Міністерство освіти і науки України
Харківський національний університет імені В. Н. Каразіна

ВІСНИК

Харківського національного університету імені В.Н. Каразіна

Серія
«Математичне моделювання.
Інформаційні технології.
Автоматизовані системи управління»

Випуск 59

Серія заснована 2003 р.

BULLETIN

of V.N. Karazin Kharkiv National University

Series
«Mathematical Modeling.
Information Technology.
Automated Control Systems»

Issue 59

First published in 2003

Харків
2023

Статті містять дослідження у галузі математичного моделювання та обчислювальних методів, інформаційних технологій, захисту інформації. Висвітлюються нові математичні методи дослідження та керування фізичними, технічними та інформаційними процесами, дослідження з програмування та комп'ютерного моделювання в наукоємних технологіях.

Для викладачів, наукових працівників, аспірантів, працюючих у відповідних або суміжних напрямках.

Наказом Міністерства освіти і науки України від 17.03.2020 № 409 наукове фахове періодичне видання Вісник Харківського національного університету імені В.Н. Каразіна серія «Математичне моделювання. Інформаційні технології. Автоматизовані системи управління» включено до Категорії «Б» Переліку наукових фахових видань України за наступними спеціальностями: 113 – Прикладна математика;122 – Комп'ютерні науки та інформаційні технології; 123 – Комп'ютерна інженерія; 125 – Кібербезпека.

Затверджено до друку рішенням Вченої ради Харківського національного університету імені В. Н. Каразіна (протокол № 18 від 30.10.2023 р.)

Адреса редакційної колегії: 61022, м. Харків, майдан Свободи, 6, ХНУ імені В. Н. Каразіна, к. 534.
Тел. +380 (57) 705-42-81, Email: journal-mia@karazin.ua.

Мова публікації: українська, англійська.

Статті пройшли внутрішнє та зовнішнє рецензування.

Свідоцтво про державну реєстрацію КВ № 21578-11478 Р від 18.08.2015.

The articles are present research in the field of mathematical modeling and computing methods, information technologies, information security. New mathematical methods of research and management of physical, technical and information processes, research on programming and computer modeling in science-intensive technologies are covered.

For teachers, researchers, graduate students working in relevant or related fields.

By the order of the Ministry of Education and Science of Ukraine from 17.03.2020 № 409 scientific professional periodical Bulletin of V.N. Karazin Kharkiv National University series "Mathematical modeling. Information Technologies. Automated control systems" is included in Category "B" of the List of scientific professional publications of Ukraine in the following specialties: 113 – Applied Mathematics, 122 – Computer Science and Information Technology; 123 – Computer engineering; 125 – Cybersecurity.

Approved for publication by the decision of the Academic Council of V.N. Karazin Kharkiv National University (Minutes № 18 of 30.10.2023).

Editorial Address: 61022, Kharkiv, Svobodi sq., 6, V.N. Karazin Kharkiv National University, r. 534.
Phone. +380 (57) 705-42-81, Email: journal-mia@karazin.ua.

Language of publication: Ukrainian, English.

The articles pass internal and external review.

Certificate of state registration: KV № 21578-11478P dated 18.08.2015

ЗМІСТ

CONTENTS

DOI: https://doi.org/10.26565/2304-6201-2023-59-01

УДК (UDC) 519.216

Danilevskyi Mykhailo *PhD student; V.N. Karazin Kharkiv National University, Svobody Square, 4, Kharkiv-22, Ukraine, 61022.*
e-mail: m.danilevskiy@gmail.com
https://orcid.org/0009-0000-0030-2218

Yanovsky Volodymyr *Doctor of Physical and Mathematical Sciences, professor; V. N. Karazin Kharkiv National University, sq. Svobody 4, Kharkiv, Ukraine, 61000; Institute of Single Crystals, National Academy of Sciences of Ukraine, Nauki Ave. 60, Kharkiv, Ukraine, 61001*
e-mail: yanovsky@isc.kharkov.ua
https://orcid.org/0000-0003-0461-749X

Modeling and Analyzing the Simplest Network of Telephone Subscribers

Abstract. Dynamic networks such as social, transport and biological networks are widely represented in the modern world. Modeling complex networks as time-varying structures opens up additional opportunities for studying their properties.
Purpose. The goal of the work is to model the simplest dynamic network of telephone subscribers. The main focus is on experiments with the resulting model and studying how the number of subscribers influences the network properties.
Research methods. The work uses the Monte Carlo method of stochastic dynamics of discrete states using time steps of the same length, as well as the methods for constructing computer models, the methods for analyzing the properties of networks, the least squares method and others. The computer model has been developed in Python using the Pandas, Numpy and NetworkX libraries.
Results. The simplest model of a network of telephone subscribers has been designed, where subscribers are connected randomly and disconnected after the phone conversation. In the model, the average daily number of outgoing calls from subscribers is distributed according to the lognormal law. The experiments have been carried out with different numbers of subscribers, but for the same time period. Based on the data obtained from the experiments, we analyzed such network properties as number of connections, density, degree distribution, average clustering coefficient, and average shortest path length.
Conclusions. The developed computer model of the simplest dynamic network of telephone subscribers forms a model similar to a random Erdes-Rényi graph, but the degrees of the vertices or the number of connections between subscribers are distributed according to a lognormal law. The developed computer model can serve as the basis for the development of more complex models and the study of the dynamic properties of such networks.
Keywords: *dynamic complex network, mobile call graph, telephone network, lognormal distribution, degree distribution, network density, clustering coefficient, average shortest path length.*

How to quote: Danilevskyi M., Yanovsky V., "Modeling and Analyzing the Simplest Network of Telephone Subscribers" *Bulletin of V.N. Karazin Kharkiv National University, series Mathematical modeling. Information technology. Automated control systems*, vol. 59, pp. 6-15, 2023. https://doi.org/10.26565/2304-6201-2023-59-01

Як цитувати: Danilevskyi M., Yanovsky V. Modeling and Analyzing the Simplest Network of Telephone Subscribers. *Вісник Харківського національного університету імені В.Н. Каразіна, серія Математичне моделювання. Інформаційні технології. Автоматизовані системи управління*. 2023. вип. 59. С.6-15. https://doi.org/10.26565/2304-6201-2023-59-01

1. Introduction

Research of the social networks has been developing for many decades. The psychiatrist Jacob Moreno, who became interested in the dynamics of social interactions within groups in the 1930s [1], is usually considered as the founder of the field. "A social network is any network in which nodes represent people and edges represent some form of connection between them, such as friendship" [2]. In this paper, we create a model and study a dynamic network of telephone subscribers or a mobile call graph. This network can be seen as a social network in which contacts are made via a telephone. The peculiarity of the network connected by the telephone calls is that connections are severed after the end of telephone conversation between subscribers. At any moment in time, only pairwise connections exist, but a fully connected network can be observed taking into account the connections that existed over a certain period of time. Thus, in each period there are different configurations of connections between subscribers, and accordingly, the network of telephone subscribers can be considered a dynamic one.

Most networks, including a telephone subscriber network, change over time. However, researchers often view networks as static entities. In some cases, this may be a reasonable approximation, but we can learn a lot by observing, analyzing and modeling network changes over time. For example, in some networks only connections may change and nodes remain permanent, in others, nodes may appear or disappear. One of the approaches to modeling social networks, including networks of telephone subscribers, is random graphs. The random graph model has been introduced in 1950-1960 by Paul Erdős and Alfred Rényi [3]. A random graph is a network model in which the values of certain properties are fixed, but other properties are random. For instance, a graph with a fixed number of nodes and edges is one of the simplest models in which edges between nodes are randomly assigned. This approach is quite primitive and has a number of disadvantages, for example, the lack of correlation between the degrees of neighboring nodes, and the degrees of nodes are distributed according to Poisson's law, which is rarely the case in real networks. Additionally, in real networks structures called communities are formed, in random networks they are absent. The next stage in the development of models of complex networks is the Watts-Strogatz model [4]. This model creates a network with the properties of a "small world" – a high clustering coefficient and a short length of the average shortest path between the vertices of the network. As a result of the analysis of the Internet network topology by Albert and Barabasi [5], it has been discovered that the distribution of vertex degrees follows a power-law. Such networks are called scale-free networks. It has been empirically established that social, communication, biological, citation graphs, Internet links, and other systems could be sufficiently modeled by scale-free graphs. It is believed that the most important characteristics of social networks, in addition to high clustering, short average paths and the presence of communities, include assortative mixing and a wide degree distribution. In [6], the authors have presented a model of such a network in which new vertices are added both to random vertices and to neighboring ones, which leads to implicit preferential joining.

Real world networks can have billions of users, and sometimes it is even impossible to determine their size. In such conditions, network modeling is an effective way to study complex networks. It helps to get an idea about the network structure, understand how the network changes when its parameters change, and also study the processes occurring in networks, for example, the dissemination of information. Dynamic network models can be divided into two groups: deterministic and stochastic. Stochastic models take into account the random nature of network parameters, and therefore are better suited for modeling social networks. For example, in [7], the authors have noted the limitations of the analytical approach and created a stochastic model of the telephone network using the GPSS/H language. Using the created model, the authors have determined the operational characteristics of the existing corporate telephone network and assessed the performance of the designed networks before their installation. In [8], a discrete event simulator for communication networks OSSIm is presented. A brief but comprehensive overview of graph modeling of complex communication networks and their application to social network analysis is provided in [9]. Based on various models of physical networks, the functions for generating artificial social networks have been created in Matlab [10], thus adding a social component to the models of physical networks. Currently, complex networks can be analyzed and modeled by describing models in programming languages, including C++, Python, R, as well as using specialized software packages such as NetworkX [11] and NetworkKit [12]. In works [13-16], the authors analyze the structure and dynamics of social networks in which connections between people are carried out by using a mobile communication network.

In this work, we present a simple model of social network in which connections between people are established via telephone calls. Based on the data obtained after experiments, such network properties as the number of edges, density, degree distribution, average clustering coefficient, and average shortest path length have been analyzed.

2. Modeling dynamic network of telephone subscribers

In this work, we present a model of a social network in which connections between people are established via telephone calls. A network of telephone subscribers is characterized by a certain number of nodes (subscribers) and connections between them. Connections in such network occur when one subscriber contacts another, and the duration of their contact is greater than 0. The number of connections or call attempts is varied for different types of subscribers. For instance, usually people call 5-7 times a day, while sales managers may call 200-300 times a day. As a result, at each moment of time a certain number of pairwise connections exist. These connections cannot be considered a network. Only if we increase the time scale, it is possible to observe a network formed during a given period of time.

As a result of computer modeling, a simple model of a telephone network has been developed. The main simplification is that subscribers choose to contact each other randomly. The number of calls over a period of time is assigned to each subscriber according to the lognormal distribution. While subscribers are in contact, they cannot make or receive calls. The call duration is modeled by the probability of ending a call at a given time.

The input parameters of the model are the following: number of subscribers, parameters of the lognormal distribution for assigning the number of calls per period of time, duration of the experiment.

3. Description of the experiment

Conducting experiments involves simulating the network of telephone subscribers over a certain period of time. The number of subscribers is fixed and set beforehand. Time in this model is discrete, one unit of time is considered to be one minute. If a subscriber calls 5 times a day, then the probability of a call at any minute is defined as 5 divided by 1440 and therefore equals to 0.00347. The experimenter sets the number of subscribers, duration of the experiment, as well as the parameters of the lognormal distribution for modeling number calls per day for each subscriber. The subscriber selects a subscriber randomly according to a uniform distribution law. The probability of a call ending at any given time is 0.99, which determines its duration. If subscribers have established a connection, they are blocked and can neither make nor receive calls. The progress of the experiment is recorded in the table:

Table 1. Experiment data sample

idx	caller_id	callee_id	start_time	finish_time
0	67	22	0	1
1	49	64	2	3
2	44	54	3	4

idx is the record number, caller_id is the caller identifier, callee_id is the callee identifier, start_time is the call start time, finish_time is the call finish, duration is the call duration.

The input data for the experiment are: the number of subscribers is 1000, the duration is 10080 time units (minutes) or 7 days, the parameters of the lognormal distribution correspond to normal subscribers ($\mu = 1.1, \sigma = 1.0$) [17].

4. Study of the obtained data

As a result of running a model, subscribers made 34834 contacts between each other. Based on this data, we examine the distribution of the number of subscribers by the number of calls per day. The chart (Fig.1) shows a histogram where the circles represent fraction of subscribers who called a certain number of times in one day. This distribution can be compared with the distribution obtained in the work based on the results of experiments, where it has been found that such data are distributed according to the lognormal law [17]. Let us examine how well the model confirms with the experimental data.

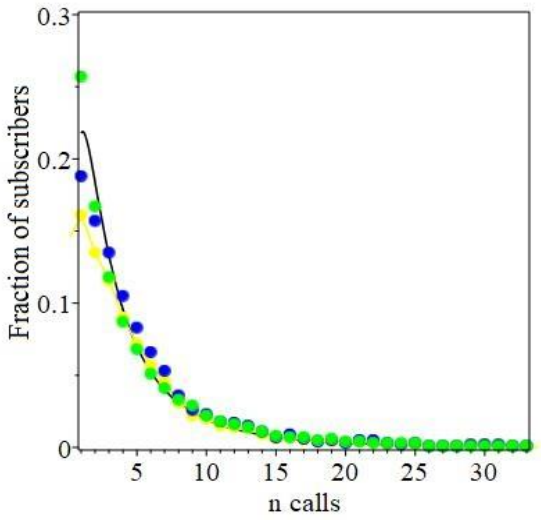


Fig. 1. Distribution of the number of subscribers by the number of calls per day. Green - experimental data from the real network. Simulation data: yellow - all subscribers are included, blue - only those subscribers who made at least one call. The lognormal law is shown as a black curve.

The green line corresponds to experimental data from the telephone network of subscribers [17]. The data obtained after experiments with the model is depicted by the blue and yellow circles. The distribution of subscribers corresponding to the blue line has been calculated without taking into account subscribers who made at least one call during the experiment. The distribution of subscribers corresponding to the yellow line has been calculated taking into account subscribers who did not make a single call during the calls simulations. The lognormal distribution (black curve) demonstrates the agreement between the simulation data and the experimental observations.

The experimental data include subscribers who made at least one call during the experiment, i.e., there are no subscribers without calls. There are such subscribers in the simulated network, which corresponds to the distribution shown by the yellow line. As a result, differences in distributions arise.

As can be seen from Fig. 1, the blue curve – the distribution of subscribers who made at least 1 call – is in good agreement with the experimental data (the green curve). Such subscribers form a network of connections over a certain period. For a more detailed study of the properties of the telephone subscriber network, we will use the data obtained as a result of additional experiments with the developed computer model.

