

УДК 519.71:612.171.1

Марценюк В.П., д.т.н., професор
Майхрук З.В., асистент

**Мультиваріативний метод якісного
аналізу моделі Ходжкіна-Хакслі з
індукцією дерева рішень**

Тернопільський державний медичний
університет ім. І. Я.Горбачевського, 46000
м. Тернопіль, майдан Воли,1 e-mail:
marceniuk@yahoo.com
e-mail:zorana_majhruk2010@mail.ru

Marceniuk V.P, professor
Mayhruk Z.V., assistant

**Multivariate method of qualitative analysis
of Hodgkin-Huxley model with the decision
tree induction**

I.Ya.Horbachevsky Ternopil State Medical
University., 46001, Ternopil, Voli Square, 1 e-mail:
marceniuk@yahoo.com
e-mail:zorana_majhruk2010@mail.ru

У роботі запропоновано підхід якісного аналізу системи Ходжкіна-Хакслі на основі мультиваріативного методу. При цьому на виході отримуємо дерево рішень для прогнозування типу збудливості нейрона. Підхід доведено до програмної реалізації в пакеті Java-класів.

Ключові слова: якісний аналіз, система Ходжкіна-Хакслі, дерево рішень.

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 to rate constants also.

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, like in the Monte-Carlo methods; the third step is running the model and classification of trajectories obtained; at the forth 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 output the set of classification rules for prediction of neuron excitability type.

We also analyze computational complexity of sequential algorithm applied. Estimate 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.

Keywords: qualitative analysis, Hodgkin-Huxley system, decision tree

Статтю представив д.т.н., п.н.с. Кудін В.І.

Вступ. Якісна теорія диференціальних рівнянь у застосуванні до динамічних систем виявилася вкрай важливою у визначенні топології і структурної стійкості. Теорія бере початок з робіт Пуанкаре, далі отримала розвиток у роботах Біркгофа, Ляпунова, Андронова, Арнольда і ін.

У багатьох практичних задачах необхідно з'ясувати характер розв'язку диференціального рівняння, що описує певний фізичний процес та описати властивості розв'язків, коли незалежна змінна знаходиться на скінченному або нескінченному інтервалі.

На жаль існує лише кілька відносно простих рівнянь, загальні розв'язки яких можуть бути виражені в термінах інтегралів відомих функцій.

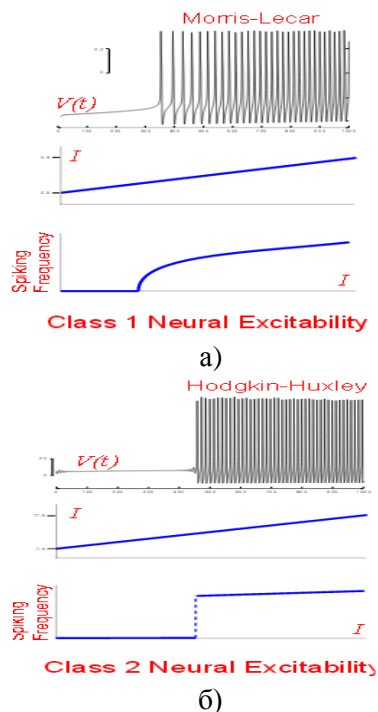
Отже, виникає проблема дослідження властивостей розв'язків диференціального рівняння виходячи безпосереднього із самого рівняння. Оскільки розв'язок диференціального рівняння задається у вигляді кривої на площині або у просторі, то задача полягає у дослідженні властивостей інтегральних кривих, їх розподілів та їх поведінки в околі сингулярних точок. Наприклад, чи вони знаходяться у обмеженій частині площини, чи вони не містять відгалужень, що прямують до нескінченності, чи деякі із них є замкнутими кривими і та. ін. [1]

Використання диференціальних рівнянь для математичного моделювання зовсім не означає отримання аналітичної формули для їх розв'язків (або апроксимацій). Тим більше, що у більшості

випадків таких формул не існує. До того ж дуже часто формула для явного або наближеного аналітичного розв'язку настільки складна, що вона нічого не зможе сказати про природу розв'язку доти, поки не буде побудований графік [2]. Тому у практичних дослідженнях переважно генерують графіки розв'язків (принаймні їх загальну форму) не вдаючись до отримання розв'язків у вигляді формул [2].

