ACTIVITY BASED ON CLASSIFICATION RULES

***Summary.** The paper presents an approach of qualitative analysis of the Hodgkin-Huxley model based on multivariate method including sequential covering algorithm. Earlier this method was developed as a multivariate approach to research dependence trajectory form on initial conditions. Here we extend this method for rates constants too.*

The method developed consists of 5 steps: at the first step we determine different excitability types, namely type I, type II, type III introduced in the Hodgkin's works; at the second step we randomly generate initial values and rate constants, as in the Monte-Carlo methods; the third step is running the model and classification of trajectories obtained; at the fourth step we construct matrix of relations between random initial values and rate constants in categorized form – as a result we have the set of learning tuples; the fifth step is application of sequential covering algorithm to learning tuples.

Thus we get as a result the set of classification rules for prediction of neuron excitability type. We also analyze computational complexity of sequential algorithm applied. Estimation of time necessary for induction of classification rules is obtained. Experiments investigating dependences of classification rules and number of attributes or amount of tuples are presented. The approach is implemented as software consisting of package of Java-classes.

***Key words:** qualitative analysis Hodgkin-Huxley model, classification rules.*

Аналіз стану загальної проблеми. При вивченні збудливості аксона виникає необхідність у класифікації типів цієї збудливості. Для цього використовують значення початкових умов з математичної моделі Ходжкіна–Хакслі електричної активності гігантського аксона кальмара [1–7]. У роботі [8] запропоновано мультіваріативну класифікацію на основі дерева рішень із використанням початкових умов системи звичайних диференціальних рівнянь (ЗДР) при фіксованих швидкісних константах. Значення швидкісних констант для моделі аксона при дослідженні його збудливості отримують з літературних джерел, їх вимірюваннями або оцінкою за часовими рядами, отриманими з траєкторій розв'язків рівнянь моделі тощо. Це викликає значне підвищення складності досліджень.

У даній роботі наведено автоматизований мультіваріативний метод визначення комбінації параметрів моделі Ходжкіна–Хакслі для класифікації типу збудливості.

Метою роботи є удосконалення мультіваріативного методу якісного аналізу моделі Ходжкіна-Хакслі на основі класифікаційних правил шляхом автоматизованого врахування як швидкісних констант, так і початкових умов цієї моделі.

Метод. Для дослідження багатопараметричних впливів використовується мультіваріативний метод якісного аналізу моделі Ходжкіна-Хакслі. Засади методу викладено в роботі [8] для випадку початкових умов ЗДР. Для поширення цього методу й на швидкісні константи використано підхід Монте-Карло – генерування випадкових значень цих параметрів та побудову на їх основі ЗДР. У подальшому за алгоритмом послідовного покриття будуються класифікаційні правила.

Модель електричної активності гігантського аксона кальмара запропоновано в роботі [9]. Кожен компонент збудливої клітини представлено електричною величиною: ліпідний шар – ємністю C_m ; іонні канали – провідністю g_i , яка залежить від напруги та часу, i – специфічний іонний канал; іонні насоси – джерелом струму I_{app} ; різницю між мембранним та залишковим потенціалами – V . Вираз струму через біліпідний шар набув вигляду

$$I_c = C_m \frac{dV}{dt}.$$

Вираз струму через заданий іонний канал

$$I_i = g_i(V - V_i),$$

де V_i – рівноважний потенціал i -го іонного каналу. Для клітини з калієвими, натрієвими та хлорними каналами загальний струм через мембрану I становитиме

$$I = I_c + I_K + I_{Na} + I_L.$$

Остаточно типова модель Ходжкіна-Хакслі набув вигляду

$$\frac{dV}{dt} = -g_K n^4 (V - V_K) - g_{Na} m^3 h (V - V_{Na}) - g_L (V - V_L) + I_{app}, \quad (1)$$

$$\frac{dm}{dt} = 0.1(1 - m) \frac{25 - V}{\exp \frac{10}{10} - 1} - 4m \exp \frac{V}{18}, \quad (2)$$

$$\frac{dn}{dt} = 0.1(1 - n) \frac{10 - V}{\exp \frac{10}{10} - 1} - 0.125n \exp \frac{V}{80}, \quad (3)$$

$$\frac{dh}{dt} = 0.07 \exp \frac{-V}{20} (1 - h) - \frac{h}{1 + \exp \frac{30 - V}{10}}. \quad (4)$$