5. Analysis of the experiments with the model

To study the properties of the modeled dynamic network, we have conducted additional experiments with a different number of subscribers, and measured the dependencies of such network properties as well as determined the type of the resulting network. We have conducted 50 experiments in which the initial number of subscribers was 10, and in subsequent experiments the number of subscribers increased by 10 up to 500 subscribers. The graphs of 6 networks out of 50 are shown in Fig. 2. The topology of the resulting networks reflects the random nature of the choice of subscribers.

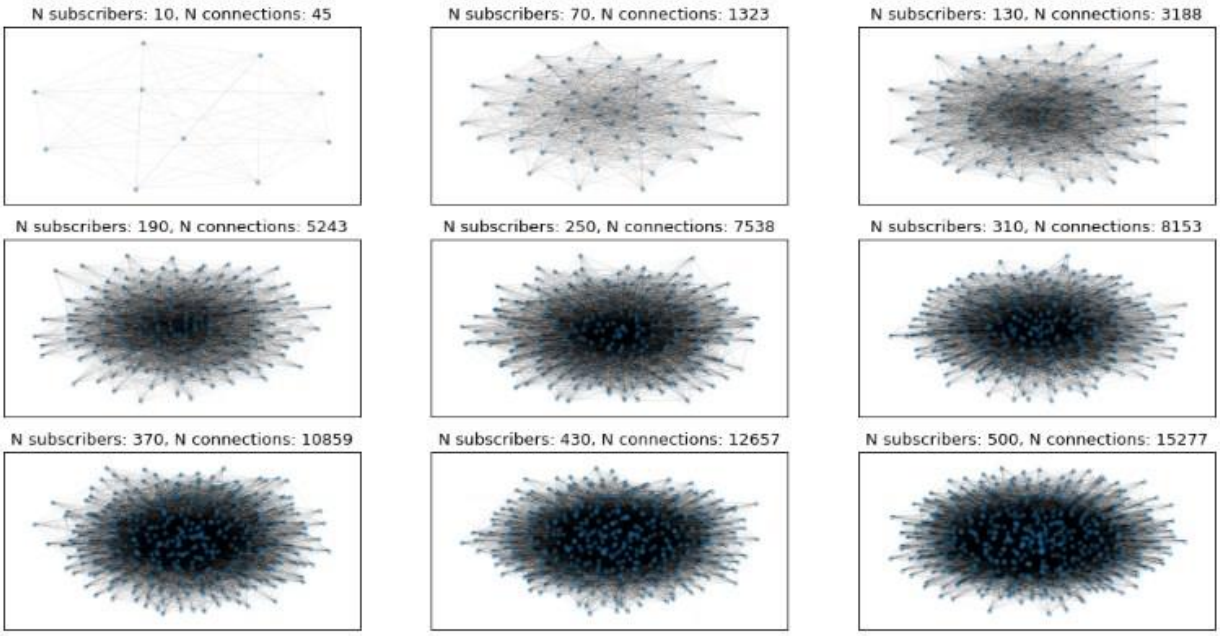


Fig. 2. Graphs of the resulting networks with selected number of subscribers

The number of connections in such networks increases linearly from 45 to 15277 depending on the number of subscribers. As a result of the simulation, we obtained data on the relationship between the number of connections and the number of subscribers. Linear approximation of these data by using the least squares method gives the dependence shown in Fig. 3 with a black line. It is easy to notice good agreement of the data with the given dependence.

$$n = 32 \times (N - 32) \tag{1}$$

where n is the number of connections, N is the number of subscribers.

From this dependence it follows that each new subscriber creates 32 new connections. The deviation is observed in case of a small number of subscribers. There are not yet enough subscribers in this case to establish an asymptotic number of connections.

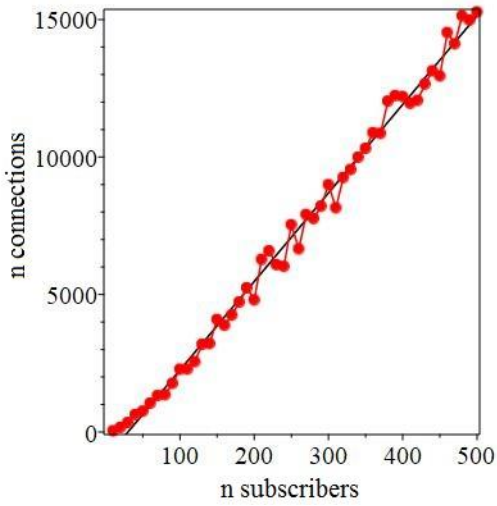


Fig. 3. Relationship between the number of connections and the number of subscribers in simulated networks. Red circles are the simulation data, black ones are the direct approximation.

Network density is another property that we have analyzed, particularly, the changes in network density with an increase in the number of subscribers. Density is the fraction of actually present edges to the maximum possible number of edges. “It can be thought of as the probability that a pair of nodes, picked uniformly at random from the whole network, is connected by an edge” [2]. As a result of the network simulation, the maximum density was 1.00 in a network with 10 subscribers, the minimum 0.12 was observed in a network with 500 subscribers. In Fig. 4 the simulated data is shown in red, and the approximating dependency in black. The data analysis has shown that the network density follows a hyperbolic dependence:

$$\rho = \frac{1.2}{1+\frac{N}{58}} \tag{2}$$

where ρ is the network density, N is the number of subscribers.

Good agreement with the simulation data is evident in Fig. 4. Thus, as the number of subscribers increases, the network density decreases and tends to 0.

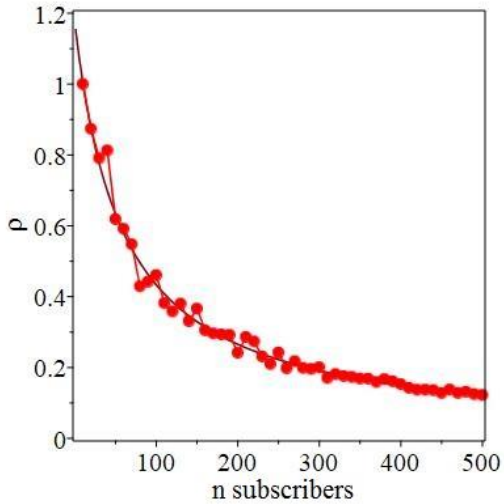


Fig. 4. Relationship between network density and the number of subscribers. Simulation data are in red, hyperbolic dependence curve are in black.

During the experiment, the network subscribers managed to connect with 50-60 unique subscribers in networks with 200 or more subscribers (Fig. 5). This subscriber or graph node property is known as the degree.

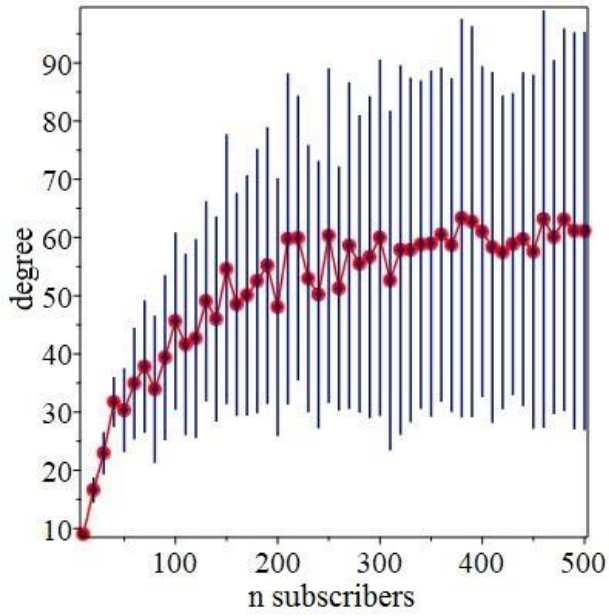


Figure 5. Relationship between the average node degree (number of created connections) and the number of subscribers in the network

Let us consider one of the fundamental properties of the network – the degree distribution. The degree distribution tells us the frequency with which nodes of different degrees appear in the network [2]. As a result of simulating 50 networks, we have obtained the average degree distribution presented in Fig. 6.

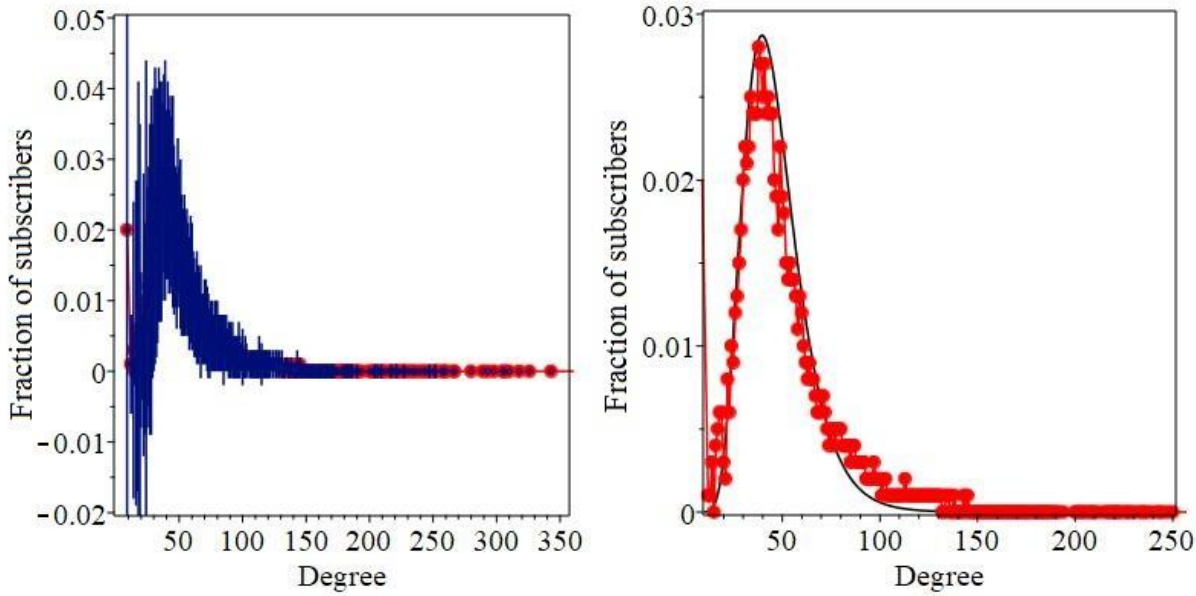


Fig. 6. Distribution of connections established by each subscriber, or node degree distribution. On the left is the distribution of node degrees with standard deviations. On the right is the simulation data in red, the black curve is the lognormal distribution (3) with $\mu = 3.79$ *and* $\sigma = 0.332$. *Good agreement with the simulation data is observed.*

The degree distribution corresponds well with the lognormal distribution with parameters $\mu = 3.79$ and $\sigma = 0.332$. Therefore, we observe that with a random selection of subscribers, the distribution of degrees in the network repeats the distribution of the number of subscribers by the number of calls.

$$f(x) = \frac{e^{-\frac{(\ln(x)-\mu)^2}{2\sigma^2}}}{x\sigma\sqrt{2\pi}} \tag{3}$$

where σ and μ are parameters that determine the lognormal distribution.

The next important network property is the clustering coefficient. The clustering coefficient is the average probability that two neighbors of the same node are themselves neighbors or friends [2]. The higher the value of clustering coefficient, the better information is transmitted over the network. As we can see from the chart in Fig. 7, the clustering coefficient decreases from 1.00 in the fully connected network with 10 subscribers to 0.21 in the network with 500 subscribers.

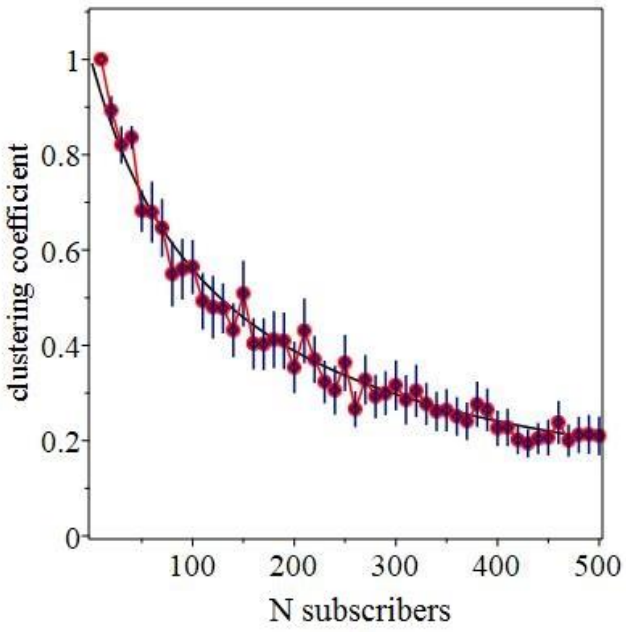


Fig. 7. Relation between the average clustering coefficient and number of subscribers in simulated networks. The red dots are the simulation data, the black curve is the dependence (4).

The simulation data is also well approximated by the hyperbolic dependency, which is shown in Fig. 7 with the black curve. It is clear that as the number of network subscribers increases, the clustering coefficient decreases:

$$C = \frac{1}{1+\frac{N}{127}} \tag{4}$$

where C is the network clustering coefficient, N is the number of subscribers.

Finally, we consider the dependency between the average shortest path and the number of subscribers in the simulated networks, which is presented in the Fig. 8. The lengths of all links between adjacent nodes are considered equal to one. The shortest path in a network, also sometimes called a geodesic path, is the shortest walk between a given pair of nodes, i.e., the walk that traverses the smallest number of edges [2]. In the case of the telephone network and connections between people, this indicator characterizes the average number of contacts required to connect any two subscribers.

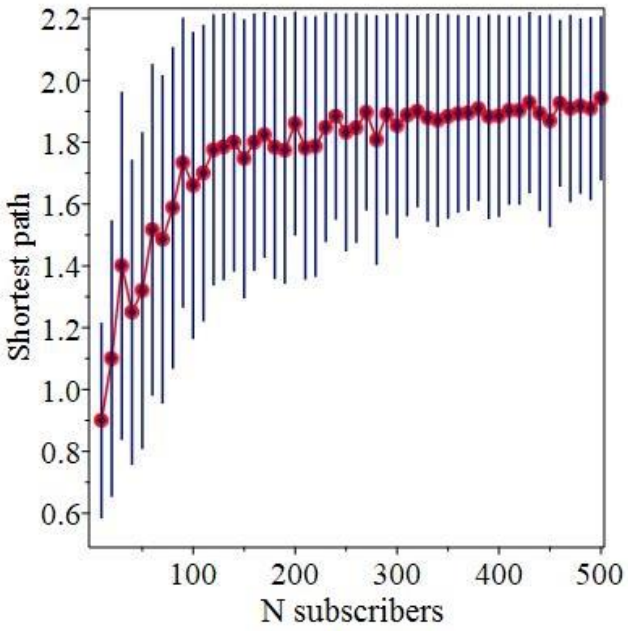


Figure 8. Relation between the average shortest path length and the number of subscribers in networks

In this telephone network model, in which subscribers communicate randomly, the average shortest path is quite low (0.90 – 1.94).