Попередньо вже вивчалось питання різних типів збудливості нейронів. При збільшенні прикладеного струму збудливі мембрани можуть переходити із стану спокою до повторних спалахів потенціалу дії кількома способами [3]:

- для збудливих нейронів типу I частота потенціалу дії збільшується плавно від початку спалаху, таким чином проходячи увесь діапазон частот;
- для збудливих нейронів типу II спостерігається різкий скачок частоти потенціалу дії для певного порогового значення сили струму. Потенціал дії не індукується для значень частоти, нижчих від порогового;
- відповіддю на стійкі надпорогові імпульси струму є один зубець. Така поведінка отримала назву «збудливість типу III» (рис.2). [4]



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

Рис. 1. Класифікація збудливості нейронів (з роботи [5]: а) тип I; б) тип II

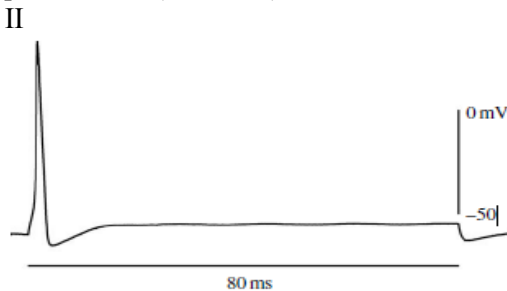


Figure 1. Response of a squid giant axon, type 3, to a sustained current pulse $15 \mu A \text{ cm}^{-2}$ in amplitude. $T = 8^\circ C$.

Рис. 2. Тип III збудливості нейронів (з роботи [4])

Модель електричної активності мембрани клітини Ходжкіна-Хакслі. Розглядається модель електричної активності гігантського аксона кальмара, запропонована в роботі [6]. У моделі кожен компонент збудливої клітини розглядається як електричний елемент. Ліпідний шар представляється як ємність C_m . Іонні канали представлені електричною провідністю g_i , де i – специфічний іонний канал, яка залежить як від напруги, так і часу. Іонні насоси представлені джерелом струму I_{app} . Позначимо через V різницю між мембранним потенціалом та залишковим потенціалом.

Струм через біліпідний шар становитиме:

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

Струм через заданий іонний канал буде:

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

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 має вигляд:

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

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

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

На сьогодні існує широкий клас пакетів програмного забезпечення з метою отримання чисельних розв'язків основних початкових задач для систем звичайних диференціальних рівнянь.

Метою даної роботи є розробити мультиваріативний метод якісного аналізу моделі Ходжкіна-Хакслі, що зводиться до дерева рішень і враховує як швидкісні константи, так і початкові умови.

Мультиваріативний метод якісного аналізу моделі Ходжкіна-Хакслі. Завданням методу є встановлення механізмів багатопараметричних впливів у моделі Ходжкіна-Хакслі.

Параметри

$$p \in P = \{(g_K, g_{Na}, g_L, V_K, V_{Na}, V_L, C_m, x_m, x_n, x_h) : g_K^{\min} \leq g_K \leq g_K^{\max}, \\ g_{Na}^{\min} \leq g_{Na} \leq g_{Na}^{\max}, g_L^{\min} \leq g_L \leq g_L^{\max}, V_K^{\min} \leq V_K \leq V_K^{\max}, V_{Na}^{\min} \leq V_{Na} \leq V_{Na}^{\max}, \\ V_L^{\min} \leq V_L \leq V_L^{\max}, C_m^{\min} \leq C_m \leq C_m^{\max}, x_m^{\min} \leq x_m \leq x_m^{\max}, x_n^{\min} \leq x_n \leq x_n^{\max}, \\ x_h^{\min} \leq x_h \leq x_h^{\max}\} \subset R^{10},$$

а початкові умови

$$(V_0, m_0, n_0, h_0) \in X_0 = \{(V_0, m_0, n_0, h_0) : V_0^{\min} \leq V_0 \leq V_0^{\max}, m_0^{\min} \leq m_0 \leq m_0^{\max}, \\ n_0^{\min} \leq n_0 \leq n_0^{\max}, h_0^{\min} \leq h_0 \leq h_0^{\max}\} \subset R^4$$

Далі випадковим чином генеруватимемо початкові значення та значення швидкісних параметрів, які б належали практично обгрунтованій області. Для кожного з наборів таких параметрів здійснюється інтегрування системи (1)-(4) з отриманням відповідних траєкторій. До отриманих результатів далі застосовується алгоритм індукції дерева рішень з метою знаходження певних шаблонів для прийняття рішень.