Встановлено, що співвідношення між параметрами у виразах (1–4) такі:

$$p \in P = \{(g_K, g_{Na}, g_L, V_K, V_{Na}, V_L, C_m, x_m, x_n, x_h) : g_K^{\min} \leq g_K \leq g_K^{\max},$$

$$g_{Na}^{\min} \leq g_{Na} \leq g_{Na}^{\max}, g_L^{\min} \leq g_L \leq g_L^{\max}, V_K^{\min} \leq V_K \leq V_K^{\max}, V_{Na}^{\min} \leq V_{Na} \leq V_{Na}^{\max},$$

$$V_L^{\min} \leq V_L \leq V_L^{\max}, C_m^{\min} \leq C_m \leq C_m^{\max}, x_m^{\min} \leq x_m \leq x_m^{\max}, x_n^{\min} \leq x_n \leq x_n^{\max},$$

$$x_h^{\min} \leq x_h \leq x_h^{\max}\} \subset R^{10},$$

а поміж початковими умовами –

$$(V_0, m_0, n_0, h_0) \in X_0 = \{(V_0, m_0, n_0, h_0) : V_0^{\min} \leq V_0 \leq V_0^{\max}, m_0^{\min} \leq m_0 \leq m_0^{\max},$$

$$n_0^{\min} \leq n_0 \leq n_0^{\max}, h_0^{\min} \leq h_0 \leq h_0^{\max}\} \subset R^4.$$

Суть удосконалення методу полягає у генеруванні випадкових значень початкових умов та швидкісних параметрів, які б належали раніше практично обґрунтованим областям. Для кожного з наборів таких параметрів інтегруються рівняння (1–4). До отриманих результатів (траєкторій) застосовується алгоритм послідовного покриття для визначення типів траєкторій. Загалом метод включає п'ять кроків.

1. *Означення класів траєкторій.* У практичних застосуваннях переважно мають справу зі складними формами збудливості. Охарактеризувати їх поняттями «стійка-нестійка» і, відповідно, вдатися до аналізу власних значень або ж експонент Ляпунова як динамічної системи складно. Тому визначення форм траєкторій поведінки якісно встановлено експертами-фізіологами. Відомі класи з типами збудливості нейронів I, II, III [10]. Для позначення класу траєкторії введемо атрибут $C \in \overline{1,3}$. На рис.1–3 наведено типові представлення поведінки цих класів траєкторій при збільшенні сили прикладеного струму — відповідно типу I, типу II і типу III.