6. Conclusions

In this paper, we propose a simple model of a dynamic network of telephone subscribers. The main simplification of the model is that subscribers choose each other randomly. That leads to some restrictions which determine the network topology and the dynamics of its properties. The basic properties of the network such as the number of connections, density, clustering coefficient, degree distribution and the average shortest path length in the network and their dependence on changes in the number of network nodes or subscribers have been obtained. The modeling results have shown that the number of connections increases linearly with the number of subscribers, which results in 32 connections when a new subscriber appears on the network. Network density decreases with increasing number of subscribers according to the hyperbolic law (2). The network clustering coefficient also decreases with increasing number of subscribers according to a similar hyperbolic law (4). Another important network characteristic that affects the propagation of information is the length of average shortest path between two subscribers. It could be noted that a low average path length of 2 is possible even in a low-density network with the density of 0.12. It may indicate that the network does not need to be fully connected so that subscribers communicate with each other through a small number of contacts. The degree distribution or the number of subscriber connections obtained by the simulation is in good agreement with the lognormal distribution. It is interesting to compare the results obtained with data on real subscriber networks. Due to the random formation of connections, the resulting model creates a network that resembles the Erdes-Rényi random graph model, but the degrees of the vertices are distributed according to the lognormal law.

REFERENCES

1. J. L. Moreno, The first book on group psychotherapy, 3rd ed. Oxford, England: Beacon, House, 1957, pp. xxiv, 138.
2. M. Newman, Networks, vol. 1. Oxford University Press, 2018. DOI: https://doi.org/10.1093/oso/9780198805090.001.0001 (Last accessed: 20.08.2022).
3. P. Erdos and A. Renyi, “On the evolution of random graphs,” Publ. Math. Inst. Hungary. Acad. Sci., vol. 5, pp. 17–61, 1960. URL: https://snap.stanford.edu/class/cs224w-readings/erdos60random.pdf (Last accessed: 20.08.2022).

4. D. Watts and S. Strogatz, “Collective dynamics of ‘small-world’ networks,” Nature, vol. 393, pp. 440–442, 1998, DOI: https://doi.org/10.1038/30918 (Last accessed: 20.08.2022).
5. A.-L. Barabási and R. Albert, “Emergence of Scaling in Random Networks,” Science, vol. 286, no. 5439, pp. 509–512, 1999, DOI: https://doi.org/10.1126/science.286.5439.509 (Last accessed: 20.08.2022).
6. R. Toivonen, J.-P. Onnela, J. Saramäki, J. Hyvönen, and K. Kaski, “A model for social networks,” Physica A: Statistical Mechanics and its Applications, vol. 371, no. 2, pp. 851–860, Nov. 2006, DOI: https://doi.org/10.1016/j.physa.2006.03.050 (Last accessed: 20.08.2022).
7. S. Niaki and Z. B. Rad, “Designing a communication network using simulation,” Scientia Iranica, vol. 11, pp. 165–180, Aug. 2004. URL: https://www.researchgate.net/publication/236107639_Designing_a_communication_network_using_simulation (Last accessed: 20.08.2022).
8. E. Uleia, “Discrete Event Simulator for Communication Networks,” 2007. URL: http://2007.telfor.rs/files/radovi/09_13.pdf (Last accessed: 20.08.2022).
9. B. S. Khan and M. A. Niazi, “Modeling and Analysis of Network Dynamics in Complex Communication Networks Using Social Network Methods,” ArXiv, vol. abs/1708.00186, 2017, URL: https://arxiv.org/ftp/arxiv/papers/1708/1708.00186.pdf (Last accessed: 20.08.2022).
10. T. Johansson, “Generating artificial social networks,” The Quantitative Methods for Psychology, vol. 15, no. 2, pp. 56–74, 2019, DOI: https://doi.org/10.20982/tqmp.15.2.p056 (Last accessed: 20.08.2022).
11. “NetworkX — NetworkX documentation.” URL: https://networkx.org/ (Last accessed: 20.08.2022).
12. C. L. STAUDT, A. SAZONOVS, and H. MEYERHENKE, “NetworKit: A tool suite for large-scale complex network analysis,” Network Science, vol. 4, no. 4, pp. 508–530, 2016, DOI: https://doi.org/10.1017/nws.2016.20 (Last accessed: 20.08.2022).
13. G. Miritello, E. Moro, R. Lara, R. Martínez-López, S. G. B. Roberts, and R. I. M. Dunbar, “Time as a limited resource: Communication Strategy in Mobile Phone Networks.” arXiv, Jan. 11, 2013. URL: https://arxiv.org/abs/1301.2464 (Last accessed: 20.08.2022).
14. A. A. Nanavati et al., “On the Structural Properties of Massive Telecom Call Graphs: Findings and Implications,” in Proceedings of the 15th ACM International Conference on Information and Knowledge Management, in CIKM ’06. New York, NY, USA: Association for Computing Machinery, 2006, pp. 435–444. DOI: https://doi.org/10.1145/1183614.1183678 (Last accessed: 20.08.2022).
15. J.-P. Onnela et al., “Structure and tie strengths in mobile communication networks,” Proc. Natl. Acad. Sci. U.S.A., vol. 104, no. 18, pp. 7332–7336, May 2007, DOI: https://doi.org10.1073/pnas.0610245104 (Last accessed: 20.08.2022).
16. M. Zignani, C. Quadri, S. Gaitto, and G. P. Rossi, “Exploiting all phone media? A multidimensional network analysis of phone users’ sociality.” Jan. 14, 2014. Accessed: Oct. 19, 2023. URL: https://arxiv.org/abs/1401.3126 (Last accessed: 20.08.2022).
17. V. Danilevskiy and V. Yanovsky, “Statistical properties of telephone communication network,” arXiv preprint arXiv:2004.03172, 2020. URL: https://arxiv.org/pdf/2004.03172.pdf (Last accessed: 20.08.2022).

Данілевський Михайло Вікторович *аспірант; Харківський національний університет імені В.Н. Каразіна, майдан Свободи, 4, Харків-22, Україна, 61022*
e-mail: m.danilevskiy@gmail.com
https://orcid.org/0009-0000-0030-2218

Яновський Володимир Володимирович *доктор фізико-математичних наук, професор, професор кафедри штучного інтелекту та програмного забезпечення Харківський національний університет імені В. Н. Каразіна, майдан Свободи 4, Харків-22, Україна, 61022 Завідувач теоретичним відділом, інститут монокристалів НАН України, пр. Науки 60, Харків, Україна, 61001*
e-mail: yanovsky@isc.kharkov.ua
https://orcid.org/0000-0003-0461-749X

Моделювання та аналіз найпростішої мережі телефонних абонентів

Актуальність. Динамічні мережі представлені у широкому спектрі областей сучасного світу, включаючи соціальні, транспортні та біологічні мережі. Моделювання складних мереж як структур, що змінюються в часі, відкриває додаткові можливості для вивчення їх властивостей.

Мета. Метою роботи є моделювання найпростішої динамічної мережі телефонних абонентів. Основна увага зосереджена на експериментах з отриманою моделлю та дослідження впливу кількості абонентів на властивості мережі.

Методи дослідження. У роботі використовуються метод Монте-Карло стохастичної динаміки дискретних станів із використанням часових кроків однакової довжини, а також методи побудови комп'ютерних моделей, методи аналізу властивостей мереж, метод найменших квадратів та інші. Комп'ютерна модель розроблена мовою Python із використанням бібліотек Pandas, Numpy та NetworkX.

Результати. Розроблено найпростішу модель мережі телефонних абонентів, у якій абоненти обирають інших абонентів випадковим чином, а зв'язки існують тільки під час телефонної розмови. В моделі середньоденна кількість вихідних дзвінків абонентів розподілена за логнормальним законом. Проведено експерименти з моделлю з різною кількістю абонентів, але за однаковий часовий відрізок. На підставі отриманих даних про дзвінки, розглянуті такі властивості мереж як кількість зв'язків, щільність, розподіл вершин, середній коефіцієнт кластеризації та середня довжина найкоротшого шляху.

Висновки. Розроблена комп'ютерна модель найпростішої динамічної мережі телефонних абонентів формує модель схожу до випадковий граф Ердеша-Реньї, але при цьому ступені вершин або кількість зв'язків абонентів розподілено за логнормальним законом. Розроблена комп'ютерна модель може бути основою розробки складніших моделей та вивчення динамічних властивостей подібних мереж.

Ключові слова: *складна динамічна мережа, граф мобільних викликів, телефонна мережа, логнормальний розподіл, розподіл ступенів, щільність мережі, коефіцієнт кластеризації, середня довжина найкоротшого шляху*.

Надійшла у першій редакції 18.06.2023, в останній - 21.08.2023.
The first version has been received on 18.06.2023, the final version - on 21.08.2023.

DOI: https://doi.org/10.26565/2304-6201-2023-59-02

УДК (UDC) 004.4`6:004.4`4

Deineha Oleksandr *PhD student*
V. N. Karazin Kharkiv National University, 4 Svobody Sq., Kharkiv, 61022, Ukraine;
e-mail: oleksandr.deineha@karazin.ua
https://orcid.org/0000-0001-8024-8812

The Clustering of Lambda Terms by Using Embeddings

Relevance. The importance of optimizing compilers and interpreters for functional programming languages, mainly through the lens of Lambda Calculus, is paramount in addressing the increasing complexity and performance requirements in software engineering. The emphasis of this study lies in this critical area, aiming to leverage advanced machine learning techniques to enhance identification and application of code reduction strategy.

Goal. The primary goal is to improve the performance and efficiency of compilers and interpreters by deepening the understanding of program code reduction strategies within Lambda Calculus. The research is aimed at using machine learning to convert lambda terms into feature vectors, facilitating the exploration of optimal reduction strategies.

Research methods. The study employs a comprehensive approach, generating a wide range of lambda terms for analysis. It utilizes OpenAI's text embedding model to transform these terms into embedding vectors, employing clustering analyses (DBSCAN with Euclidean measurements) and visualizations (PCA and t-SNE) to identify patterns and assess feature separability. The research navigates the complexities of choosing between specific and universal reduction strategies.

The results. Findings have revealed clear distinctions among lambda term representations within the embedding vectors, supporting the hypothesis that cluster analysis can uncover identifiable patterns. However, the challenges have been encountered due to OpenAI Embeddings' training being generally focused on human-readable text and code, and that complicates the precise representation of Lambda Calculus terms.

Conclusions. This exploration underscores the challenges in pinpointing the optimal reduction strategy for Lambda Calculus terms, highlighting the limitations of current mathematical models and the need for tailored machine learning applications. Despite the hurdles with the OpenAI Embeddings model's adaptability, the research offers significant insight into the potential of machine learning to refine the optimization processes of compilers and interpreters in functional programming environments.

Keywords: *Pure Lambda Calculus, Clustering Analysis, Pretrained Embedding, Hidden Space.*

Як цитувати: Дейнега О. А. Кластеризація Лямбда Термів з використанням Вбудувань. *Вісник Харківського національного університету імені В.Н. Каразіна, сер. «Математичне моделювання. Інформаційні технології. Автоматизовані системи управління»*. 2023. вип. 59. С.16-23.
https://doi.org/10.26565/2304-6201-2023-59-02

How to quote: Deineha O., "The Clustering of Lambda Terms by Using Embeddings" *Bulletin of V.N. Karazin Kharkiv National University, series "Mathematical modeling. Information technology. Automated control systems*, vol. 59, pp.16-23, 2023.
https://doi.org/10.26565/2304-6201-2023-59-02

1. Introduction

In the modern world of software engineering, functional programming languages are pivotal, providing sophisticated solutions to intricate challenges [1]. With escalating demands for enhanced performance, the importance of optimizing compilers cannot be overstated. Our study focuses on Lambda Calculus, an essential construct of functional programming languages, to achieve this objective. The aim is to analyze software code to identify the most effective reduction strategy, therefore improving the efficiency of both compilers and interpreters [2].

Lambda calculus is the main framework for our study, enabling the simulation of interpreters and compilers in their task to find the best reduction strategies. By generating a broad spectrum of lambda terms, we establish a solid testing environment to evaluate various methods to enhance normalization quality [2, 3]. The intricate decision-making process, whether to develop the best strategies for each

term or to adopt a universal strategy like "Rightmost Innermost," demonstrates the nuanced comprehension of its complexity.

The primary objective of this research is to enhance the functionality of interpreters and compilers used in functional programming languages, specifically through a detailed examination of lambda calculus. The focus is to increase our knowledge of reduction strategies and to improve the operational efficiency of compilers and interpreters. A novel aspect of our scientific inquiry involves applying advanced machine learning techniques to represent lambda terms as vectors of features; that includes analyzing such vectors for their ability to be separated and comparing these separation methods against a strategy prioritized for its efficiency.

Moreover, we encounter a computational dilemma: if it is better to continuously execute Lambda term reduction using various strategies in parallel, selecting the most suitable one based on a specific criterion, or transform a Lambda term into a more straightforward representation and input it into artificial neural networks (ANNs), thereby determining the optimal strategy.

2. Literature Review and Problem Statement

The use of sophisticated machine learning methods to enhance the efficiency of programming language compilers and interpreters is a concept that has been explored previously [4, 5]. CompilerGym [6] provides a platform for broader compiler research, offering an environment for experimentation without delving into specific optimizations. Most studies on optimizing compilers and interpreters focus on the most widely used object-oriented programming languages [7, 8, 9]. The clustering to identify similarities between functions have been utilized in [7]. The transforming program data by using PCA for the LLVM compiler and implementing optimizations based on expert knowledge has been examined in [8]. The iterative compilation method, testing a limited set of optimization possibilities and demonstrating its effectiveness for optimizing code fragments has been adopted in [9]. Furthermore, the application of reinforcement learning to compiler optimization, employing neural optimization agents to support manually crafted optimization sequences has been explored in [10].

Optimizations for compilers of functional programming languages have been comparatively less investigated. The heap profiling for a functional compiler using hand-crafted logic has been explored in [11]. Similarly, custom optimizations have been applied in [12], focusing on a functional web application. Furthermore, model functional languages have typically concentrated on specific reduction strategies, such as Haskell's call-by-need and call-by-value with unique mechanisms and OCaml's call-by-value.

Given this context, the challenge of optimizing compilers and interpreters for functional programming emerges as a significant area of interest. While machine learning techniques are applicable in optimizing compilers for object-oriented languages, this work seeks to identify features within functional code that could indicate the effectiveness of optimization strategies, using machine learning as a tool for exploration.

In the previous work, we have used a similar approach to clustering analysis of terms by using a large language model, Code BERT [13]. This research has shown us some possible ways of creating meaningful vector representations of vector terms.