Отже, в цілому підхід включає такі п'ять кроків.

Загальні ідеї методу були розроблені в роботі [7] для випадку ЗДР. При цьому використовуємо підхід Монте-Карло, який полягає у випадковій генерації параметрів та побудові на їх основі моделі Ходжкіна-Хакслі. Далі застосовують алгоритм індукції дерева рішень. Зауважимо, що метод роботи [7] застосовувався лише для дослідження впливу початкових умов на траєкторію системи ДР. В даній роботі метод буде розвинено також до швидкісних параметрів.

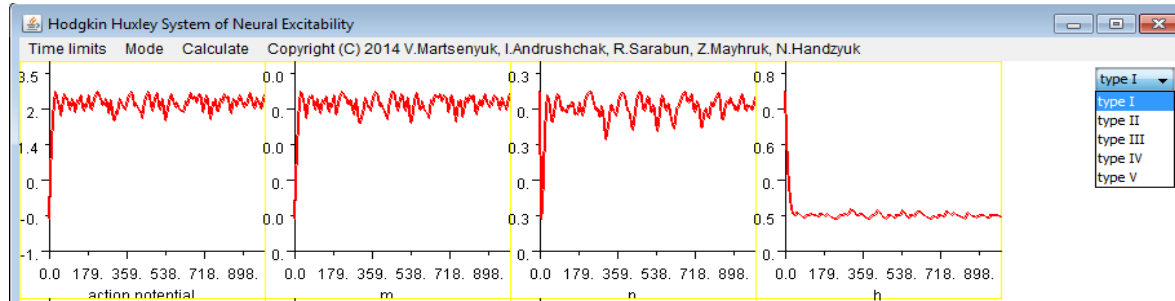
Отже, припускається існування моделі на основі (1)-(4) при початкових значеннях та швидкісних параметрах із заданих інтервалів.

Параметри

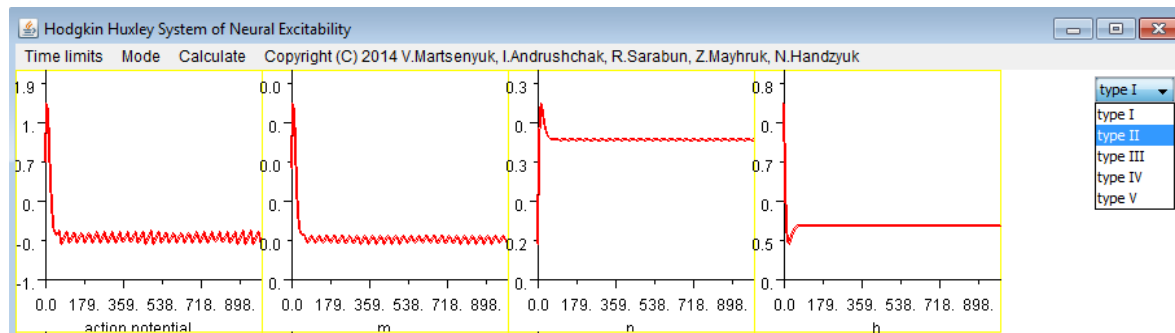
1. Означення класів траєкторій системи. Зазначимо, що в практичних застосуваннях переважно мають справу з набагато складнішими формами поведінки, щоб охарактеризувати їх поняттями «стійка-нестійка» і відповідно вдатися до аналізу власних значень або ж експонент Ляпунова динамічної системи. Тому визначення якісних форм процесу доцільно передати до компетенції експертів. В даному випадку використовуватимемо класи, пов'язані з формами збудливості нейронів: тип I, тип II, тип III. Для позначення класу траєкторії вводиться атрибут

класу C , який приймає одне з 3-х дискретних значень $C \in \overline{1,3}$. На рисунках наведені типові представлення для 3-х класів траєкторій – типів збудливості нейронів:

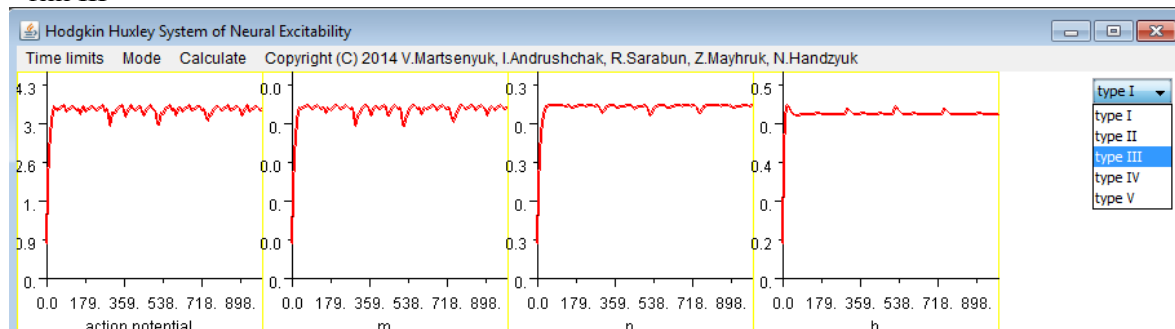
- тип I



- тип II



- тип III



Зазначимо, що при цьому визначальним є зміна частоти потенціалу дію у відповідь на зростання сили прикладеного струму.

2. Генерація матриці випадкових початкових значень та швидкісних параметрів. Для того, щоб дослідити весь простір початкових значень та швидкісних параметрів щодо генерації класів траєкторій, визначених на першому кроці, генерується матриця випадкових початкових значень та швидкісних параметрів на основі

ймовірнісних розподілів у визначених інтервалах. У даній роботі ми припускаємо, що початкові значення та швидкісні параметри розподілені рівномірно на інтервалах. Кожен стовпчик відповідає множині значень одного параметру – або початкове значення, або швидкісний параметр. Кожен рядок є набором початкових значень та швидкісних параметрів для одного запуску моделі на основі (1)-(4):

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

3. Запуск моделі і класифікація набору вхідних даних. Кожен набір початкових значень та швидкісних параметрів, згенерованих на другому кроці, використовуються в якості входу для моделі Ходжкіна-Хакслі. Чисельне інтегрування рівнянь здійснюється за допомогою методу Адамса [8]. Вихідні траєкторії класифікуються на основі критеріїв, запропонованих на першому кроці. Виходячи з результатів класифікації наборам початкових значень і швидкісних параметрів приписуються відповідні атрибути класів.

4. Побудова матриці залежностей між початковими значеннями та між швидкісними параметрами. Метод припускає, що для форми траєкторій системи співвідношення між початковими значеннями та між швидкісними значеннями є набагато важливішими порівняно з їх абсолютними значеннями. Тому будується матриця, що включає інформацію у категорованому кодованому вигляді про співвідношення між початковими значеннями та між швидкісними параметрами, з генерованими на кроці 2

$$D = \begin{pmatrix} m_0 \otimes n_0 & m_0 \otimes h_0 & n_0 \otimes h_0 & g_K \otimes g_{Na} & g_K \otimes g_L & g_{Na} \otimes g_L & V_K \otimes V_{Na} \\ x(m_0^1, n_0^1) & x(m_0^1, h_0^1) & x(n_0^1, h_0^1) & p(g_K^1, g_{Na}^1) & p(g_K^1, g_L^1) & p(g_{Na}^1, g_L^1) & p(V_K^1, V_{Na}^1) \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ x(m_0^k, n_0^k) & x(m_0^k, h_0^k) & x(n_0^k, h_0^k) & p(g_K^k, g_{Na}^k) & p(g_K^k, g_L^k) & p(g_{Na}^k, g_L^k) & p(V_K^k, V_{Na}^k) \end{pmatrix}$$

$$\left. \begin{pmatrix} V_K \otimes V_L & V_{Na} \otimes V_L & C_m \otimes 1 & x_m \otimes x_n & x_m \otimes x_h & x_n \otimes x_h & C \\ p(V_K^1, V_L^1) & p(V_{Na}^1, V_L^1) & p(C_m^1, 1) & p(x_m^1, x_n^1) & p(x_m^1, x_h^1) & p(x_n^1, x_h^1) & C_1 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ p(V_K^k, V_L^k) & p(V_{Na}^k, V_L^k) & p(C_m^k, 1) & p(x_m^k, x_n^k) & p(x_m^k, x_h^k) & p(x_n^k, x_h^k) & C_2 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ p(V_K^1, V_L^1) & p(V_{Na}^1, V_L^1) & p(C_m^1, 1) & p(x_m^1, x_n^1) & p(x_m^1, x_h^1) & p(x_n^1, x_h^1) & C_k \end{pmatrix} \right\} \in R^{k \times 14}$$