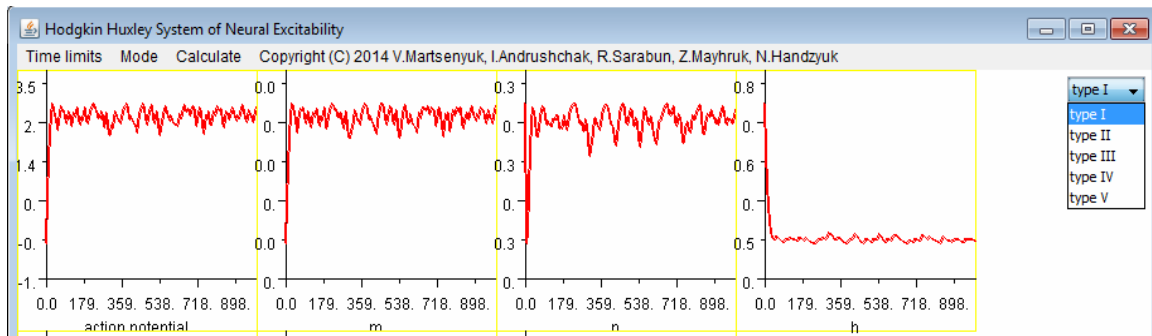


Рисунок 1. Графіки поведінки нейрона типу I

Figure 1. Type I of action of neuron

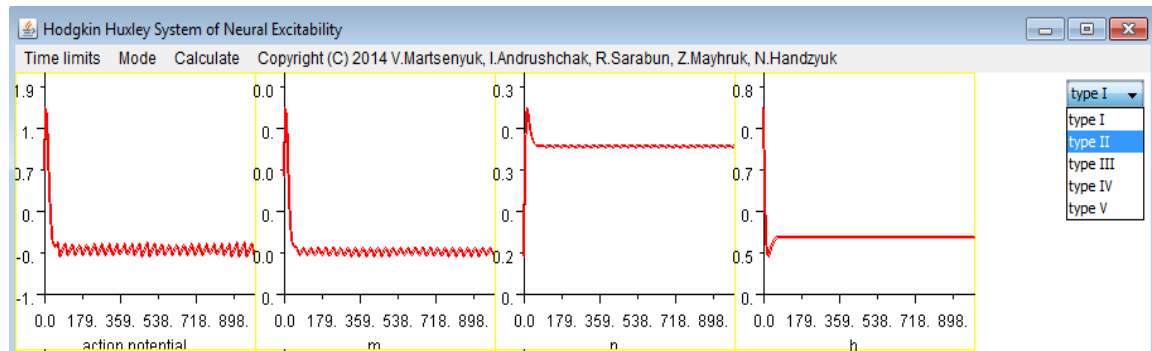


Рисунок 2. Графіки поведінки нейрона типу II

Figure 2. Type II of action of neuron

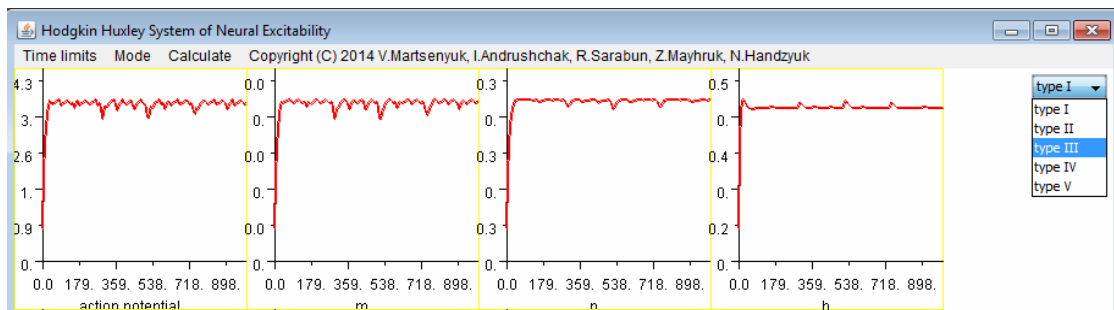


Рисунок 3. Графіки поведінки нейрона типу III

Figure 3. Type III of action of neuron

2. Генерування матриці випадкових початкових умов та швидкісних констант. Для дослідження класів траєкторій, визначених на першому кроці, початкові умови та швидкісні параметри генеруються на основі ймовірнісних розподілів їх значень у визначених інтервалах. Ми припускаємо, що ці розподіли рівномірні. Кожен стовпчик матриці відповідає множині значень одного параметра – або початкове значення, або швидкісна константа. Кожен рядок є набором початкових значень та швидкісних констант для одного «запуску» моделі:

$$M = \begin{pmatrix} V_0^1 & m_0^1 & n_0^1 & h_0^1 & g_K^1 & g_{Na}^1 & g_L^1 & V_K^1 & V_{Na}^1 & V_L^1 & C_m^1 & x_m^1 & x_n^1 & x_h^1 \\ V_0^2 & m_0^2 & n_0^2 & h_0^2 & g_K^2 & g_{Na}^2 & g_L^2 & V_K^2 & V_{Na}^2 & V_L^2 & C_m^2 & x_m^2 & x_n^2 & x_h^2 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ V_0^N & m_0^N & n_0^N & h_0^N & g_K^N & g_{Na}^N & g_L^N & V_K^N & V_{Na}^N & V_L^N & C_m^N & x_m^N & x_n^N & x_h^N \end{pmatrix} \in R^{N \times 14}$$

3. «Запуск» моделі й класифікація набору вхідних даних. Кожен набір початкових значень та швидкісних констант, згенерованих на другому кроці, є вхідними даними для моделі Ходжкіна-Хакслі. Чисельне інтегрування її рівнянь здійснюється за допомогою методу Адамса [11]. Вихідні траєкторії класифікуються на основі критеріїв, заданих на першому кроці. За результатом класифікації наборам початкових значень і швидкісних констант приписуються відповідні значення атрибуту $C_l \in \overline{1,3}$.