3. Research Goals and Objectives

To research further the inner structure of generated lambda terms space, the main aim of this study is to enhance the performance of interpreters and compilers for functional programming languages, with a special emphasis on Lambda Calculus. The primary objective is to deepen our comprehension of program code reduction strategies and to improve the efficiency of both compilers and interpreters. The scientific novelty of this research is utilizing of sophisticated machine learning methods to encode lambda calculus terms into feature vectors, followed by an analysis of these vectors for separability and a comparison of their separation to the prioritization of term strategies.

In line with the main goal, the specific objectives of this research are outlined as follows:

- To conduct a clustering analysis of the lambda terms dataset to identify patterns or potential for data division.
- To evaluate the effectiveness of OpenAI's text embedding model in extracting features from lambda terms.
- To investigate the relationships between the extracted features and the identified clusters.

4. Materials and Methods of Research

The objective of this study is to refine the functionality of existing interpreters and compilers for functional programming languages. Lambda Calculus has previously been recognized as a basic model of functional programming languages [2, 3]. It facilitates the simulation of interpreters or compilers to select the most effective reduction strategies. Moreover, it provides a way for the synthetic generation of a wide array of lambda terms, enabling accurate evaluation of the strategies. Selecting an optimal reduction strategy involves devising a unique strategy for each term or applying a specific strategy, such as Rightmost Innermost, for the whole term reduction process. Both methodologies have been explored. The former strategy allows for the creation of a greedy algorithm that selects the best redex for reduction at any given moment. In this context, we have evaluated the disparity in redexes, indicating the varying resources needed for their reduction, with computational complexity as the measure of this disparity, gauged by the time taken per reduction step [3]. Additionally, our research aimed to predict the number of lambda term reduction steps by using a particular strategy, employing a simplified representation of terms that keeps only their tree structure, and excluding variable information [14]. For this purpose, standard Artificial Neural Network (ANN) models commonly used in Natural Language Processing have been applied to forecast the number of steps required for specific strategies [14].

Through our experiments, we have observed that certain redexes suggest an inclination towards one or another standard reduction strategy. Nonetheless, the mere presence of these redexes does not guarantee that a term is best suited to be reduced by that strategy. It indicates a need for a deep analysis of the term to determine if a redex indeed suggests a priority for reduction. Previous studies utilized a simplified term representation [3, 14], which was found to lack a qualitative analysis of terms and omitted critical information. Therefore, we have decided to analyze terms while preserving their variable information, potentially enhancing the differentiation of terms according to their preferred reduction strategy. This approach suggests the possibility of identifying terms that require fewer reduction steps under a specific strategy without specifying the exact number of reductions. As one of the most sophisticated and modern approaches to transforming text representations into vectors, we have selected the text-embedding-ada-002 model of OpenAI embedding as one of the most stable and the one that has shown promising results during our research. During our work, we also experimented with other new OpenAI embedding models but focused on the one mentioned above.

4.1. Gathering Feature Representations

Our work focuses on OpenAI embeddings which are one of the most popular and easy-to-use. Such an approach also significantly reduces computational requirements, as on our side we use a simple API that allows us to input text representations of lambda terms and retrieve vectors as an output. All the calculations are done on the OpenAI servers.

What are text embeddings? Text embeddings, as developed by OpenAI, evaluate how closely text strings are related. Common uses for embeddings include:

- Search (ranking search results by how relevant they are to a search term);
- Clustering (grouping text strings based on how similar they are);
- Recommendations (suggesting items that have text closely related to each other);
- Anomaly detection (spotting text strings that do not fit in with the rest due to low relatedness);
- Measuring diversity (examining how varied the similarity among text strings is);
- Classification (assigning text strings to categories they closely align with); The distance between their vectors gauges the degree of similarity between texts; shorter distances suggest more significant similarity in content, while longer distances signify reduced similarity.

Embeddings can serve as versatile encoders for free-text features within machine learning models. By integrating embeddings, the efficiency of any machine learning model can be significantly enhanced, particularly when some of the input data includes free text. Additionally, embeddings can encode categorical features in a machine learning framework. This is especially useful when dealing with categorical variables that have meaningful and numerous names, like job titles, where similarity embeddings tend to outperform search embeddings for such applications.

4.2. Analysis of Embeddings

For our analysis, we have utilized a collection of 4,000 artificially generated lambda terms. These terms were produced using a method based on random recursion, ensuring a balanced distribution of

Variables, Applications, and Abstractions throughout the structure of each term to include a wide variety of term types. The specific method of generating these terms is described in [14]. These lambda terms are represented as text, making them suitable for direct input into our chosen embedding model.

Hence, we have employed the OpenAI Embeddings model to transform lambda terms, including variable data, into vectors that should represent their unique meaning. This model interprets the textual representations of lambda terms, translating them into vectors.

The outcome, referred to as embeddings, captures the essence of the input text [15]. Then the clustering algorithm processes these vectors. This approach is akin to the Word2Vec methodology [16], which involves manipulating word embeddings (vectors representing the significance of words in a multidimensional space) through operations like addition or subtraction, thus creating new embeddings that retain the semantic value of the original word embeddings involved in the operation [16, 17].

4.3. Clustering Algorithms

During our previous work and analysis of the distribution patterns in the embedding space within this particular research, we noticed that unsupervised learning techniques, especially clustering analysis, are quite effective in distinguishing among the data. We have identified potential groupings with a natural, logical structure by illustrating various strategy priorities on a graphical plot, indicating a viable path for automated segmentation. It is essential to underline that our research aimed not to classify strategies per se but to investigate how data is distributed and uncover significant trends using unsupervised methods. Common methods for carrying out clustering analysis include K-means, DBSCAN (Density-Based Spatial Clustering of Applications with Noise), the Gaussian Mixture Model (GMM), and Agglomerative Clustering. Our study specifically has delved into the DBSCAN technique.

DBSCAN clusters data points based on proximity, effectively distinguishing between densely populated and less dense regions. It defines clusters by identifying areas of high density separated by regions of low density [18].

The DBSCAN algorithm uses common metrics such as Euclidean, cosine, L1, and L2. This algorithm exhibits resilience against outliers and offers the flexibility of not needing a predetermined number of clusters to operate effectively. It excels in identifying clusters of diverse shapes and sizes, showcasing its versatility in cluster formation. However, the performance of DBSCAN is significantly influenced by the selection of hyperparameters, and it may face challenges in accurately clustering data with variable densities.

Taking into account our previous studies [13], in this research we have focused on one clustering method: DBSCAN, with the Euclidean metric as one of the most promising. Since the chosen clustering method depends on fine-tuning hyperparameters, selecting the proper ones is crucial. To determine the most suitable hyperparameter set, we have evaluated the following metrics to assess the quality of clustering:

- The Silhouette score [19] evaluates how well an object fits within its cluster compared to others. It ranges from -1 to +1, where a higher score signifies better matching within the cluster and poorer matching with adjacent clusters. Scores above 0.7 denote robust clustering, above 0.5 indicate reasonable clustering, and above 0.25 suggest weak clustering. However, silhouette scores may converge in high-dimensional clustering, making differentiation challenging. The Silhouette score excels in assessing cluster quality for convex-shaped clusters but may falter with irregularly shaped or variably sized data clusters. This score is adaptable to any metric.

- The Davies-Bouldin Index (DBI) [20] measures the average similarity of each cluster with its closest cluster, based on the ratio of distances within the cluster to distances between clusters. Clusters that are more separated and less scattered receive better scores. The best score is zero, and lower scores signify superior clustering quality.

- The Calinski-Harabasz Index (CH Index) [21] is an internal metric that judges clustering quality based solely on the dataset and clustering outcomes without reference to external truth labels. It calculates the ratio of between-cluster dispersion to within-cluster dispersion, offering insight into effectiveness of clustering.

- The Within-Cluster Sum of Squares (WCSS) [22] evaluates the cohesiveness of clusters, especially in K-Means clustering. It totals the squared distances between each point in a cluster and its center, providing a numeric value for the cluster compactness.

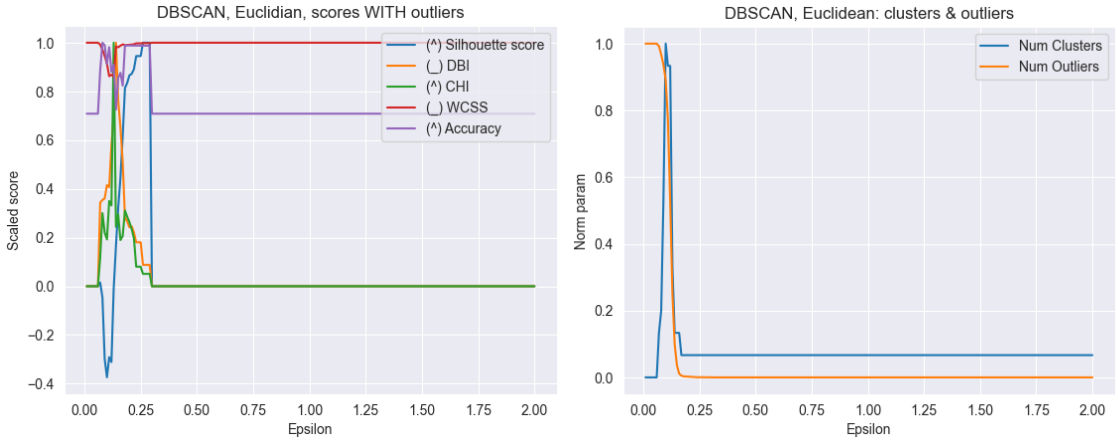


Fig. 1. Tuning the DBSCAN epsilon hyperparameter with various metrics (Silhouette, DPI, CHI, WCSS) for the Euclidean metric and estimating the number of outliers and clusters.

Collected mean embedding vectors can be visualized with Principal Component Analysis (PCA) [23] and t-distributed Stochastic Neighbor Embedding (t-SNE) [24]. The results of such data compression of the mean embeddings are shown in Fig. 2 for PCA and Fig. 3 for t-SNE.

5. Findings from Analyzing Lambda Terms

5.1. Collecting Embeddings

The mean embedding vectors, we have collected, can be graphically represented by using Principal Component Analysis (PCA) [23] and t-distributed Stochastic Neighbor Embedding (t-SNE). The visual compression of these mean embeddings through these techniques is presented in Fig. 2 for PCA and Fig. 3 for t-SNE. As can be seen, some clusters can be visually distinguished, particularly in the t-SNE plot. On the other hand, PCA does not show us such results; most clusters are visually interconnected and cannot be easily separated.

5.2. Analysis of Clustering

In our evaluation by using the chosen clustering techniques and quality assessment metrics, we initially sought to identify an optimal epsilon value for DBSCAN applied to the dataset of 4k embeddings (illustrated in Fig. 1). The selection of the epsilon value has proved difficult for this dataset, given the conflicting results from the clustering quality metrics and our aim to reduce the number of outliers. Notably, the Elbow method did not apply to determining WCSS for clustering. We aimed to achieve the highest possible Silhouette score and CH Index while aiming to lower the Davies-Bouldin Index (DBI). Consequently, for DBSCAN, we have opted for an epsilon value of 0.125.

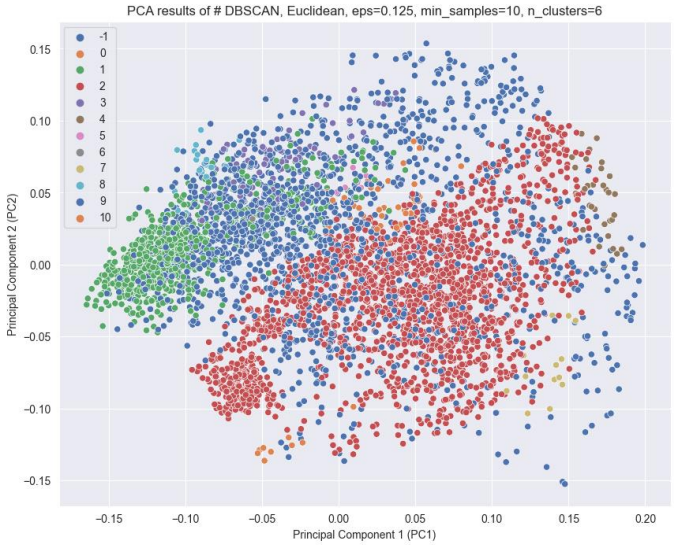


Fig. 2. Visualizing clustering results by using PCA dimension shrinking of embedding data with DBSCAN clustering.

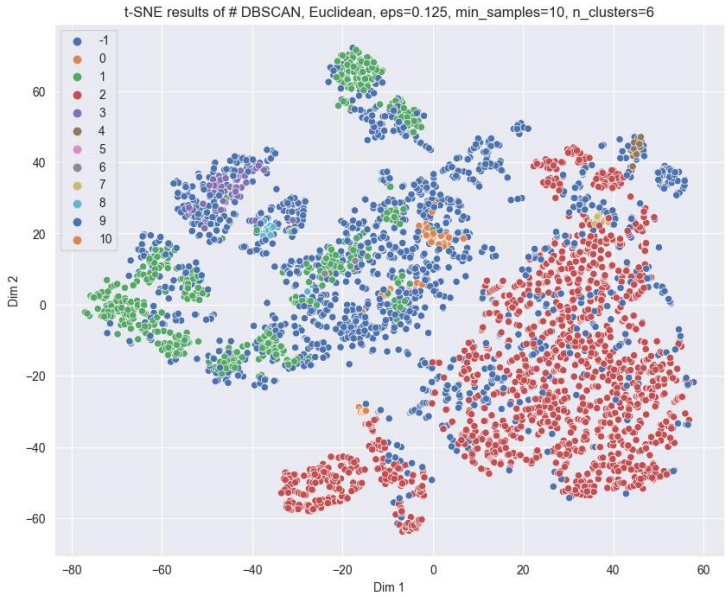


Fig. 3. Visualizing clustering results by using t-SNE *dimension shrinking of embedding data with DBSCAN clustering.*

6. Discussion of the results

In the discussion on the outcomes of our research, identifying the optimal strategy for term reduction emerges as a complex challenge, fundamentally unsolvable through mathematical means, as indicated by [2, 3, 14]. This complexity underscores the absence of a singular solution for optimal strategy selection across all conceivable terms or a universal reduction strategy. Nevertheless, developing viable methods remains feasible within certain constraints. Our approach, which involves generating lambda terms as a cost-effective means of data collection, may not fully encapsulate the breadth or critical characteristics of real-world terms. An additional challenge is the reliance on the OpenAI Embeddings model, originally trained on human text and code but not specifically on lambda terms or similar representations. That could potentially lead to inaccurate representations of lambda calculus terms in embedding matrices. This misalignment also complicates the translation of these embeddings into a universal format for subsequent analysis.