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

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

Отже, на даному кроці чисельні значення початкових значень та швидкісних параметрів трансформуються у категоріальні значення атрибутів наборів навчальних даних. Оскільки ймовірність рівності випадкових чисел дорівнює нулю, то матриця D виглядає свого роду «бінаризацією» співвідношень між початковими значеннями та між швидкісними параметрами. Тобто матриця D включатиме лише значення 0 та 2.

5. Застосування алгоритму індукції дерева рішень до співвідношень між початковими значеннями та між швидкісними параметрами. Матриця бінарних співвідношень D , побудована на кроці 4, є набором навчальних даних для алгоритму індукції дерева рішень. Побудоване дерево рішень міститиме перевірку співвідношень між початковими значеннями та швидкісними параметрами у своїх вузлах. В якості листків дерева будуть класи траєкторій моделі $C \in \overline{1,4}$.

Програмна реалізація мультиваріативного методу. Для реалізації методу розроблено пакет Java-класів `decision_tree.fde.hh`. До складу пакету входять класи (рис.1):

`DataManager` – клас – менеджер даних для отримання інформації з бази даних через посередництво відповідних класів-слуг;

MultiVariateMethod – клас для реалізації мультіваріативного методу, представленого в роботі – головний клас пакету;

TuplesPeer – клас-слуга для формування і обробки навчальних наборів, що використовуватиметься у класифікаційному алгоритмі.

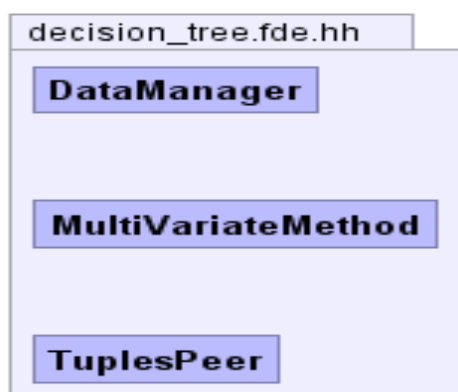


Рис. 3. Пакет decision_tree.fde.hh

У класі MultiVariateMethod (рис.4) здійснюється генерація випадкових значень параметрів (крок 2):

```
M_x0 = dm.getRandomInitialValues();
M_rateConstants = dm.getRandomRateConstants();
```

Далі запускається аплет інтегрування системи Ходжкіна-Хакслі. При цьому експерт здійснює вибір форми отриманої траєкторії (крок 3). Після цього запускається крок генерації 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 індукції дерева рішень (крок 5):

```
dtDecision_tree_HH = new
decision_tree.fde.hh.DecisionTree(dmtnRoot,
dataManager_FDE, htAttribute_list, "Information
gain");
```

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

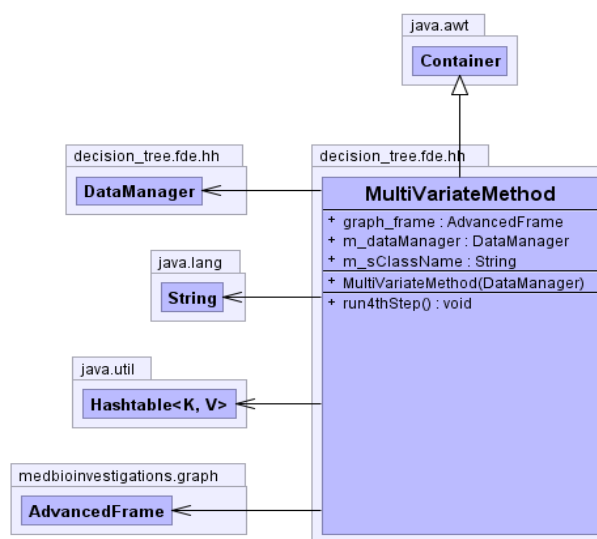


Рис.2. UML-діаграма класу MultiVariateMethod

База даних hh, що використовується в пакеті, реалізована в СУБД MySQL. Вона включає такі таблиці (рис.3):