4. Побудова матриці залежностей. Припущено, що для форм траєкторій співвідношення між початковими умовами та між швидкісними константами є набагато важливішими, ніж їх числові значення. Тому будується матриця кодів категорій співвідношень між значеннями початкових умов та швидкісних констант, генерованих на другому кроці:

$$D = \begin{pmatrix} m_0 \otimes n_0 & m_0 \otimes h_0 & n_0 \otimes h_0 & g_K \otimes g_{Na} & g_K \otimes g_L & g_{Na} \otimes g_L & V_K \otimes V_{Na} \\ x(m_0^1, n_0^1) & x(m_0^1, h_0^1) & x(n_0^1, h_0^1) & p(g_K^1, g_{Na}^1) & p(g_K^1, g_L^1) & p(g_{Na}^1, g_L^1) & p(V_K^1, V_{Na}^1) \\ x(m_0^2, n_0^2) & x(m_0^1, h_0^1) & x(n_0^1, h_0^1) & p(g_K^1, g_{Na}^1) & p(g_K^1, g_L^1) & p(g_{Na}^1, g_L^1) & p(V_K^1, V_{Na}^1) \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ x(m_0^k, n_0^k) & x(m_0^1, h_0^1) & x(n_0^1, h_0^1) & p(g_K^1, g_{Na}^1) & p(g_K^1, g_L^1) & p(g_{Na}^1, g_L^1) & p(V_K^1, V_{Na}^1) \\ V_K \otimes V_L & V_{Na} \otimes V_L & C_m \otimes 1 & x_m \otimes x_n & x_m \otimes x_h & x_n \otimes x_h & C \\ p(V_K^1, V_L^1) & p(V_{Na}^1, V_L^1) & p(C_m^1, 1) & p(x_m^1, x_n^1) & p(x_m^1, x_h^1) & p(x_n^1, x_h^1) & C_1 \\ p(V_K^1, V_L^1) & p(V_{Na}^1, V_L^1) & p(C_m^1, 1) & p(x_m^1, x_n^1) & p(x_m^1, x_h^1) & p(x_n^1, x_h^1) & C_2 \\ p(V_K^1, V_L^1) & p(V_{Na}^1, V_L^1) & p(C_m^1, 1) & p(x_m^1, x_n^1) & p(x_m^1, x_h^1) & p(x_n^1, x_h^1) & C_k \end{pmatrix} \in R^{k \times 14},$$

де $C_l \in \overline{1,3}$ – значення атрибуту l - класу, пов'язані з відповідними формами траєкторій,

$$x(u, v) = p(u, v) = \begin{cases} 0, & \text{if } u < v \\ 1, & \text{if } u = v \\ 2, & \text{if } u > v \end{cases}$$

На даному кроці значення початкових умов та швидкісних констант трансформуються у категоріальні значення атрибутів наборів навчальних даних. Оскільки ймовірність рівності випадкових чисел дорівнює нулю, то матриця D виражає своєрідну «бінаризацію» співвідношень між початковими умовами та між швидкісними константами. Тобто матриця D включатиме лише значення 0 та 2.

5. Застосування алгоритму послідовного покриття до співвідношень між початковими умовами та між швидкісними константами. Матриця «бінаризованих» співвідношень D , побудована на четвертому кроці, є набором навчальних даних для алгоритму послідовного покриття. Побудований набір класифікаційних правил міститиме перевірку співвідношень в антецедентах. В якості консеквентів правил будуть класи $C \in \overline{1,3}$ траєкторій моделі.

Реалізація методу. Для реалізації методу розроблено пакет Java-класів – rule.model. До складу пакета rule.model входять такі класи (рис.4): beans-класи Attribute, Attribute_for_list – для роботи з даними відповідних таблиць; Rule – для представлення правил; AttributeListPeer, TuplesPeer – SQL-запити отримання відповідних даних;

Rule_set – набір навчальних правил і алгоритм послідовного покриття (менеджер даних m_dataManager, хеш-таблиці наборів навчальних даних m_htTuples та усіх атрибутів з їх можливими значеннями m_htAtt_vals, множину правил m_htRule_set).