7. Summary of Findings

This research has transformed Lambda terms into embedding vectors with 1536 dimensions by applying the OpenAI Embeddings model, as explained in section 5.1. Analysis by using PCA and t-SNE to visualize these compressed mean vectors has revealed clear distinctions among Lambda term representations in these embeddings, confirming our initial hypothesis that patterns could be recognized through cluster analysis.

Then we have examined data cluster formation by using the DBSCAN technique with Euclidean measurements as could be seen in section 5.2. This exploration highlighted the capacity of the OpenAI Embeddings model to draw out significant attributes from Lambda terms. However, the broad training of OpenAI Embeddings has not be done on lambda term representations explicitly, but mostly for human-readable text and code, adding complexity to depicting Lambda calculus terms within embedding matrices precisely.

REFERENCES

1. Chitil, O. (2020). Functional Programming. *Clean C++20.* doi: https://doi.org/10.1007/978-1-4471-3166-3.
2. Deineha, O., Donets, V., & Zholtkevych, G. (2023). On Randomization of Reduction Strategies for Typeless Lambda Calculus. *Communications in Computer and Information Science* 1980, pp. 25–38.

3. Deineha, O., Donets, V., & Zholtkevych, G. (2023). Estimating Lambda-Term Reduction Complexity with Regression Methods. *International Conference "Information Technology and Interactions"*.
4. Ashouri, A.H., Bignoli, A., Palermo, G., Silvano, C., Kulkarni, S., & Cavazos, J. (2017). “MiCOMP: Mitigating the Compiler Phase-Ordering Problem Using Optimization Sub-Sequences and Machine Learning”. *ACM Trans. Archit. Code Optim., 14, 29:1-29:28.* doi: https://doi.org/10.1145/3124452.
5. Chen, J., Xu, N., Chen, P., & Zhang, H. (2021). Efficient Compiler Autotuning via Bayesian Optimization. *2021 IEEE/ACM 43rd International Conference on Software Engineering (ICSE)*, 1198-1209. doi: https://doi.org/10.1109/ICSE43902.2021.00110.
6. Cummins, C., Wasti, B., Guo, J., Cui, B., Ansel, J., Gomez, S., Jain, S., Liu, J., Teytaud, O., Steiner, B., Tian, Y., & Leather, H. (2021). CompilerGym: Robust, Performant Compiler Optimization Environments for AI Research. *2022 IEEE/ACM International Symposium on Code Generation and Optimization (CGO)*, 92-105. doi: https://doi.org/10.1109/CGO53902.2022.9741258.
7. Martins, L.G., Nobre, R., Cardoso, J.M., Delbem, A.C., & Marques, E. (2016). Clustering-Based Selection for the Exploration of Compiler Optimization Sequences. *ACM Transactions on Architecture and Code Optimization (TACO)*, 13, 1 - 28. doi: https://doi.org/10.1145/2883614.
8. Ashouri, A.H., Bignoli, A., Palermo, G., Silvano, C., Kulkarni, S., & Cavazos, J. (2017). MiCOMP: Mitigating the Compiler Phase-Ordering Problem Using Optimization Sub-Sequences and Machine Learning. *ACM Trans. Archit. Code Optim.*, 14, 29:1-29:28. doi: https://doi.org/10.1145/3124452.
9. Xavier, T.C., & Silva, A.F. (2018). Exploration of Compiler Optimization Sequences Using a Hybrid Approach. *Comput. Informatics*, 37, 165-185. doi: https://doi.org/10.4149/cai_2018_1_165.
10. Mammadli, R., Jannesari, A., & Wolf, F.A. (2020). Static Neural Compiler Optimization via Deep Reinforcement Learning. 2020 IEEE/ACM 6th Workshop on the LLVM Compiler Infrastructure in HPC (LLVM-HPC) and Workshop on Hierarchical Parallelism for Exascale Computing (HiPar), 1-11. doi: https://doi.org/10.1109/LLVMHPCHiPar51896.2020.00006.
11. Runciman, C., & Wakeling, D. (1992). Heap Profiling of a Lazy Functional Compiler. *Functional Programming*. doi: https://doi.org/10.1007/978-1-4471-3215-8_18.
12. Chlipala, A. (2015). An optimizing compiler for a purely functional web-application language. *Proceedings of the 20th ACM SIGPLAN International Conference on Functional Programming*. doi: https://doi.org/10.1145/2784731.2784741.
13. Deineha, O., Donets, V., & Zholtkevych, G. (2023). Unsupervised Data Extraction from Transformer Representation. *EEJET, vol. 3, In press*.
14. Deineha, O., Donets, V., & Zholtkevych, G. (2023). Deep Learning Models for Estimating Number of Lambda-Term Reduction Steps. *3rd International Workshop of IT-professionals on Artificial Intelligence “ProfIT AI 2023”.*
15. Feng, Z., Guo, D., Tang, D., Duan, N., Feng, X., Gong, M., Shou, L., Qin, B., Liu, T., Jiang, D., & Zhou, M. (2020). CodeBERT: *A Pre-Trained Model for Programming and Natural Languages*. doi: https://doi.org/10.18653/v1/2020.findings-emnlp.139.
16. Dwivedi, V.P., & Shrivastava, M. (2017). Beyond Word2Vec: *Embedding Words and Phrases in Same Vector Space*. ICON.
17. Hartigan, J.A., & Wong, M.A. (1979). *A k-means clustering algorithm*. doi: https://doi.org/10.2307/2346830.
18. Zhang, Y., Li, M., Wang, S., Dai, S., Luo, L., Zhu, E., Xu, H., Zhu, X., Yao, C., & Zhou, H. (2021). Gaussian Mixture Model Clustering with Incomplete Data. ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM), 17, 1 - 14. doi: https://doi.org/10.1145/3408318.
19. Ros, F., Riad, R., & Guillaume, S. (2023). PDBI: A partitioning Davies-Bouldin index for clustering evaluation. Neurocomputing, 528, 178-199. doi: https://doi.org/10.1016/j.neucom.2023.01.043.
20. Lima, S.P., & Cruz, M.D. (2020). A genetic algorithm using Calinski-Harabasz index for automatic clustering problem. *Revista Brasileira de Computação Aplicada*. doi: https://doi.org/10.5335/rbca.v12i3.11117.

21. Li, X., Liang, W., Zhang, X., Qing, S., & Chang, P. (2020). *A cluster validity evaluation method for dynamically determining the near-optimal number of clusters*. Soft Computing, 24, 9227-9241. doi: https://doi.org/10.1007/s00500-019-04449-7.
22. Chung, Heewon, Hoon Ko, Wu Seong Kang, Kyung Won Kim, Hooseok Lee, Chul Park, Hyun-Ok Song, Tae-Young Choi, Jae Ho Seo and Jinseok Lee. "Prediction and Feature Importance Analysis for Severity of COVID-19 in South Korea Using Artificial Intelligence: Model Development and Validation." *Journal of Medical Internet Research 23 (2021): n. Pag*. doi: https://doi.org/10.2196/27060.
23. Oliveira, F.H., Machado, A.R., & Andrade, A.D. (2018). On the Use of t-Distributed Stochastic Neighbor Embedding for Data Visualization and Classification of Individuals with Parkinson's Disease. *Computational and Mathematical Methods in Medicine*, 2018. doi: https://doi.org/10.1155/2018/8019232.

Дейнега Олександр Андрійович
Аспірант
Харківський Національний Університет імені В.Н. Каразіна, майдан Свободи 4, 61022, Харків, Україна;
e-mail: oleksandr.deineha@karazin.ua
https://orcid.org/0000-0001-8024-8812

Кластеризація Лямбда Термів з використанням Вбудувань

Актуальність. Важливість оптимізації компіляторів та інтерпретаторів для функціональних мов програмування, зокрема через призму лямбда-числення, має першочергове значення для вирішення зростаючих вимог до складності та продуктивності в розробці програмного забезпечення. Це дослідження розміщується в цій критичній області, спрямоване на використання передових методів машинного навчання для покращення ідентифікації та застосування стратегії скорочення коду.

Мета. Основною метою є підвищення продуктивності та ефективності компіляторів та інтерпретаторів шляхом поглиблення розуміння стратегій скорочення програмного коду в лямбда-численні. Дослідження спрямоване на використання машинного навчання для перетворення лямбда-термінів у вектори ознак, полегшуючи дослідження оптимальних стратегій зменшення.

Методи дослідження. Дослідження використовує комплексний підхід, створюючи широкий спектр лямбда-термінів для аналізу. Він використовує модель вбудовування тексту OpenAI для перетворення цих термінів у вектори вбудовування, використовуючи кластеризаційний аналіз (DBSCAN з евклідовими вимірюваннями) та візуалізацію (PCA та t-SNE) для виявлення шаблонів і оцінки відокремлюваності функцій. Дослідження орієнтується через складність вибору між конкретними та універсальними стратегіями скорочення.

Результати. Отримані дані виявляють чіткі відмінності між представленнями лямбда-термів у векторах вбудовування, підтверджуючи гіпотезу про те, що кластерний аналіз може виявити шаблони, які можна ідентифікувати. Однак виникли труднощі через загальну спрямованість навчання OpenAI Embeddings на текст і код, які читаються людиною, що ускладнює точне представлення термінів лямбда-числення.

Висновки. Це дослідження підкреслює труднощі у визначенні оптимальної стратегії скорочення термінів лямбда-числення, підкреслюючи обмеження поточних математичних моделей і потребу в адаптованих програмах машинного навчання. Незважаючи на перешкоди з адаптивністю моделі OpenAI Embeddings, дослідження з'ясовує суттєве розуміння потенціалу машинного навчання для вдосконалення процесів оптимізації компіляторів та інтерпретаторів у середовищах функціонального програмування.

Ключові слова: *Лямбда-терми, Кластерний Аналіз, Претреновані Вбудування, Прихований Простір.*

Надійшла у першій редакції 11.07.2023, в останній - 03.09.2023.

DOI: https://doi.org/10.26565/2304-6201-2023-59-03

УДК (UDC) 519.6, 51-76

Zolotukhin Volodymyr *student*
V.N. Karazin Kharkiv National University, 4 Svobody Square, Kharkiv, Ukraine, 61022
e-mail: zolotukhinvolodymyr@gmail.com
https://orcid.org/0009-0004-0303-0624

Yanovsky Volodymyr *Doctor of Physical and Mathematical Sciences, professor*
V. N. Karazin Kharkiv National University, 4 Svobody Square, Kharkiv, Ukraine, 61022;
Institute of Single Crystals, National Academy of Sciences of Ukraine, 60 Nauky Ave., Kharkiv, Ukraine, 61001.
e-mail: yanovsky@isc.kharkov.ua
https://orcid.org/0000-0003-0461-749X

Impact of violation of democratic strategies with memory on population evolution

Relevance. The lack of trust in modern society often hinders the development of humanity and sometimes calls into question the future of the human population as a whole. Throughout the history of societal development, there has been an observed phenomenon where a particular idea captures the minds of people, leading them to adopt similar (or very similar) behavioral strategies. To improve understanding of internal processes in a society where the uniform distribution of strategies among the population is disrupted, detailed research is necessary, which is impossible without appropriate software. **Objective.** The aim of the study is to investigate the influence of the number of agents of a particular strategy on the outcome of population evolution as a whole. The study explores the nature of changes in evolution under the conditions of gradual, monotonous increase in agents of a specific strategy from 1 agent to 10% of the democratic population. The research also aims to identify strategies that are evolutionarily viable only under the condition of increasing their carriers in the population. **Research Methods.** The evolution of the population with a full set of behavioral strategies, limited only by a memory depth of 2, was considered with an increased number of agents of a specific strategy. Each agent interacts with every other, including itself, according to the iterative model of the prisoner's dilemma. Rewards are determined by payoff matrices. Each subsequent generation of the population sequentially loses agents of the most disadvantageous behavioral strategy from the previous generation. Agents that bear the chosen strategy interact with each other and with another population according to standard laws. Several strategies were considered, the number of agents of which was increased. Among them were strategies with complexity lower than the average complexity of the population and higher than the average complexity of the population. A variant was also considered where the number of agents of the strategy that won in a democratic society increased. **Results.** The study demonstrates how the presence of a highlighted strategy with an increased number of carriers affects the dynamics of the population. An increase in the final average earnings of the population was observed. It was found that increasing the number of agents does not lead to the victory of a strategy that did not win in the democratic population. **Conclusions.** The results of the study identify the main consequences of the influence of the number of agents of a particular strategy on population evolution.

***Keywords**: aggressiveness, evolution, population, strategy, complexity, society*

How to quote: Zolotukhin V., and Yanovsky V., "Impact of violation of democratic strategies with memory on population evolution", *Bulletin of V.N. Karazin Kharkiv National University, series Mathematical modelling. Information technology. Automated control systems*, vol. 59, pp.24-34, 2023. https://doi.org/10.26565/2304-6201-2023-59-03

Як цитувати: Zolotukhin V., and Yanovsky V. Impact of violation of democratic strategies with memory on population evolution. *Вісник Харківського національного університету імені В.Н.Каразіна серія. Математичне моделювання. Інформаційні технології. Автоматизовані системи управління*. 2023. вип. 59. С.24-34. https://doi.org/10.26565/2304-6201-2023-59-03

1 Introduction.

The interaction of large collectives of agents has sparked sustained interest for several decades [1-2]. Interest in such a phenomena arises from several different fields. It initially emerged in biology where the central question requiring scientific explanation was the diversity of observed animals. Charles Darwin, leveraging observational data, proposed an explanation that resulted in the creation of the remarkable theory of evolution [3]. This provided a significant impetus for applying similar reasoning to

an extraordinarily wide range of questions in other domains. Starting from the evolution of languages [4,5] to the evolution of computer programs and artificial intelligence. In cybernetics, this interest transformed into the management of complex multi-element systems [6], the potential creation of artificial life, and artificial intelligence [7]. Recently, research related to swarm intelligence has gained relevance [8]. Sociology, also being one of these sciences, requires an understanding of interactions among individuals in society and the emergence of macro-behaviors in societies [9,10]. Another intriguing direction is associated with the emergence of altruistic behavior in multi-agent systems [11]. In building population models and models of interaction among individual members of a population, elements of game theory are commonly used [12]. The reward rule is chosen in such a way that mutual cooperation always requires more resources than responding aggressively to cooperation [13]. In other words, local cooperation is always disadvantageous. At each step of evolution, the population discards the agent with the fewest points. By evolution of the population, we mean an algorithm based on three principles: inheritance, variability, and selection.