attribute – опис атрибутів для побудови дерева рішень, тобто взаємозв'язків між початковими значеннями та між швидкісними константами; categorized_data – навчальні набори, що використовуються в класифікаційному алгоритмі (в даному випадку індукції дерева рішень) і представляють собою матрицю D на четвертому кроці; init_values_values – матриця згенерованих 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 значення);

parameter_kind – вид параметру;

rate_constants – опис швидкісних констант (включаючи мінімальні та максимальні значення);

rate_constants_values – матриця згенерованих випадковим чином швидкісних констант:

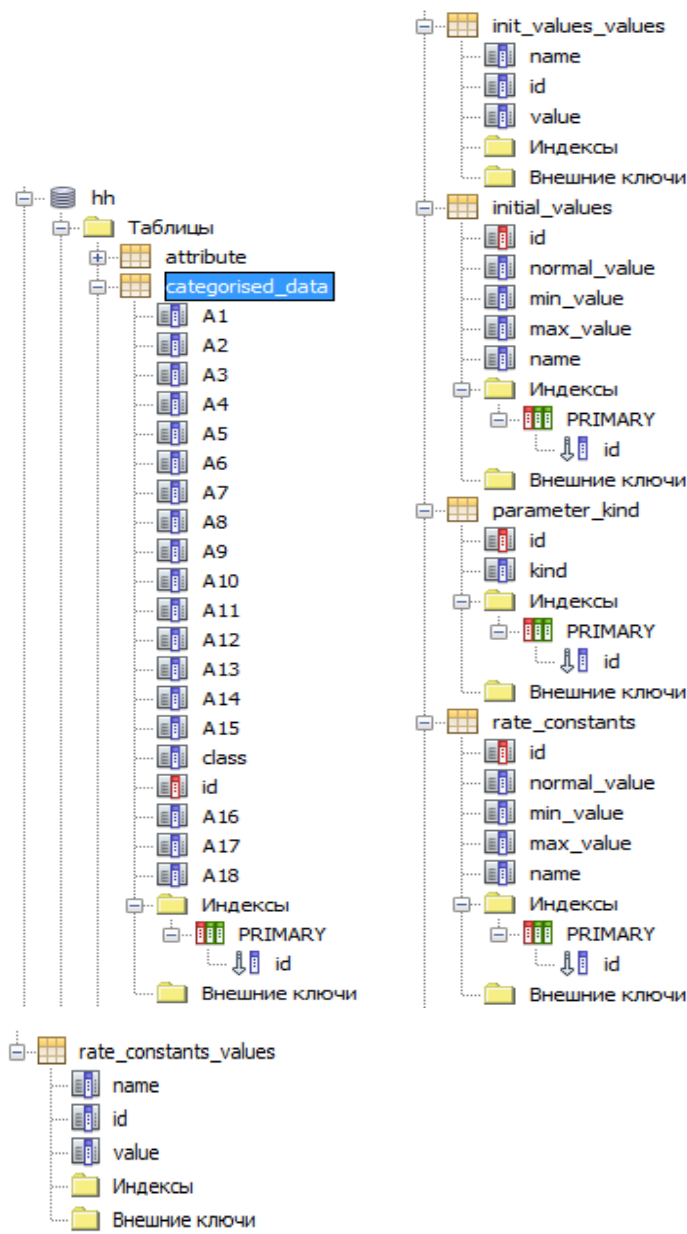


Рис.3. Таблиці бази даних hh

Особливості програмної реалізації моделі Ходжкіна-Хакслі. Пакет `decision_tree.fde.hh` може бути використаний для широкого класу систем функціонально-диференціальних рівнянь. Для цього модель на основі функціонально-

диференціальних рівнянь повинна бути реалізована у вигляді відповідного пакету Java-класів. Прикладом такого пакету у випадку моделі Ходжкіна-Хакслі є пакет `medbioinvestigations.hodgkin_huxley` (рис.4).