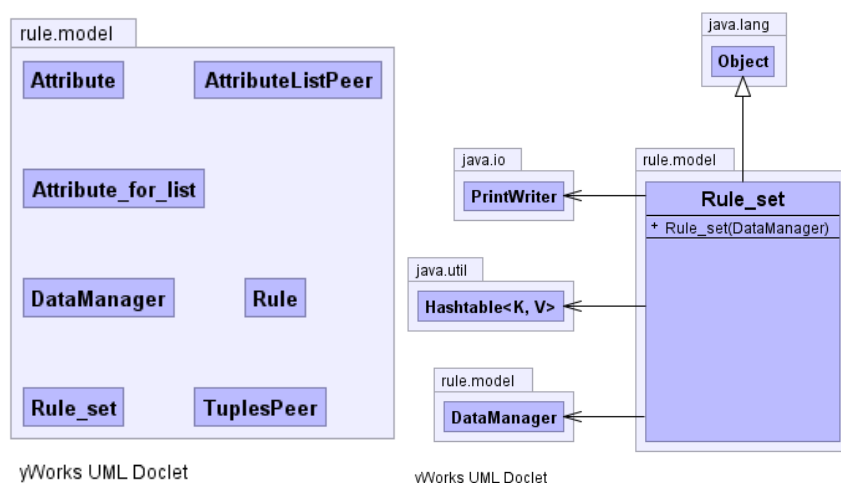


Рисунок 4. Схема пакета rule.model

Figure 4. Schema of package karule.model

У конструкторі класу Rule_set здійснюється побудова хеш-таблиць m_htTuples та m_htAtt_vals, а також застосування алгоритму послідовного покриття – через виклик методу Sequential_covering(m_htTuples, m_htAtt_vals). Отримана множина правил виводиться в текстовий файл.

Клас Rule призначений для зберігання окремих правил. Його членами класу є дві хеш-таблиці: m_htAntecedent – для зберігання антецеденту правила та m_htConsequent – для консеквента.

За допомогою методу public void conjunctCondition(Attribute_for_list attribute, String sAttribute_value) здійснюється кон'юнкція нової умови до правила.

За допомогою методу public Rule copy() створюється «глибока» копія правила. При цьому використовується протокол JOS (Java Object Serialization).

Підрахунок кількості позитивних та негативних навчальних наборів здійснюється у методах класу TuplesPeer. У класі fde.hh.MultiVariateMethod (рис.5).

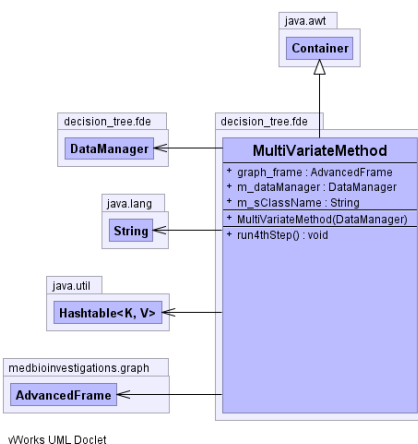


Рисунок 5. UML-діаграма класу MultiVariateMethod

Figure 5. UML-class diagram MultiVariateMethod здійснюється генерація випадкових значень параметрів (крок 2):

M_x0 = dm.getRandomInitialValues();

M_rateConstants = dm.getRandomRateConstants().

Потім запускається аплет інтегрування рівнянь Ходжкіна-Хакслі. При цьому експерт здійснює вибір форми отриманої траєкторії (крок 3). Після цього запускається крок генерування матриці взаємозв'язків параметрів (крок 4). Зауважимо, що послідовність кроків 2–4 може виконуватися як завгодно багато разів. У будь-який момент користувач може запустити алгоритм послідовного покриття (крок 5)

```
rule.model.Rule_set rule_set = new rule.model.Rule_set(rule_dataManager, sql0, rules_file_url)
```

База даних fde, що використовується в пакеті, реалізована в СУБД MySQL. Вона містить такі таблиці (рис.6):

