

Qualitative Analysis of the Hodgkin-Huxley Model of Neuron Excitability based on Classification Rules

Vasyl Martsenyuk¹[0000-0001-5622-1038], Yuriy Rudyak²[0000-0003-1836-9132],

Andriy Sverstiuk²[0000-0001-8644-0776], Zorana Mayhruk²[0000-0001-8909-0644],

Andriy Horkunenko²[0000-0002-2021-006X], Mykhailo Kasianchuk³[0000-0002-4469-8055]

¹ University of Bielsko-Biala, Department of Informatics and Automatics, Poland

² I. Gorbachevsky Ternopil National Medical University, Ternopil, Ukraine

³ Ternopil National Economic University, Ternopil, Ukraine

vmartsenyuk@ath.bielsko.pl, sverstyuk@tdmu.edu.ua,
majhruk@tdmu.edu.ua, horkunenkoab@tdmu.edu.ua

Abstract. The proposed multivariate method of qualitative analysis of models of electrophysiological processes is an approach that allows to solve problems of classification of excitability of cells which cannot be solved by other traditional methods, for example stability theory, or boundary cycles. As a whole, the method combines the Monte Carlo approach for the formation of training kits and the data mining classification algorithms: the sequential coverage method with the generation of classification rules and the induction method of the decision tree. The advantages of the classification rules, which can be built on the 5th step of the algorithm, are that they correspond to the natural reflection of knowledge in the thinking of people and are more expressive. In addition, the sequential coverage algorithm is easier to implement and debug than the recursive algorithms of decision trees, and its computational complexity is simpler than that of finite state machines. The decision tree induction method, being more complex to implement, allows visualization and a priori probability values for the type of cell excitability based on the relationship between the initial values and the velocity parameters in the Hodgkin-Huxley model. The developed method consists of 5 stages. The approach is proven in the form of software in the Java-class package.

Keywords: Qualitative analysis Hodgkin-Huxley model, classification rules.

1 Introduction

This paper examines the electrical activity model of a giant squid axon based on the Hodgkin-Huxley mathematical model [1-7]. When studying the excitability of the axon through the construction of classification rules of the type of excitability, the initial conditions are calculated.

Copyright © 2019 for this paper by its authors.

Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

The systems of ordinary differential equations of mathematical biology use velocity constants and initial values as parameters. In this paper, we present an automated multivariate method that determines the conditions for parameter combinations to obtain a particular type of neuron excitability in the Hodgkin-Huxley model.

At work [8] proposed a multivariate approach based on a decision tree that investigates only the influence of initial conditions in ODE (systems of ordinary differential equations) at fixed velocity constants. In real models, it is rarely possible to use or directly measure the values of velocity constants in the literature. As a rule, they are estimated based on time series [9] based on solution trajectories [10].

2 Development of a multivariate method of qualitative analysis of the Hodgkin-Huxley model

2.1 The objective of the method is to establish mechanisms of multiparameter effects in the Hodgkin-Huxley model.

General ideas of the method were developed in [8] for the case of the initial conditions of the ODE. In this paper, it will also be developed for the velocity constants of ODE. In doing so, we use the Monte Carlo approach, which consists in the random generation of parameters and the construction of the ODE model based on them. Next, we apply the sequential coverage algorithm to construct the classification rules.

The model of electrical activity of a giant squid axon, proposed in [11], is considered. In the model, each component of the excitatory cell is considered as an electrical element. The lipid layer is represented as a container C_m . Ion channels are represented by electrical conductivity g_i , where i is a specific ion channel that depends on both voltage and time. Ion pumps are a current source I_{app} . Denote by V the difference between the membrane potential and the residual potential.

The current through the bilipid layer will be:

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

The current through a given ion channel will be:

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

where V_i – is the equilibrium potential of the i -th ion channel.

For cells with potassium, sodium and chlorine channels, the total current through the membrane I will be:

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

The final typical Hodgkin-Huxley model is:

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

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

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

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

A system based on (1) - (4) is assumed to exist at initial values and velocity parameters at specified intervals.