Для інтеграції з пакетом `decision_tree.fde.hh` в 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 тип

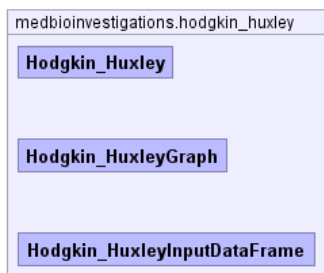


Рис. 4. Пакет `medbioinvestigations.hodgkin_huxley`

збудливості нейрона та запускати виконання 4-го кроку методу:

```
String[] classStrings = {"type I","type II","type III","type IV","type V"};
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.getSelected
Item();
mvm.m_sClassName = m_sClassName;
((AdvancedFrame)getParent()).dispose();
mvm.run4thStep();
}
});
```

Також слід внести відповідні зміни у таблиці бази даних hh в такому порядку:

- описати усі початкові значення та швидкісні константи в таблицях initial_values та rate_constants відповідно;
 - описати залежності між початковими значеннями та між швидкісними константами, які досліджуватимемо, у таблиці attribute;
 - у таблиці categorised_data створити поля у відповідності з даними таблиці attribute.
- Оцінка складності виконання алгоритму.

Як показано в роботі [9] обчислювальна складність для індукції дерева рішень для множини з n навчальних наборів, що використовують p атрибутів, $T(n)$ виражається таким рекурентним співвідношенням:

$$T(n) = \begin{cases} b \equiv const, & \text{if } n = 1 \\ 2T(n/2) + pn, & \text{if } n > 1 \end{cases} \quad (5)$$

Зауважимо, що дане співвідношення відображає оптимальне значення обчислювальної складності, яке має місце у випадку добре збалансованої множини навчальних наборів. У такому разі з (5) маємо:

$$T(n) = pn \sum_{i=0}^{\log_2 n} 1^i = pn \log_2 n. \quad (6)$$

На основі моделі (1)-(4) було проведено чисельний експеримент з метою з'ясування узгодженості часу побудови дерева рішень з оцінкою (6). Використовувалась система на основі процесора Celeron(R) Dual-Core CPU

T3300 @ 2.00 GHz та 2 Гб RAM. Результати представлені на рис. 5.

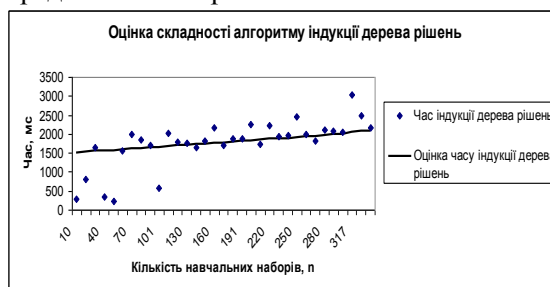


Рис.5. Оцінка складності алгоритму індукції дерева рішень

З рисунка 5 бачимо певний розкид в часі індукції дерева рішень залежно від величини n , що пов'язано в першу чергу із незбалансованістю множини навчальних наборів. Дійсно, застосування методу Монте-Карло гарантує рівномірне представлення усіх початкових значень та швидкісних констант. У той же час це не гарантує рівномірного представлення значень атрибуту класу, від чого залежить виконання порядку редуційного алгоритму індукції дерева рішень.

Побудоване дерево рішень для $n=17$ наведено на рисунку 6. Час побудови дерева рішень 1402 мс.

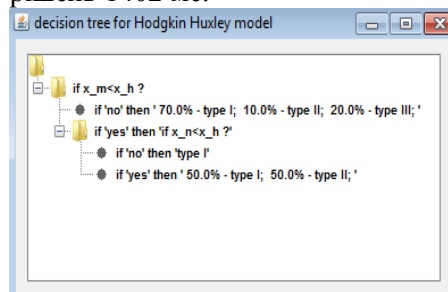


Рис.6. Дерево рішень

Висновки. В роботі запропоновано та програмно реалізовано мультіваріативний метод якісного аналізу моделі Ходжкіна-Хакслі.