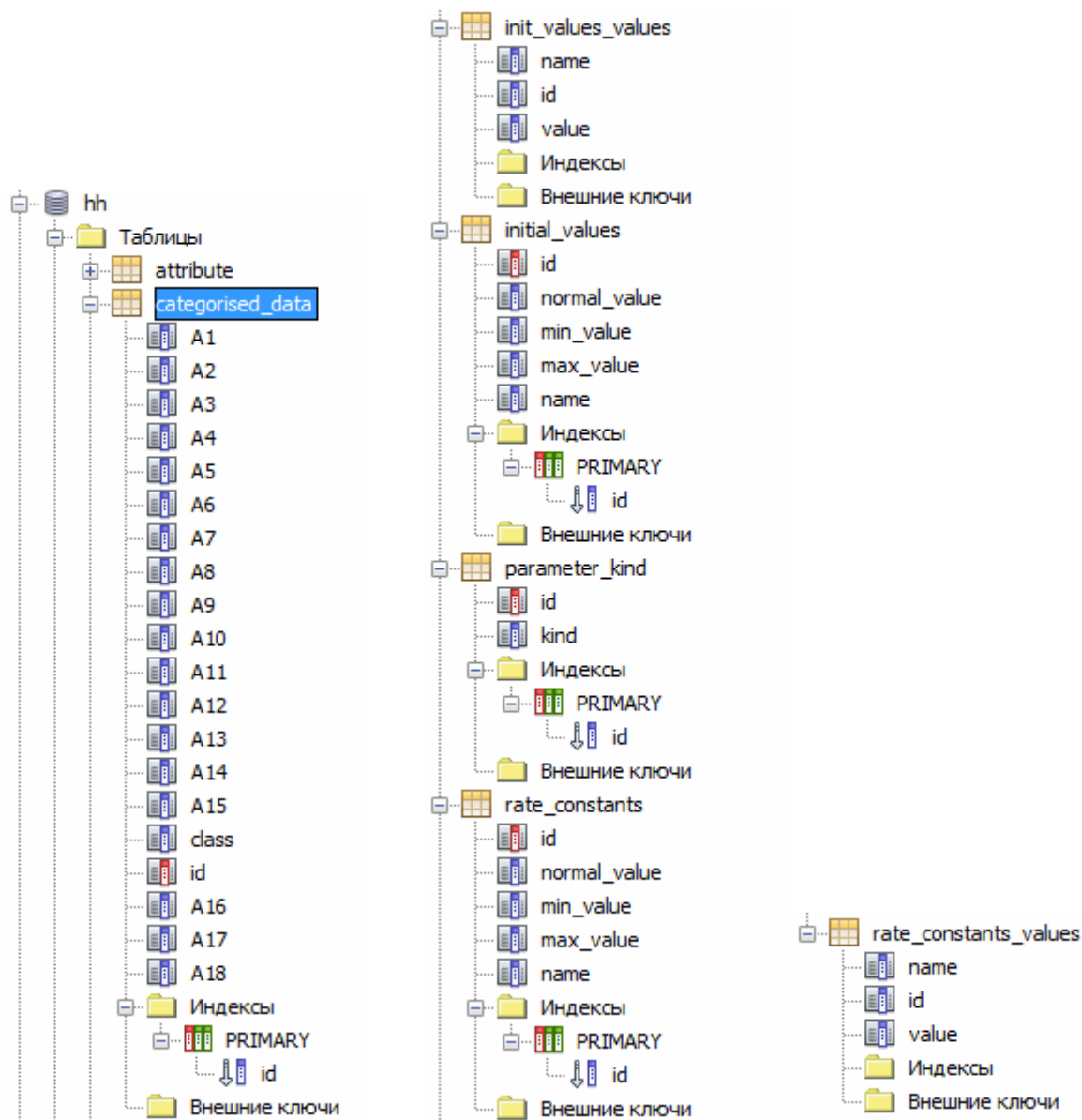


Рисунок 6. Таблиці hh бази даних

Figure 6. Database Tables hh

- attribute – опис атрибутів для побудови класифікаційних правил, тобто взаємозв'язків між початковими умовами та між швидкісними константами;
- categorized_data – навчальні набори, що використовуються в алгоритмі послідовного покриття і являють собою матрицю D на четвертому кроці;
- init_values_values – матриця генерованих випадковим чином початкових значень:

$$\begin{pmatrix} V_0^1 & m_0^1 & n_0^1 & h_0^1 \\ V_0^2 & m_0^2 & n_0^2 & h_0^2 \\ \dots & \dots & \dots & \dots \\ V_0^N & m_0^N & n_0^N & h_0^N \end{pmatrix} \in R^{N \times 4};$$

- initial_values – опис початкових значень (включаючи мінімальні та максимальні);
- parameter_kind – вид параметра;
- rate_constants – опис швидкісних констант (включаючи мінімальні та максимальні значення);
- rate_constants_values – матриця згенерованих випадковим чином швидкісних констант

$$\begin{pmatrix} g_K^1 & g_{Na}^1 & g_L^1 & V_K^1 & V_{Na}^1 & V_L^1 & C_m^1 & x_m^1 & x_n^1 & x_h^1 \\ g_K^2 & g_{Na}^2 & g_L^2 & V_K^2 & V_{Na}^2 & V_L^2 & C_m^2 & x_m^2 & x_n^2 & x_h^2 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ g_K^N & g_{Na}^N & g_L^N & V_K^N & V_{Na}^N & V_L^N & C_m^N & x_m^N & x_n^N & x_h^N \end{pmatrix} \in R^{N \times 10}.$$