Options

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

and the initial conditions

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

We will then randomly generate the initial and velocity parameters that belong to a practically valid region. For each of the sets of such parameters, the system (1) - (4) is integrated to obtain the corresponding trajectories. The results are further applied by the sequential coverage algorithm to find specific patterns for decision making.

So, as a whole, the following five steps are taken.

1. **Definition of classes of trajectories of the system.** It should be noted that in practical applications they are mostly dealing with much more complex behaviors in order to characterize their concepts as “stable-unstable” and accordingly to resort to the analysis of eigenvalues or Lyapunov exponents of the dynamic system. There-

fore, it is advisable to transfer the definition of qualitative forms of trajectories to the competence of expert physiologists. In this case, we will use classes related to the types of neuronal excitability [12]: type I, type II, type III. To denote a trajectory class C , a class attribute is introduced that takes one of 3 discrete values $C \in \overline{1,3}$. Figures 1-3 show typical representations of 3 classes of trajectories – types of neuron excitability when the applied current increases:

- type I

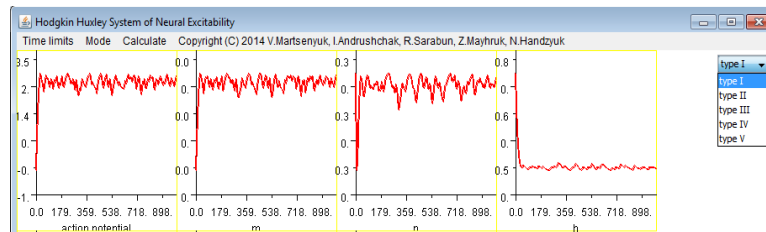


Fig. 1. Neuronal excitability and type I.

- type II

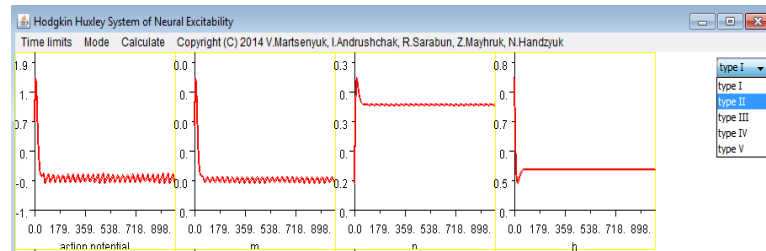


Fig. 2. Neuronal excitability and type II.

- type III

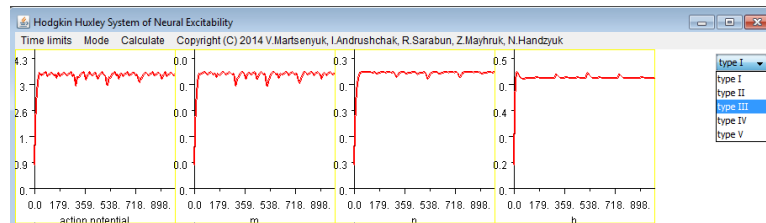


Fig. 3. Neuronal excitability and type III.

2. Generation of a matrix of random initial values and velocity parameters. In order to explore the entire space of initial values and velocity parameters with respect to the generation of trajectory classes defined in the first step, a matrix of random ini-

tial values and velocity parameters is generated based on probability distributions at defined intervals. In this paper, we assume that the initial values and velocity parameters are evenly distributed at intervals. Each column corresponds to a set of values of one parameter – either the initial value or the speed parameter. Each row is a set of initial values and velocity parameters for one run of the model based on the ODE:

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

3. **Run the model and classify the input set.** Each set of initial values and velocity parameters generated in the second step is used as input for the Hodgkin-Huxley model. The numerical integration of the equations is carried out using the Adams method [13]. Output trajectories are classified on the basis of the criteria proposed in the first step. Based on the classification results, sets of initial values and velocity parameters are assigned to the corresponding class attributes.
4. **Construction of a matrix of relationships between initial values and between velocity parameters.** The method assumes that for the shape of the trajectories of the system, the relationship between the initial values and between the velocity values is much more important than their absolute values. Therefore, a matrix is constructed that includes information in categorized coded form about the relationship between the initial values and between the velocity parameters generated in step 2:

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

Here $x(u, v) = p(u, v) = \begin{cases} 0, & \text{if } u < v, \\ 1, & \text{if } u = v \\ 2, & \text{if } u > v \end{cases}$, $C_i \in \overline{1, 3}$ – values of the class attribute associated

with the corresponding trajectory forms.