1. На п'ятому кроці алгоритму до набору навчальних даних можуть бути застосовані інші алгоритми data mining – наприклад, з метою отримання класифікаційних правил – метод послідовного покриття.
2. Набори класифікаційних правил також можуть бути отримані виходячи з дерева рішень.
3. Існує чимало реалізацій алгоритму індукції дерева рішень, зокрема у складі програм Mathcad, StatSoft Statistica і ін. Суттєвою перевагою даної реалізації є те, що алгоритм розвинено до Інтернет-проекту, де до формування навчальних

наборів залучаються експерти з Інтернет-множини навчальних наборів. доступом, що значно розширює та балансує

Список використаних джерел

1. *Alexandrov A. D. Mathematics: Its Content and Meaning / A. D. Alexandrov, A. N. Kolmogorov // Courier Dover Publications, 1999. – 1120pp.*
2. *Lea A. Segel, Mathematical Models in Molecular and Cellular Biology / Segel Lea A. // CUP Archive, 1980. – 757pp.*
3. *Hodgkin A. L. The local electric changes associated with repetitive action in a non-modulated axon / A. L. Hodgkin // J. Physiol., 107: 1948. – 165–181pp.*
4. *J. R. Clay et al. A simple modification of the Hodgkin and Huxley equations explains type 3 excitability in squid giant axons / J. R. Clay et al // J. R. Soc. Interface 5, 2008. – 1421 – 1428pp.*
5. *Izhikevich E. M. Neural Excitability, Spiking and Bursting / Izhikevich E.M. // International Journal of Bifurcation and Chaos, Volume 10, Issue 6: 2000. – 1171 – 1266pp.*
6. *Hodgkin A. L. A quantitative description of membrane current and its application to conduction and excitation in nerve / A. L. Hodgkin, A. F. Huxley // The Journal of physiology 117 (4): 1952. – 500 – 544pp.*
7. *Koch Y. Decision-Tree Based Model Analysis for Efficient Identification of Parameter Relations Leading to Different Signaling States / Y. Koch, T. Wolf, P. K. Sorger, R. Eils, B. Brars // PLOS ONE | www.plosone.org, December 2013, Volume 8, Issue 12, e82593.*
8. *Хайпер Э. Решение обыкновенных дифференциальных уравнений. Нежесткие задачи. / Э. Хайпер, С. Норсетт, Г. Ваннер. Под ред. С. С. Филиппова. – 1 изд. – М.: Мир, 1990. – 512 с.*
9. *Latkowski R. High Computational Complexity of the Decision Tree Induction with many Missing Attribute Values / R. Latkowski // In: L. Czaja (Ed.) “Proceedings of Concurrency, Specification and Programming. CS&P’2003, Czarna, September 25-27, Volume 2.” Zaklad Graficzny UW, 318 – 325pp.*

References

1. *ALEXANDROV, A. D., KOLMOGOROV A.N., (1999) Mathematics: Its Content and Meaning. Courier Dover Publications.*
2. *LEA A. SEGEL, (1980) Mathematical Models in Molecular and Cellular Biology. CUP Archive.*
3. *HODGKIN, A. L. (1948) The local electric changes associated with repetitive action in a non-modulated axon. J. Physiol.*
4. *CLAY, J. R. et al. (2008) A simple modification of the Hodgkin and Huxley equations explains type 3 excitability in squid giant axons. J. R. Soc. Interface 5, 1421–1428pp.*
5. *IZHIKEVICN, E. M. (2000) Neural Excitability, Spiking and Bursting. International Journal of Bifurcation and Chaos, Volume 10, Issue 6.*
6. *HODGKIN, A. L., HUXLEY, A. F. (1952). A quantitative description of membrane current and its application to conduction and excitation in nerve. The Journal of physiology 117 (4): 500–544. PMC 1392413. PMID 12991237.*
7. *KOCN, Y., WOLF, T, SORGER, P. K., EILS, R., BRARS, B. (2013) Decision-Tree Based Model Analysis for Efficient Identification of Parameter Relations Leading to Different Signaling States / PLOS ONE | www.plosone.org, December 2, Volume 8, Issue 12, e82593.*
8. *HAIRER, E., NORSETT, S., WANNER H. (2008) Solving Ordinary Differential Equations I. — 2 edition. — Berlin: Springer.*
9. *LATKOWSKI, R. (2003) “High Computational Complexity of the Decision Tree Induction with many Missing Attribute Values” In: L. Czaja (Ed.) “Proceedings of Concurrency, Specification and Programming. CS&P’2003, Czarna, September 25 – 27, Volume 2.” Zaklad Graficzny UW, 318 – 325pp.*

Надійшла до редколегії 07.10.14