In this work, we explore the impact of deviations from the equality of strategies on the evolution of the population by increasing the quantities of a particular strategy. The main question of interest is how such deviations from "democracy" affect the course of evolution and in what manner. It should be noted that, unlike previous works [16-18], agents of strategies will not unite into a cluster, meaning that agents interact directly and not strategies. In each case, the initial population includes agents of all strategies constrained by a memory depth of two. The increase in the quantity of specific strategies in the population starts from 1 up to a maximum of 10% of the total number of all strategies in a "democratic" population. Subsequently, for each population change, its evolution was modeled to obtain collective characteristics of the population that emerged as a result of evolution. This allowed identifying changes in such characteristics with the quantity of added agents of a specific strategy. Through the conduct of multiple series of experiments, it has been shown that an increase in the number of agents of a particular strategy leads to some, sometimes quite significant, changes in the population. The effect of variability in the average complexity of the final population has been observed with a monotonous increase in the quantity of added strategies.

2 Main requirements.

Strategy is an unchanging law, which defines what move a strategy bearer (an agent) must do in certain conditions. Move of the strategy can be zero (0) or one (1), corresponding to aggression or cooperation. Memory of the strategy is an ability of a certain strategy to use information about previous moves of the opponent strategy, which were made against it during the game, to perform a certain move: zero or one. Thus, strategies which use information only about the current move and don’t use information about previous [moves] are called “Strategies without memory”. Strategies that use information about previous moves are called “Strategies with memory”. Memory depth is a value that equals the number of previous moves of the opponent strategy that impact the course of the strategy. In other words, strategies without memory have memory depth that equals 0, and strategies with memory which use information only about the last move, have memory depth that equals 1, and further the same way by analogy.

Strategy without memory has two answers for the opponent’s move: zero or one. The opponent, in turn, as it was mentioned before, also has two possible ways to move: zero or one – aggresion or cooperation. Such a strategy may be described with a binary sequence, where the number of a bit is an opponent’s move, and its value is an answer for that move. This way of displaying differs from the way that was provided in articles written by Kuklin, Pryimak, Yanovsky [16-18]. In those articles, the name of strategy, its representation in the form of a binary sequence, was written from left to right, like words in most European languages. In this paper and in further papers, the way of displaying is reversed, in other words, it is written as a binary number where the least significant bit is to the right. This is done for the convenience of representing the strategy in the program and for the correct implementation of bitwise operations.

So, there are four strategies with a memory depth that equals 0: 00, 01, 10, 11. All of them respond to the opponent's move according to the rules that were described above. But at the first move, there is a situation in which the strategy has to make a move under the condition of uncertainty, in other words, without having information about the opponent's previous moves. This situation very often arises during the interaction of individuals in the society, so it must be simulated realistically. In the above-mentioned articles, the concept of the first step is introduced, a move, which is made in the case when there is a lack

of information to make a decision by the main algorithm. Such a move is written before the main name of the strategy in square quotes "[]". For the strategies without memory, the first move is described with one bit. As a result, the final number of different strategies in the population is doubled: [0]00, [1]00, [0]01, [1]01, [0]10, [1]10, [0]11, [1]11 .

Strategies with depth that equals 1 must do their move taking into account both the opponent's current move and the previous one, in other words, a number with two bits is used to define the bit's number. That means that the binary representation of a strategy must be twice longer than with the memory depth that equals 0. Further considerations lead us to the fact that the length of the binary representation doubles with every increase of the memory depth index. For all strategies with memory, there is a valid statement that the first move is some strategy with memory depth decreased by one. For instance, for the strategy 1001, there are 8 variants of the first move which fully correspond to the 0 memory depth strategies: [0][00]1001, [1][00]1001, etc. For convenience, below, the strategy means a binary sequence without taking into account moves under conditions of uncertainty. The strategy taking these conditions into account will be called "sub-strategy". It's important to focus on the fact that some strategies with memory depth 1 correspond to strategies with memory depth that equals 0. It's obvious that these are strategies which even having the information about the opponent's previous move make a move without considering that information. For example: 0000 ~ 00, 0101 ~ 01, 1010 ~ 10, 1111 ~ 11. From here we can see, at the memory depth 1, all strategies with memory depth 0 are represented. Thus, we can conclude that all strategies from previous depths including zero are represented at every memory depth.

From all the aforementioned, it is possible to extract formulas of the dependency of the number of strategies from memory depth. Based on the fact that to describe the strategy with memory depth that equals *k*, 2^{k+1} bits a needed from which $2^{2^{k+1}}$ sequences can be built and that each strategy has a number of sub-strategies that equals the number of strategies from the previous memory depth (2^{2^k}), we can conclude that the formula of the number of the strategies that take into account unique first moves looks like:$2^{(2^{k+2}-1)}$.

3 Distribution of rewards.

In the paper, the same model of distribution as in articles by Kuklin, Pryimak, Yanovsky [16-18] is used, where, at the same time, Robert Axelrod's model for prisoner's dilemma[13] is used. Every agent (a bearer of the strategy) may choose aggression or cooperation, points are counted by the payoff matrix that was also suggested by Axelrod (Table 1):

Table 1 – payoff matrix

	0 (aggression)	**1 (cooperation)**
0 (aggression)	1	5
1 (cooperation)	0	3

This table implies that by answering aggression with aggression, every agent receives a point. If responds to aggression with cooperation, an aggressive agent receives 5 points, and the other – zero. This rule works the same way in reverse, like a response to cooperation with aggression.Each agent receives 3 points if cooperation is responded with cooperation. Thus, maximum total profit is reached by mutual cooperation, but maximum personal [profit] is reached by betrayal, in other words by responding to cooperation with aggression.

It is important to mention that the represented matrix (Table 1) is not the only right. There may be different other payoff matrices, but all of them should be guided by the rule:

$$t > r > p > s \tag{3.1}$$

where t – the amount of points received by the agent who responds to cooperation with aggression.

r – the amount of points received by the agent who responds to cooperation with cooperation.

p – the amount of points received by the agent who responds to aggression with aggression.

s – the amount of points received by the agent who responds to aggression with cooperation.

4 Population's characteristics.

There are typical characteristics for every population: aggression, complexity, average amount of points received for a move. Version of populations with uneven distribution of agents among strategies involves one more characteristic: the amount of agents of a certain strategy in the population. For our case, which is the series of experiments with modeling evolution of strategies agents' population with a gradual increase in the number of a certain strategy's agents at the first stage of its evolution, it's advisable to use final characteristics of society.

The complexity of the strategy is defined according to the principle of describing the complexity of finite 0 and 1 sequences, assuming that a polynomial of a greater degree is more complex than polynomial of a lesser grade. Such a sequence may be considered as a function, then complexity of this function is perceived as a display like $A: M \rightarrow M$, if:

$$y = Ax \tag{4.1}$$

where $= y_1, y_2 \dots y_n$, sequence, which elements are defined as:

$$y_i = x_{i+1} - x_i \tag{4.2}$$

where $i = 1,2,\dots,n$ is an element of the sequence.

The amount of received points, or income, is defined as the average value of received points per every agent's move.

Aggression is the average value of the amount of all agents' aggressive moves. The connection between the income and aggression is examined[16-18] and is expressed by ratio:

$$A(t) = \lambda * (P_{max} - P(t)) - \alpha \tag{4.3}$$

where $\lambda = 5.3/8;\ \alpha = 0.2$ – selected empirically coefficients,

P – income of the strategy.

This particular ratio is used for calculations of aggression in this paper. The direct calculation of the number of zeros that were made is associated with an excessive increase in the requirements for calculating power, so a more computationally simple way was chosen.

5 Terms for running experiments.

The purpose of the modeling is to determine the impact of increasing the quantity of a particular strategy on the evolution of the strategy population with a memory depth of 2. In a certain sense, the strategies lose equality in this process. Then, by increasing the initial quantity of a specific strategy in the population, the evolution of a new population is simulated. Upon completion of evolution, collective characteristics of the population that emerged as a result of evolution are obtained from the simulation data. The increase in the initial quantity of the designated strategies ranges from 1 to 3276. The maximum number of added strategies is equal to 10% of the total number of agents in the population in the classical scenario with a memory depth of 2 [3-4]. Thus, this quantity can be considered small in comparison to the size of the "democratic" population with an equal number of all strategies. Therefore, in each series of experiments, the increase in the initial quantity occurs discreetly with a step of 3276/10. Then, in each series, a complete evolution of the population with an increased number of strategies is performed. The obtained data allows determining how collective characteristics of the final population change depending on the quantity of initially added strategies. These dependencies help identify characteristic changes that arise with a change in the quantity of a particular strategy.

6 Case with an increased number of agents of strategy 1011.

Strategy 1011 belongs to a memory depth 1 strategy, with a complexity of 4. That is significantly less than the average complexity of the population, which is 8 with a memory depth of 2. At first glance, it seems that such a strategy should not significantly influence the complexity of the strategies that emerge as a result of evolution. However, the obtained data indicates quite irregular fluctuations in complexity

when changing the quantity of this strategy (See Fig. 6.1). The amplitude of these fluctuations is quite significant, ranging from a maximum of 8 to values as low as 2, even lower than the complexity of strategy 1011 that is being added. The structure of the minima is shown on Fig. 6.1 to the right when changing the quantity with a smaller scale in the minima region. It is easy to notice that the number of such fluctuations increases with the reduction in the scale of changes in the initial quantity. Thus, the complexity of the strategies that remain after evolution is a variable function of the initially added strategy quantity.

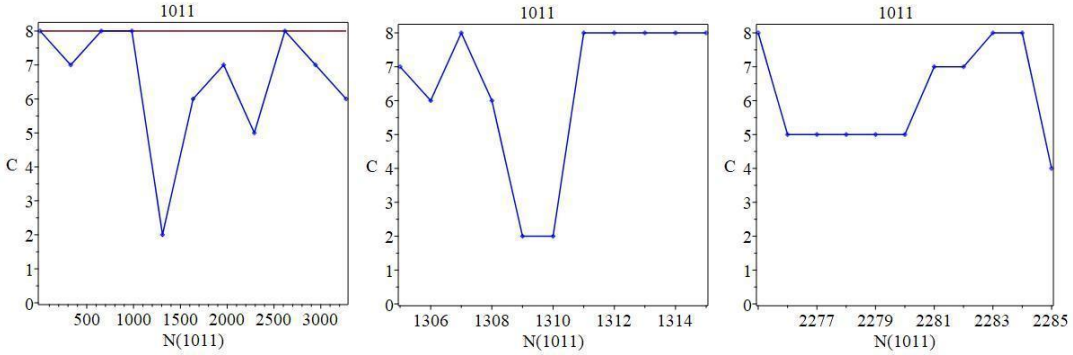


Fig. 6.1 On the left, the dependence of the average complexity of strategies that "survived" on the initial quantity of strategy 1011 is shown. The brown color represents the complexity of strategies that survived in the "democratic" population. On the right, the structure of minimal outliers is shown with a smaller scale of changes in the initial quantity of strategy 1011. The intervals of changes in the figures on the right correspond to [1305-1315] and [2285-2295], respectively.

A more detailed investigation of complexity with a smaller scale of changes requires significant time resources. However, it can be stated that it will be a variable function even on small scales. It is interesting to note that the relative part of the interval of changes in the quantity of 1011 on which the complexity reaches the complexity of the democratic population is significantly less than one. If we estimate this ratio from Fig. 6.1 (left), it equals 1/10. In other words, adding the 1011 strategy typically causes a decrease in the complexity of the population that remains after evolution. However, with a monotonic change in quantity, unforeseen intervals of changes occur where the complexity of the populations reaches the maximum complexity of 8. Interestingly, the average memory depth of the surviving strategies does not show such variability. It remains constant throughout the entire interval of changes in the quantity of the 1011 strategy and equals 2.

The next collective variable of interest is the average number of points a strategy receives per move. In a certain sense, it characterizes the efficiency of strategies in the population. The dependence on the quantity of added 1011 strategies is illustrated in Fig. 6.2. The modeling results indicate that the changes in payoff per move do not vary significantly (see Fig. 6.2).

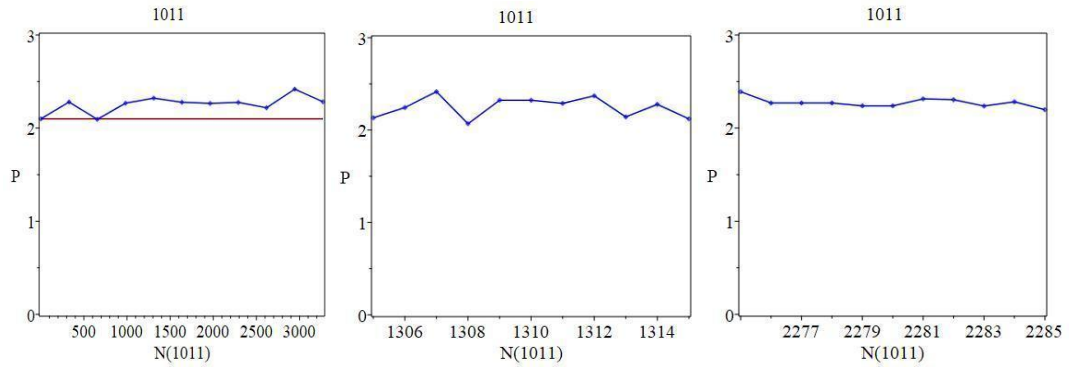


Fig. 6.2 On the left, dependence of the average payoffs per move for strategies surviving the evolution on the initial quantity N(1011) of 1011 strategies. Brown color represents the payoff level in the "democratic" population. On the right, behavior of average payoffs per move within the intervals specified in Fig. 6.1.

It is noticeable that the variability of the dependencies in Fig. 6.2 is significantly smaller than the average complexity. Thus, fluctuations in complexity do not affect the payoffs per move. This is evident from minor changes in the figures on the right, constructed within the intervals of sharp changes in

average complexity. Typically, in most cases, an increase in the quantity of strategies emerging in the population through evolution results in higher payoffs per move than in the "democratic" population.

The quantity of points obtained is closely related to a characteristic known as the aggressiveness of strategies. In previous works [X, Y], relationships that align well with simulation data were obtained. After verifying them in the case considered in this work and to expedite computation time, this relationship was utilized for calculating aggressiveness. Therefore, aggressiveness is computed on the basis of the simulation data of payoffs per move. Figure 6.3 illustrates the dependencies of aggressiveness on the quantity of initially added strategies 1011.