Therefore, in this step, the numerical values of the initial values and velocity parameters are transformed into categorical values of the attributes of the training data sets. Since the probability of equality of random numbers is zero, the matrix D looks like a "binarization" of the relations between the initial values and between the velocity parameters. That is, the matrix D will include only the values 0 and 2.

5. **Application of sequential coverage algorithm to correlation between initial values and between velocity parameters.** The binary ratio matrix D constructed in step 4 is the training data set for the sequential coverage algorithm. A built-in set of

classification rules will contain a check of the relation between the initial values and the velocity parameters in antecedents. As a consequence of the rules will be the trajectory classes of the model $C \in \overline{1,3}$.

3 Software implementation of the multivariate method for the Hodgkin-Huxley model

A rule.model Java-package was developed to implement the method rule.model. The package includes classes (fig.4): beans-classes Attribute, Attribute_for_list to work with the data in the corresponding tables and Rule to represent the rules. SQL queries to obtain the relevant data are implemented in the AttributeListPeer and TuplesPeer classes [14].

The Rule_set class stores a set of training rules. In addition, this class implements the sequential coverage algorithm directly. The class contains members: the m_dataManager data manager, the hash tables of the m_htTuples training datasets, all the attributes with their possible m_htAtt_vals values, and directly a set of m_htRule_set rules.

The Rule_set class constructor constructs the hash tables m_htTuples and m_htAtt_vals, as well as applying the sequential coverage algorithm by calling the Sequential_covering method (m_htTuples, m_htAtt_vals). The resulting set of rules is output to a text file.

The Rule class is designed to store individual rules. Its class members are two hash tables: m_htAntecedent – for storing the antecedent of the rule and m_htConsequent - for storing the rule. Using the method

```
public void conjunctCondition(Attribute_for_list attribute, String sAttribute_value)
the new condition is conjugated to the rule. Using the method
public Rule copy()
```

the new condition is conjugated to the rule. Using the method

a "deep" copy of the rule is created. This uses the JOS (Java Object Serialization) protocol.

The number of positive and negative training sets is calculated using the methods of the TuplesPeer class.

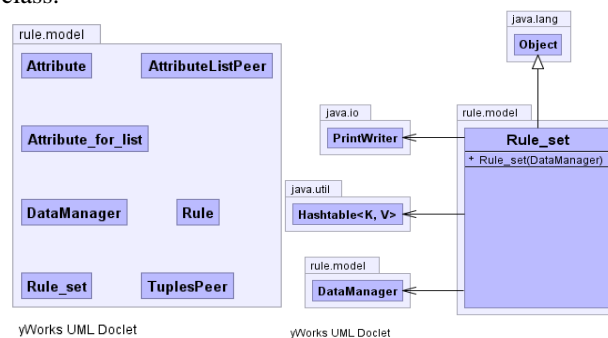


Fig. 4. Package karule.model.

In the `fde.hh.MultiVariateMethod` class (Fig. 5), random parameter values are generated (step 2):

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

Next, the Hodgkin-Huxley system integration applet is launched. The expert chooses the shape of the obtained trajectory (step 3). After that, the step of generating the parameter matrix is started (step 4). Note that the sequence of steps 2-4 can be performed as many times as desired. At any time, the user can run the sequential coverage algorithm (step 5):

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

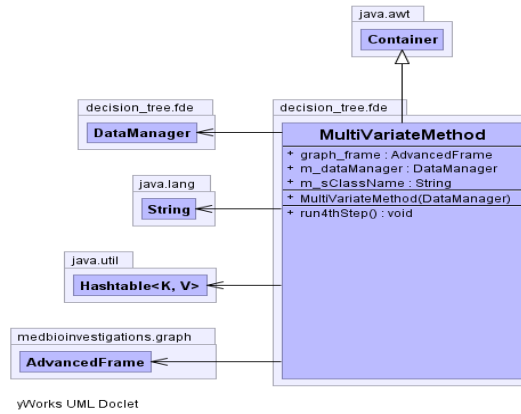


Fig. 5. UML-class diagram MultiVariateMethod.