Особливості програмної реалізації моделі Ходжкіна-Хакслі. Пакет rule.model може бути використаний для широкого класу систем диференціальних рівнянь. Для цього модель на основі диференціальних рівнянь повинна бути реалізована у вигляді відповідного пакета Java-класів. Прикладом такого пакета у випадку моделі Ходжкіна-Хакслі є пакет medbioinvestigations.hodgkin_huxley (рис.7).

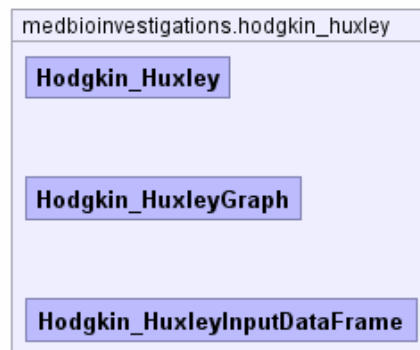


Рисунок 7. Пакет medbioinvestigations.hodgkin_huxley

Figure 7. Packet medbioinvestigations.hodgkin_huxley

Для інтеграції з пакетом decision_tree.fde.hh у класі Hodgkin_HuxleyGraph, що здійснює графічну візуалізацію моделі, поряд з існуючим було додано новий конструктор, що використовує посилання за значенням на інстанцію класу MultiVariateMethod. У цьому конструкторі додатково створюється об'єкт класу JComboBox, що дозволяє вибирати форму траєкторії та запускати виконання 4-го кроку методу:


```
String[] classStrings = {"subclinical", "chronic", "acute", "lethal"};
JComboBox m_jcbClassName = new JComboBox(classStrings);

m_jcbClassName.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        JComboBox jcbClass = (JComboBox) e.getSource();
        m_sClassName = (String)jcbClass.getSelectedItem();
        mvm.m_sClassName = m_sClassName;
        ((AdvancedFrame)getParent()).dispose();
        mvm.run4thStep();
    }
});
```

Також внесено відповідні зміни у таблиці бази даних hh у такому порядку:

- описано усі початкові значення та швидкісні константи в таблицях initial_values та rate_constants відповідно;
- описано залежності між початковими значеннями та між швидкісними константами, які досліджуватимемо, у таблиці attribute;
- у таблиці categorised_data створено поля у відповідності з даними таблиці attribute.

Оцінювання складності методу. Аналізом алгоритму послідовного покриття встановлено, що обчислювальна складність визначається добутком кількості можливих значень атрибуту класу $K = 3$ (кількість ітерацій зовнішнього циклу) та обчислювальної складності процедури *Добути_одне_правило* (D, Att_vals, c), яка виконується всередині кожного циклу.

Процедура *Добути_одне_правило* (D, Att_vals, c) включає виконання $p = 18$ ітерацій. На кожній ітерації для певного атрибуту A_i проводиться розрахунок міри *FOIL_Gain* для кожного з $K_i = 2$ значень атрибуту. Тобто внутрішнє тіло

циклу в процедурі *Добути_одне_правило* (D, Att_vals, c) виконується $\sum_{i=1}^p K_i = 36$ разів.

Міра *FOIL_Gain* обчислюється в результаті 3-х SQL-запитів, складність яких $O(\log(N))$ (див. документацію до MySQL 5.0 – <http://dev.mysql.com/doc/refman/5.0/en/select-speed.html>). Отже, в цілому процедура *Добути_одне_правило* (D, Att_vals, c) має обчислювальну складність

$O\left(\sum_{i=1}^p K_i \log(N)\right) = O(36 \log(N))$. Складність усього алгоритму послідовного покриття тоді порядку

$$O\left(K \sum_{i=1}^p K_i \log(N)\right) = O(108 \log(N)) = O(\log(N)). \quad (5)$$

На основі моделі (1–4) було проведено чисельний експеримент для порівняння часу побудови набору класифікаційних правил з його оцінюванням (5). Використовувалася система на основі процесора Celeron(R) Dual-Core CPU T3300 @ 2.00 GHz та 2 GB RAM. Результати представлені на рис.8.