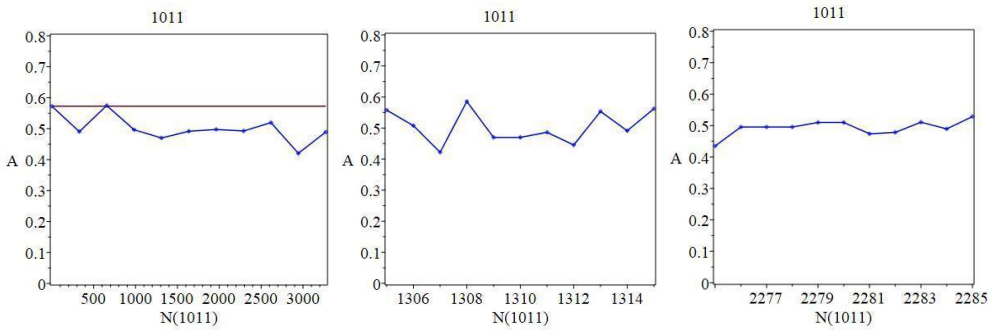


Fig. 6.3 On the left, the dependence of the average aggressiveness of strategies that "survived" on the initial quantity of strategies 1011. The level of aggressiveness in the case of the "democratic" population corresponds to the brown line. On the right, aggressiveness within the intervals [1305-1315] and [2285-2295], respectively.

Thus, aggressiveness, for the majority of initial quantities of strategies 1011, decreases compared to the aggressiveness of the "democratic" population (see Fig. 6.3). An exception is observed when N(1011)=655, while the average aggressiveness of surviving strategies is higher than the aggressiveness of the corresponding strategies in the "democratic" population. Clearly, with this initial quantity of strategies 1011, the payoffs per move are lower than in the "democratic" case. Among the finalists of evolution, strategy 1011 is absent.

7. Case with an increased number of agents of strategy 01001011.

Now let's consider the impact of increasing the quantity of a more complex strategy on the evolution of the population. Strategy 01001011 belongs to a depth-2 memory strategy with a complexity of 8. This is greater than the initial average complexity of strategies in the "democratic" population with a depth-2 memory. Again, the main question is to determine the collective characteristics of the strategies that remain as a result of evolution. Figure 7.1 presents data on the average complexity of strategies formed through evolution based on the quantity of added strategies 01001011.

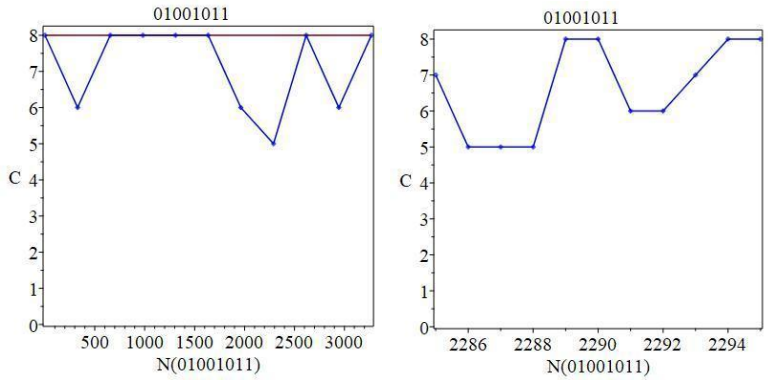


Fig. 7.1 On the right, the dependence of post-evolutionary average complexity on the initial quantity of added strategies 01001011. The complexity level in the "democratic" population corresponds to the brown line. On the left, a portion of the dependency is shown where a minimum is observed with a smaller scale of changes in the interval [2291; 2292].

It is evident that the addition of 2288 agents of the 01001011 strategy results in a sharp decrease in complexity to 5 (see Figure 7.1 on the right). The structure of this decline is shown on the left and exhibits a variable pattern. In a certain sense, similar to the previous case, the average complexity of surviving strategies has a variable structure with significant fluctuations in average complexity. The relative proportion of intervals of initial quantity of added strategy where the maximum complexity of 8 is achieved, compared to the case of division by 6, increases and reaches 3/10. Thus, the complexity of the added strategy affects this indicator.

Now let's move on to the changes in the average payoffs per move of the strategy. The results obtained during the evolution modeling are presented in Figure 7.2.

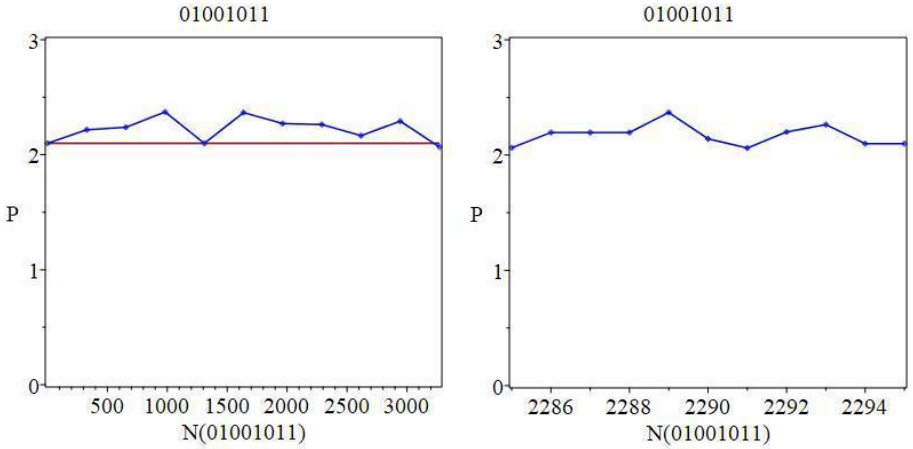


Fig. 7.2 on the right illustrates the dependence of payoffs per move on the initial quantity of 01001011 strategies. On the left, a part of the dependence is shown where a minimum is observed with a smaller scale of changes in the interval [2291; 2292].

In this case as well, at certain initial quantities of 01001011 strategies, it is observed that the obtained profits per move are lower than in the "democratic" scenario. However, it is typical to achieve higher payoffs per move with an increase in the quantity of the additional strategy, although these increases are quite modest. It is evident that the average aggressiveness of strategies after evolution will be lower than in the "democratic" case (see Figure 7.3).

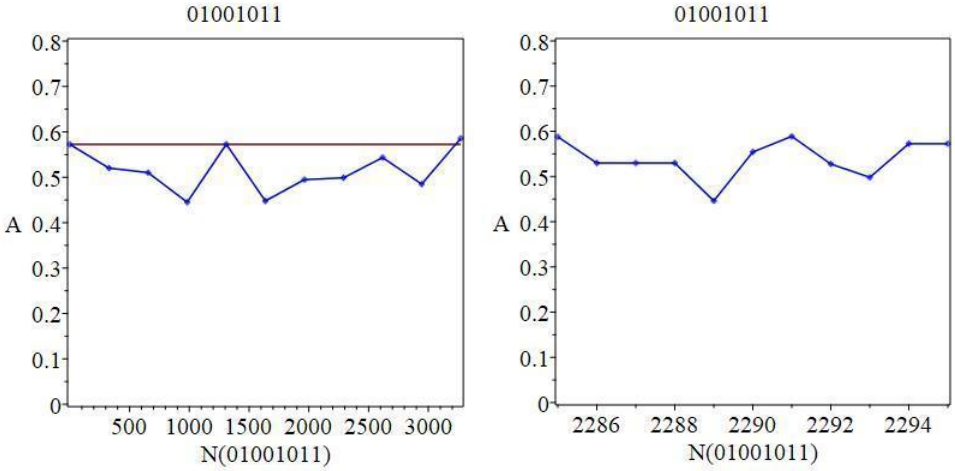


Fig. 7.3 on the right, dependence of aggressiveness on the initial quantity of 01001011 strategies. On the left, a part of the dependency with a smaller scale of changes in the interval [2291; 2292].

Therefore, when choosing a strategy of maximum complexity, a decrease in the aggressiveness of the strategies that remain after evolution should be expected. Thus, the influence of this strategy has a moderate nature due to its absence among the finalists of evolution. It altered the course of evolution at intermediate stages. Hence, the average aggressiveness of finalists is typically lower than the average aggressiveness of the "democratic" population.

8. Case with an increased number of agents of strategy 10001011

In this section, let us consider the winning strategy of evolution in the "democratic" population. Strategy 10001011 belongs to a memory depth 2 strategy with a complexity of 8. This strategy emerged victorious in the classic evolution scenario [3-4]. In this case, it is expected that neither the average memory depth nor the average complexity of surviving strategies change with different initial quantities of this strategy, and they remain at their maximum values. In this sense, they do not provide insight into the events occurring during evolution. Therefore, let's consider the more informative characteristic - the number of agents specifically with the 10001011 strategy that remain in the population after evolution.

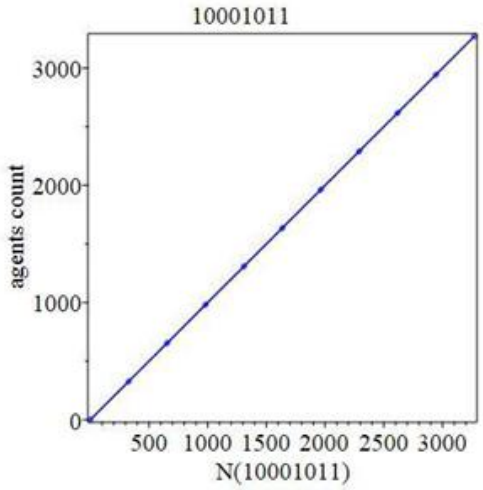


Fig. 8.1 The number of agents with the strategy 10001011 among the winners of evolution.

From the obtained data, it is easy to notice that all added strategies are retained during evolution. Therefore, there are no changes in either the complexity or the depth of memory in the population as a result of evolution. This creates a special case of "democracy" violation. The number of payoffs per move for the strategy, according to the modeling data, is shown in Fig. 8.2 and is determined by the dominance of this strategy during evolution.

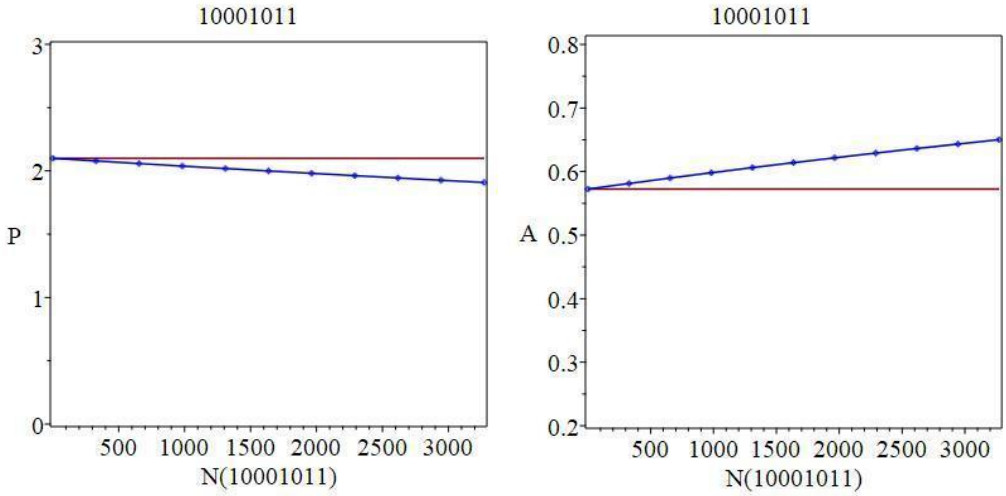


Fig. 8.2 On the left, the change in average payoffs per move for the surviving strategies with evolution. On the right, the average aggressiveness of these strategies.

Therefore, adding the winning strategy, with an increased quantity, reduces payoffs per move and increases the average aggressiveness of the population. This may indicate that with an increase in agents, there is a growing "competition" among the strategy agents.

9. Conclusion.

Thus, it can be noted that an increase in the number of agents of a certain strategy by 10% does not lead to the dominance of that strategy. In all series (except the last one), fluctuations in final complexity were observed, indicating a variable nature of changes with monotonous increases in the initial quantity of added strategies. An increase in the number of agents undoubtedly leads to changes in the population, but these changes do not result in the victory of the increased group. Therefore, it is typical at the final stage to observe a decrease in the aggressiveness of the strategies. It should be noted that with an increase in the quantity by more than 10% from the number of strategies in the democratic population, the dominance of some additional strategies and an increase in population aggressiveness compared to the level of aggressiveness in the "democratic" population should be expected. This can be observed from the behavior of the strategy 01001011 when its quantity is maximally increased (10%), resulting in the population's aggressiveness exceeding the level of aggressiveness in the "democratic" population. Additional research is required to determine the possibility of such a peculiar phase transition. It is worth noting that the evolution of the population requires significant computational resources; therefore, a relatively large scale of changes in the quantity of added strategies was chosen to reduce equipment requirements. This scale was 1/10 of the maximum initial quantity which, in turn, constituted 10% of the size of the "democratic" population. This prevented the identification of changes on smaller scales throughout the entire interval [1, 3276]. These additional complexities arise when attempting to increase the quantity of added strategies.

СПИСОК ЛІТЕРАТУРИ

1. Michael Wooldridge. “An Introduction to MultiAgent Systems”. *John Wiley & Sons Ltd*, 2002, https://www.researchgate.net/publication/200027549_An_Introduction_to_MultiAgent_Systems
2. Munindar P. Singh. “Multiagent Systems: A Theoretical Framework for Intentions”. *Know-How and Communications*, 1999, https://www.researchgate.net/publication/243777883_Multiagent_Systems_A_Theoretical_Framework_for_Intentions
3. Darwin, Charles. “On the Origin of Species by Means of Natural Selection, or the Preservation of Favoured Races in the Struggle for Life”. *Jhon Murray*, 1859, https://www.researchgate.net/publication/220045363_On_the_Origin_of_Species_by_Means_of_Natural_Selection
4. Ferdinand de Saussure, “Course in General Linguistics”, 1998, https://www.academia.edu/32877516/Course_in_General_Linguistics_Ferdinand_de_Saussure
5. Lieberman F. “The Biology and Evolution of Language”, *Harvard University Press*, 1984, https://www.researchgate.net/publication/299483779_The_Biology_and_Evolution_of_Language
6. George Luger. “Artificial Intelligence: Structures and Strategies for Complex Problem Solving 5th Edition”, *Addison-Wesley*, 2005, https://www.academia.edu/26150689/GEORGE_F_LUGER_Structures_and_Strategies_for_Complex_Problem_Solving_at_BULLET_at_BULLET
7. Luc Steels. “The Artificial Life Roots of Artificial Intelligence”, *Artif Life*, 1993, 1 (1_2): 75–110. doi: https://doi.org/10.1162/artl.1993.1.1_2.75
8. Bonabeau, Eric, Marco Dorigo, Guy Theraulaz, “Swarm Intelligence: From Natural to Artificial Systems”, *Oxford Academic*, https://doi.org/10.1093/oso/9780195131581.001.0001
9. Scott John. “Sociology: The Key Concepts”, *Routledge*, 2006, https://www.shortcutstv.com/wp-content/uploads/2020/01/Sociology_the_key_concept.pdf
10. Rogers, Kimberly B. Smith-Lovin, Lynn. Action, Interaction, and Groups // The Wiley-Blackwell companion to Sociology / G. Ritzer (ed.). — Oxford, etc.: Wiley-Blackwell, 2012. P. 121—138. ISBN 978-1-4443-4735-7.
11. Nowak M., Sigmund K. “A strategy of win-stay, lose-shift that outperforms tit-for-tat in the Prisoner’s Dilemma game.”, *Nature*, 1993, vol. 364, c. 56-58. https://www.nature.com/articles/364056a0
12. Weibull J. W. “Evolutionary game theory.”, *Cambridge*: *MIT press*, 1997. 265c. https://mitpress.mit.edu/9780262731218/evolutionary-game-theory/