The `fde` database used in the package is implemented in MySQL DBMS. It includes the following tables (Fig. 6):

- attribute – a description of the attributes used to construct the classification rules, that is, the relationship between the initial values and the velocity constants;
- categorized_data – training kits used in the classification algorithm (in this case, the sequential coverage algorithm) and represent the matrix in the fourth step;
- init_values_values – matrix of randomly generated initial values:

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

initial_values – description of initial values (including minimum and maximum values);

parameter_kind – type of parameter;

rate_constants – description of velocity constants (including minimum and maximum values);

rate_constants_values – matrix of randomly generated velocity constants:

$$\left(\begin{array}{cccccccccc} g_K^1 & g_{Na}^1 & g_L^1 & V_K^1 & V_{Na}^1 & V_L^1 & C_m^1 & x_m^1 & x_n^1 & x_h^1 \\ g_K^2 & g_{Na}^2 & g_L^2 & V_K^2 & V_{Na}^2 & V_L^2 & C_m^2 & x_m^2 & x_n^2 & x_h^2 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ g_K^N & g_{Na}^N & g_L^N & V_K^N & V_{Na}^N & V_L^N & C_m^N & x_m^N & x_n^N & x_h^N \end{array} \right) \in R^{N \times 10}$$

3.1 Features of software implementation of the Hodgkin-Huxley model

The rule.model package can be used for a wide range of functional-differential equation systems. For this purpose, the model based on functional-differential equations should be implemented in the form of a corresponding package of Java classes. An example of such a package in the case of the Hodgkin-Huxley model is the medbioinvestigations.hodgkin_hyxley package (Fig. 7).

To integrate with the decision_tree.fde.hh package in the Hodgkin_HuxleyGraph class, which renders the model graphically, a new constructor was added to the existing one, using the instance reference to the MultiVariateMethod instance. This constructor additionally creates a JComboBox class object that lets you choose the shape of the trajectory and run the method step 4:

```
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();
    }
});
```

You should also make appropriate changes to the hh database table in the following order:

- describe all initial values and velocity constants in the tables initial_values and rate_constants respectively;
- describe the dependencies between the initial values and the velocity constants to be examined in the attribute table;
- in the categorised_data table, create fields according to the attribute table data.

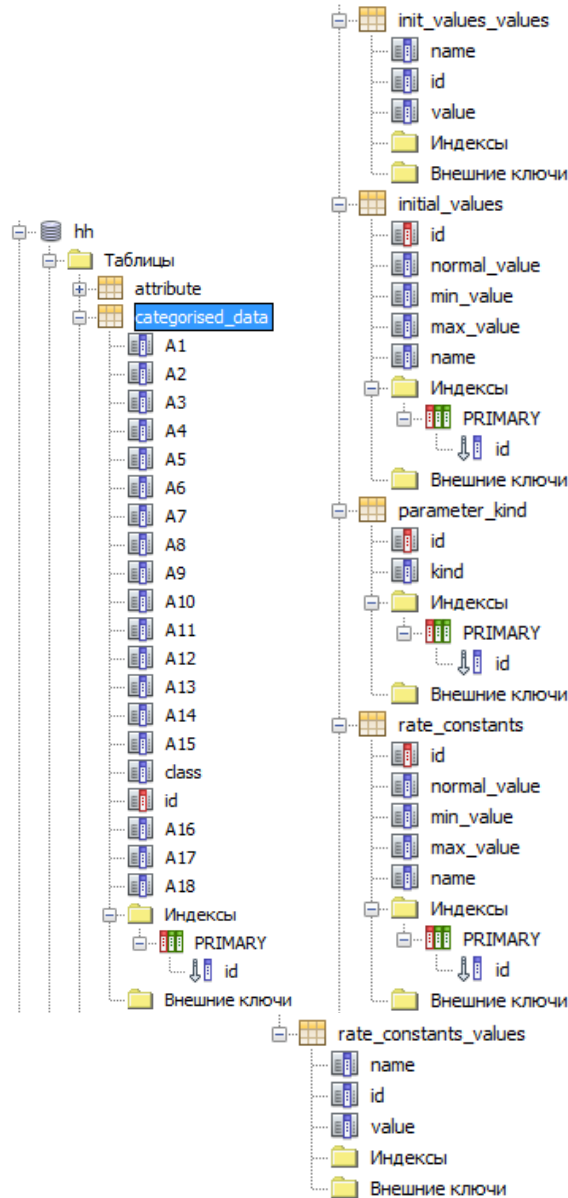


Fig. 6. Database Tables hh.