Рисунок 8. Графік залежності часової складності від розміру навчального набору

Figure 8. Complexity in time units vs size of the teaching sample

Побудований набір класифікаційних правил для $n = 397$ наведено нижче.

IF $x_m < x_h$ AND $x_n > x_h$ THEN class=type I

IF $n_0 < h_0$ AND $x_n < x_h$ THEN class=type II

IF $v_{Na} > v_L$ AND $x_m > x_h$ AND $x_n < x_h$ THEN class=type III

Умови в антецеденті цих правил мають біологічну інтерпретацію.

Значне зростання відхилення від оцінки (5) часу побудови класифікаційних правил при $N > 350$ пов'язане із неврахуванням в цій оцінці зростання апаратного ресурсу.

Висновки. Результати виконаного дослідження у порівнянні з існуючими можливостями реалізації побудови класифікаційних правил із використанням алгоритму послідовного покриття (зокрема у складі програм Mathcad, StatSoft Statistica і ін.) мають суттєву перевагу, оскільки, через їх інтегрованість в Інтернет-проекти отримано можливість формування навчальних наборів при залученні експертів через Інтернет-доступ, що значно розширює та балансує їх множину.

Conclusions. There are many possibilities for implementations of algorithm of sequential covering for constructing of rules of classification, in particular as part of programs Mathcad, StatSoft Statistica, etc. A significant advantage of this implementation is that this is developed for the Internet projects. Consequently, the formation of the learning sets can be involving experts from Internet access. That is extending and is balancing the set of the learning sets.

Список використаної літератури

1. Brian Hassard. Bifurcation of periodic solutions of the Hodgkin-Huxley model for the squid giant axon [Text] // J. Theoret. Biol. – V.71 (1978), Issue 3. – P. 401–420.
2. Fukai H, Doi S, Nomura T, Sato S. Hopf bifurcations in multiple-parameter space of the Hodgkin-Huxley equations [Text] // Biol. Cybern. – V.82 (2000). – P.223–229.
3. J. Guckenheimer, I. S. Labouriau, Bifurcation of the Hodgkin–Huxley equations: A new twist [Text] // Bull. Of Math. Biol. – V.55 (1993), No.5. – P. 937–952.
4. B. Hassard, L.-J. Shiau. A special point of Z2-codimension three Hopf bifurcation in the Hodgkin-Huxley model [Text] // Appl. Math. Lett. – V.9 (1996). – P.31–34.
5. S. Labouriau. Degenerate Hopf bifurcation and nerve impulse. Part II [Text] // SIAM J. Math. Anal. – V.20 (1989), No.1. – P.1–12.
6. Майхрук З.В. Програмна реалізація чисельного методу оптимального керування біфуркацією в моделі Ходжкіна-Хакслі [Текст] / З.В. Майхрук // Вісник Хмельницького національного університету. – 2014. – №1(209). – С.186–194.
7. Майхрук, З.В. Побудова оптимального керування біфуркацією в моделі Ходжкіна-Хакслі на основі принципу максимуму [Текст] / З.В. Майхрук, В.П. Марценюк // Математичне та комп'ютерне моделювання: наук. журнал інституту кібернетики імені В.М. Глушкова НАН України Кам'янець-Подільського національного університету імені Івана Огієнка. – Вип.9. – 2014. – С.78–90.

8. Koch Y, Wolf T, Sorger PK, Eils R, Brors B (2013) Decision-Tree Based Model Analysis for Efficient Identification of Parameter Relations Leading to Different Signaling States. PLoS ONE; Dec2013, Vol. 8, Issue 12, p1: e82593. doi:10.1371/journal.pone.0082593.
9. Hodgkin, A.L., Huxley, A.F. A quantitative description of membrane current and its application to conduction and excitation in nerve [Text]// J Physiol. Aug 28, 1952; 117(4). – P.500–544. PMC 1392413. PMID 12991237.
10. J. R. Clay et al. A simple modification of the Hodgkin and Huxley equations explains type 3 excitability in squid giant axons [Text] // J. R. Soc. Interface (2008) 5(29), – P. 1421–1428
11. Ваннер, Г. Решение обыкновенных дифференциальных уравнений. Нежесткие задачи [Текст] / Г. Ваннер, С. Нерсетт, Э. Хайпер – М.: Мир, 1990. – 512 с.

Отримано 17.12.2014