3.2 Estimation of complexity of algorithm execution

From the analysis of the sequential coverage algorithm, we see that the computational complexity is determined by the product of the number of possible values of the class attribute $K = 3$ (the number of iterations of the outer loop) and the computational

complexity of the *Get_one_rule* (D, Att_vals, c) procedure that is performed inside each cycle.

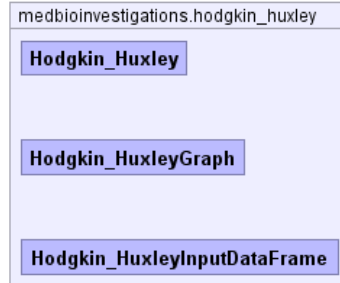


Fig. 7. Packet medbioinvestigations.hodgkin_huxley.

The *Get_One_Rule* (D, Att_vals, c) procedure involves performing $p = 18$ iterations. At each iteration for a particular attribute A_i , a measure is calculated *FOIL_Gain* for each of the $K_i = 2$ attribute values. That is, the inner body of the loop in the *Get_one_rule* (D, Att_vals, c) procedure $\sum_{i=1}^p K_i = 36$ is executed once. The measure *FOIL_Gain* is calculated as a result of 3 – SQL-queries that can be evaluated with complexity $O(\log(N))$ (see MySQL 5.0 documentation - <http://dev.mysql.com/doc/refman/5.0/en/select-speed.html>). Therefore, in general, the *Get_one_Regulation* (D, Att_vals, c) procedure has computational complexity $O\left(\sum_{i=1}^p K_i \times \log(N)\right) = O(36 * \log(N))$.

To sum up, we have the computational complexity of the entire sequential order algorithm

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

A numerical experiment was conducted on the basis of model (1) - (4) to find out the consistency of the time of construction of a set of classification rules with an estimate (6). A system based on the Celeron (R) Dual-Core CPU T3300 @ 2.00 GHz and 2 GB RAM was used. The results are presented in Fig. 8.

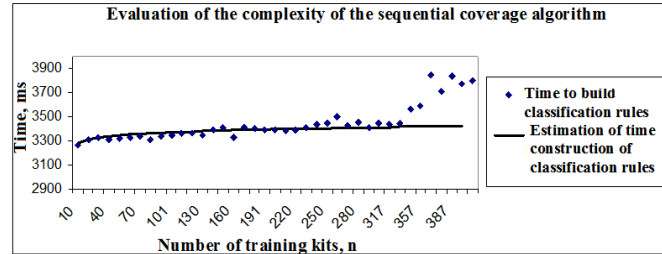


Fig. 8. Difficulty of sequential covering algorithm.

Figure 8 shows a significant deviation from the estimation time of the construction of classification rules $N > 350$, which is associated with an increase in computing resources.

A set of classification rules for $n = 397$ below.

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

Some of the conditions in the antecedent have some biological meaning.

4 Conclusions

There are many implementations of sequential covering algorithm constructing classification rules, in particular as part of programs Mathcad, StatSoft Statistica, etc. A significant advantage of this implementation is that the algorithm developed for the Internet project, where the formation of learning sets involved experts from Internet access, an approach to select such experts with a “close” opinion is described in [15, 16], that extends and balancing set of training sets.

References

1. Hassard, B.: Bifurcation of periodic solutions of the Hodgkin-Huxley model for the squid giant axon, *J. Theoret. Biol.* (71), 401 - 420 (1978).
2. Fukai, H., Nomura, T., Doi, S., Sato, S.: Hopf bifurcations in multiple-parameter space of the Hodgkin-Huxley equations, I, II, *Biol. Cybern.*, (82), 215 - 222; 223 - 229, (2000).
3. Guckenheimer, J., Labouriau, I. S.: Bifurcation of the Hodgkin-Huxley equations: A new twist, *Bull. Math. Biol.*, (55), 937 - 952 (1993).
4. Hassard, B., Shiau, L.-J. A special point of Z2-codimension three Hopf bifurcation in the Hodgkin-Huxley model, *Appl. Math. Lett.*, (9), 31 - 34 (1996).
5. Labouriau S.: Degenerate Hopf bifurcation and nerve impulse II, *SIAM J. Math. Anal.*, (20), 1 - 12 (1989).
6. Martsenyuk, V., Warwas K., Augustynek, K., Klos-Witkowska, A., Karpinskyi, V., Klymuk N., Mayhruk Z.: On Multivariate Method of Qualitative Analysis of Hodgkin-Huxley Model with Decision Tree Induction, International Conference on Control, Automation and Systems. IEEE Catalog Number: CFP1610D-USB, ISBN: 978-89-93215-12-0 ISSN: 2093 - 7121, 489 - 494 HICO, Gyeongju, Korea(2016).

7. Martsenyuk, V. P., Mayhruk, Z. V.: Construction of optimal bifurcation control in the Hodgkin-Huxley model based on the maximum principle. *Scientific journal of the V. M. Glushkov Institute of Cybernetics of the National Academy of Sciences of Ukraine Kamianets-Podilsky National University named after Ivan Ogienko Mathematical and Computer Modeling Series: Technical Sciences* (9), 78 - 90 (2014).
8. Koch, Y., Wolf, T., Sorger, P., Eils, R., Brars B.: Decision-Tree Based Model Analysis for Efficient Identification of Parameter Relations Leading to Different Signaling States, *www.plosone.org*, **8**(12), e82593 (2013).
9. Kovalenko, I., Davydenko, Ye., Shved, A.: Development of the procedure for integrated application of scenario prediction methods. *Eastern-European Journal of Enterprise Technologies*, **2/4**(98), 31–37 (2019). <https://doi.org/10.15587/1729-4061.2019.163871>
10. Martsenyuk, V., Sverstiuk, A., Andrushchak, I. Approach to the Study of Global Asymptotic Stability of Lattice Differential Equations with Delay for Modeling of Immunosensors. *Journal of Automation and Information Sciences*, **48**(8), 58 - 71 (2019).
11. Hodgkin, A. L.; Huxley, A. F.: A quantitative description of membrane current and its application to conduction and excitation in nerve. *The Journal of physiology* **117**(4): PMC 1392413. PMID 12991237, 500 - 544 (1952).
12. Clay, J. R. et al.: A simple modification of the Hodgkin and Huxley equations explains type 3 excitability in squid giant axons, *J. R. Soc. Interface* (5), 421 - 1428 (2008).
13. Hairer, E., Nersett, S., Vanner, G.: *Solution of ordinary differential equations. Tough tasks.* - M.: Mir, 512 (1990).
14. Fisun, M., Dvoretzkyi, M., Shved, A., Davydenko, Y.: Query parsing in order to optimize distributed DB structure. In: 9th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS), IEEE Press, Bucharest, Romania, 172 - 177 (2017). <https://doi.org/10.1109/IDAACS.2017.8095071>.
15. Kovalenko, I., Davydenko, Y., Shved, A.: Formation of consistent groups of expert evidences based on dissimilarity measures in evidence theory. In: 14th International conference on Computer sciences and Information technologies (CSIT 2019), 113 - 116, IEEE Press, Lviv, Ukraine (2019).
16. Shved, A., Kovalenko, I., Davydenko, Y.: Method of Detection the Consistent Subgroups of Expert Assessments in a Group Based on Measures of Dissimilarity in Evidence Theory. In: Shakhovska, N., Medykovsky, M. O. (eds) *Advances in Intelligent Systems and Computing IV. Selected Papers from the International Conference on Computer Science and Information Technologies (CSIT 2019)*. Springer International Publishing (2020). https://doi.org/10.1007/978-3-030-33695-0_4.