13. Axelrod R. “The evolution of cooperation” *New York: Basic Books*, 1984.9c. https://ee.stanford.edu/~hellman/Breakthrough/book/pdfs/axelrod.pdf
14. Куклін В.М., Приймак О.В., Яновський В.В. “Influence of memory on population evolution” *Вісник Харківського національного університету імені В.Н. Каразіна, серія «Математичне моделювання. Інформаційні технології. Автоматизовані системи управління»*, Вип. 29, 2016, с. 41-66. https://periodicals.karazin.ua/mia/article/view/6557
15. Куклін В.М., Приймак О.В., Яновський В.В. “The memory and the evolution of populations” *Вісник Харківського національного університету імені В.Н. Каразіна, серія «Математичне моделювання. Інформаційні технології. Автоматизовані системи управління»*, Вип. 35, 2017, с. 38-60. https://periodicals.karazin.ua/mia/article/view/9841/9365
16. Куклін В.М., Приймак О.В., Яновський В.В. “The evolution of strategies communities in the presence of sources”, *Вісник Харківського національного університету імені В.Н. Каразіна, серія «Математичне моделювання. Інформаційні технології. Автоматизовані системи управління»*, Вип. 36, 2017, с. 68-84. https://periodicals.karazin.ua/mia/article/view/10098

REFERENCES

1. Michael Wooldridge. “An Introduction to MultiAgent Systems”. *John Wiley & Sons Ltd*, 2002, https://www.researchgate.net/publication/200027549_An_Introduction_to_MultiAgent_Systems
2. Munindar P. Singh. “Multiagent Systems: A Theoretical Framework for Intentions”. *Know-How and Communications*, 1999, https://www.researchgate.net/publication/243777883_Multiagent_Systems_A_Theoretical_Framework_for_Intentions
3. Darwin, Charles. “On the Origin of Species by Means of Natural Selection, or the Preservation of Favoured Races in the Struggle for Life”. *Jhon Murray*, 1859, https://www.researchgate.net/publication/220045363_On_the_Origin_of_Species_by_Means_of_Natural_Selection
4. Ferdinand de Saussure, “Course in General Linguistics”, 1998, https://www.academia.edu/32877516/Course_in_General_Linguistics_Ferdinand_de_Saussure
5. Lieberman F. “The Biology and Evolution of Language”, *Harvard University Press*, 1984, https://www.researchgate.net/publication/299483779_The_Biology_and_Evolution_of_Language
6. George Luger. “Artificial Intelligence: Structures and Strategies for Complex Problem Solving 5th Edition”, *Addison-Wesley*, 2005, https://www.academia.edu/26150689/GEORGE_F_LUGER_Structures_and_Strategies_for_Complex_Problem_Solving_at_BULLET_at_BULLET
7. Luc Steels. “The Artificial Life Roots of Artificial Intelligence”, *Artif Life*, 1993, 1 (1_2): 75–110. doi: https://doi.org/10.1162/artl.1993.1.1_2.75
8. Bonabeau, Eric, Marco Dorigo, Guy Theraulaz, “Swarm Intelligence: From Natural to Artificial Systems”, *Oxford Academic*, https://doi.org/10.1093/oso/9780195131581.001.0001
9. Scott John. “Sociology: The Key Concepts”, *Routledge*, 2006, https://www.shortcutstv.com/wp-content/uploads/2020/01/Sociology_the_key_concept.pdf
10. Rogers, Kimberly B. Smith-Lovin, Lynn. Action, Interaction, and Groups // The Wiley-Blackwell companion to Sociology / G. Ritzer (ed.). — Oxford, etc.: Wiley-Blackwell, 2012. — P. 121—138. — ISBN 978-1-4443-4735-7
11. Nowak M., Sigmund K. “A strategy of win-stay, lose-shift that outperforms tit-for-tat in the Prisoner’s Dilemma game.”, *Nature*, 1993, vol. 364, c.56-58. https://www.nature.com/articles/364056a0
12. Weibull J. W. “Evolutionary game theory.”, *Cambridge: MIT press*, 1997. 265c. https://mitpress.mit.edu/9780262731218/evolutionary-game-theory/
13. Axelrod R. “The evolution of cooperation” *New York: Basic Books*, 1984.9c. https://ee.stanford.edu/~hellman/Breakthrough/book/pdfs/axelrod.pdf
14. V. M. Kuklin, O. V. Pryimak, V. V. Yanovsky. “Influence of memory on population evolution” *Bulletin of V.N. Karazin Kharkiv National University, series «Mathematical modeling. Information technology. Automated control systems»*, vol. 29, 41-66, April 2016. [In Russian] https://periodicals.karazin.ua/mia/article/view/6557

15. V. M. Kuklin, O. V. Pryimak, V. V. Yanovsky. “The memory and the evolution of populations” *Bulletin of V.N. Karazin Kharkiv National University, series «Mathematical modeling. Information technology. Automated control systems»*, vol. 35, 38-60, November 2017. [In Russian] https://periodicals.karazin.ua/mia/article/view/9841/9365
16. V. M. Kuklin, O. V. Pryimak, V. V. Yanovsky. “The evolution of strategies communities in the presence of sources”, *Bulletin of V.N. Karazin Kharkiv National University, series «Mathematical modeling. Information technology. Automated control systems»*, vol. 36, 68-84, December 2017. [In Russian] https://periodicals.karazin.ua/mia/article/view/10098

Золотухін Володимир Олександрович
студент магістратури
Харківський національний університет ім. В.Н. Каразіна, майдан Свободи, 4, Харків, Україна , 61022
e-mail: zolotukhinvolodymyr@gmail.com
https://orcid.org/0009-0004-0303-0624

Яновський Володимир Володимирович
д. ф-м. н., професор
Харківський національний університет ім. В.Н. Каразіна, майдан. Свободи, 6, Харків, Україна , 61022
Інститут монокристалів, Національна Академія Наук України, проспект Науки, 60., Харків, Україна, 61001.
e-mail: yanovsky@isc.kharkov.ua
https://orcid.org/0000-0003-0461-749X

Вплив порушення демократії стратегій з пам’яттю на еволюцію популяції

Актуальність. Брак довіри у сучасному суспільстві часто є тормозить розвиток людства та, іноді, ставить під питання майбутнє людської популяції в цілому. Протягом всієї історії розвитку суспільства спостерігається явище, коли та чи інша ідея захоплює розуми людей, що призводить до того, що вони притримуються однакової (або дуже схожої) поведінкової стратегії. Для покращення розуміння внутрішніх процесів у суспільстві, в якому порушено рівномірність розподілу стратегій по членам популяції, необхідні детальні дослідження, які неможливі без відповідного програмного забезпечення.
Мета. Метою роботи є дослідження впливу кількості агнетів певної стратегії на результат еволюції популяції в цілому. Досліджується характер змін в еволюції за умови поступового, монотонного збільшення агентів певної стратегії від 1 агнета до 10% від демократичної популяції. Дослідження також має на меті встановити стратегії, що є еволюційно доцільними тільки за умови збільшення їх носіїв у популяції.
Методи дослідження. Розглянуто еволюцію популяції з повним набором стратегій поведінки, обмежених тільки глибиною пам’яті 2,зі збільшеною кількістю агентів певної стратегії. Кожен носій агент з кожним, включаючи себе згідно з ітеративною моделлю дилеми ув’язненого. Винагороди визначаються за матрицями виплат. Кожне наступне покоління популяції послідовно втрачає агентів найбільш невигідної стратегії поведінки попереднього покоління. Агенти, що є носіями обраної стратегії взаємодіють між собою та з іншою популяцією за стандартним законом. Розглянуто декілька стратегій, кількість агентів яких було збільшено. Серед них стратегії зі складністю нижче ніж середня складність популяції, вище ніж середня складність популяції. Також було розглянуто варіант, коли збільшилась кількість агентів стратегії, що перемогла у демократичному суспільстві.
Результати. В роботі показано, як наявність виділеної стратегії зі збільшеною кількістю носіїв впливає на динаміку популяції. Виявлено зростання кінцевого середнього заробітку популяції. Встановлено, що збільшення кількості агентів не приводить до перемоги стратегії, що не перемогли у демократичній популяції.
Висновки. За результатами роботи визначено головні наслідки впливу кількості агентів певної стратегії на еволюцію популяції.

Ключові слова: *агресивність, еволюція, популяція, стратегія, складність, суспільство*

Надійшла у першій редакції 27.07.2023, в останній - 12.08.2023.
The first version has been received on 27.07.2023, the final version - on 12.08.2023.

DOI: https://doi.org/10.26565/2304-6201-2023-59-04

УДК 681.31

Мірошник Анатолій Миколайович *аспірант кафедри автоматики і управління в технічних системах Національного технічного університету «Харківський політехнічний університет», м. Харків, вул. Кирпичова,2, Україна, 61000*
e-mail: anatolii.miroshnyk@nure.ua
https://orcid.org/0000000157029611

Качанов Петро Олексійович *докт. техн. наук, професор; професор ЗВО кафедри автоматики і управління в технічних системах Національного технічного університету «Харківський політехнічний університет», м. Харків, вул. Кирпичова,2, Україна, 61000*
e-mail: petro.kachanov@khpi.edu
https://orcid.org/0000-0002-7532-5913

Ситнік Борис Тимофійович *к.т.н., доцент, доцент ЗВО кафедри інформаційних технологій Українського державного університету залізничного транспорту м. Харків, площа Фейєрбаха,7, Україна, 61000*
e-mail: bts12021947@gmail.com
http://orcid.org/0000-0002-9664-5617

Синтез структури та моделювання адаптивних цифрових формуючих фільтрів

У роботі обґрунтовано метод автоматичної ідентифікації дисперсії випадкових корисних сигналів та випадкових перешкод із заданими значеннями спектрально-кореляційних характеристик, що дозволяє визначати поточні оцінки дисперсії та їх зміну для довільних випадкових впливів з невідомими характеристиками. Показано, що параметричний вихід адаптивного цифрового фільтра можна використовувати при автоматичному коригуванні параметрів регулятора в контурі регулювання системи управління в діапазоні адаптації коефіцієнта адаптації з урахуванням розрядності перетворювачів АЦП и ЦАП, що розширює сферу застосування запропонованого методу ідентифікації.

Актуальність. Актуальність роботи полягає у можливості синтезу структури та моделювання адаптивних цифрових формуючих фільтрів.

Методи дослідження. Основним методом дослідження є метод автоматичної ідентифікації дисперсії випадкових корисних сигналів та випадкових перешкод із заданими значеннями спектрально-кореляційних характеристик, що дозволяє визначати поточні оцінки дисперсії та їх зміну для довільних випадкових впливів з невідомими характеристиками.

Результати. Показано, що параметричний вихід адаптивного цифрового фільтра можна використовувати при автоматичному коригуванні параметрів регулятора в контурі регулювання системи управління в діапазоні адаптації коефіцієнта адаптації з урахуванням розрядності перетворювачів АЦП и ЦАП, що розширює сферу застосування запропонованого методу ідентифікації.

Висновки. Обґрунтовано метод автоматичної ідентифікації дисперсії випадкових корисних сигналів та випадкових перешкод із заданими значеннями спектрально-кореляційних характеристик, що дозволяє визначати поточні оцінки дисперсії та їх зміну для довільних випадкових впливів з невідомими характеристиками. Автоматична ідентифікація статистичних параметрів випадкових сигналів і перешкод у формуючому адаптивному фільтрі дозволяє враховувати їх зміну в оптимальних параметрах налаштування систем управління. Отримано формули розрахунку оптимального значення постійного часу адаптивного фільтра в залежності від коефіцієнта адаптації *Kb*, що характеризують оцінку поточного відношення рівнів корисного сигналу та перешкоди. Показано, що параметричний вихід адаптивного цифрового фільтра можна використовувати при автоматичному коригуванні параметрів регулятора в контурі регулювання системи управління в обмеженому діапазоні змін амплітудних і частотних характеристик корисного сигналу та перешкод в діапазоні адаптації коефіцієнта адаптації при заданих співвідношеннях або з урахуванням розрядності перетворювачів АЦП та ЦАП, що розширює сферу застосування запропонованого методу ідентифікації. Результати моделювання на програмній моделі адаптивного цифрового фільтру (m-файл програми модулювання) та графіки результатів моделювання показало високий коефіцієнт придушення перешкод у всьому діапазоні його зміни та зміну рівня завад на виході АФ в залежності від рівня вхідного сигналу. При збільшенні рівня вхідних завад рівень вихідних завад зменшується. При послідовному з'єднанні АФ загальний коефіцієнт адаптації *Kb*Kb*...*автоматично збільшується.

Ключові слова: *структурно-параметрична ідентифікація, моделі індексної ідентифікації, адаптивна система керування, високошвидкісний рух, завадозахищеність, формуючі адаптивні фільтри, формуючі фільтри моделі, формуючі фільтри регулятора*.

Як цитувати: Мірошник А. М., Качанов П. О., Ситнік Б. Т. Синтез структури та моделювання адаптивних цифрових формуючих фільтрів. *Вісник Харківського національного університету імені В.Н.Каразіна, серія Математичне моделювання. Інформаційні технології. Автоматизовані системи управління*. 2023. вип. 59. С.35-48. https://doi.org/10.26565/2304-6201-2023-59-04

How to quote: Miroshnyk A., Kachanov P., Sytnik B., "Structure synthesis and modeling of adaptive digital shaping filters", *Bulletin of Kharkiv National University named after V. N. Karazin, series Mathematical modeling. Information Technology. Automated control systems*, 2023. Vol. 59. S.35-48. https://doi.org/10.26565/2304-6201-2022-59-04 [In Ukrainian].

1 Вступ

Методи та способи опрацювання інформації, математичні моделі обчислювальних процесів, технології виконання обчислень, в тому числі високопродуктивних, безпечних, автономних, адаптивних, інтелектуальних, архітектура та організація функціонування відповідних програмно-технічних засобів.

2 Постановка задачі

Завданням дослідження є розроблення методу автоматичної ідентифікації статистичних параметрів випадкових сигналів та перешкод, визначення поточних сигналів оцінок дисперсії та меж змін довільних випадкових впливів з невідомими характеристиками на параметричному виході адаптивного фільтру, що дозволить враховувати їх зміну в оптимальних параметрах налаштування регуляторів систем управлінням.

Метою дослідження є розроблення нової моделі та алгоритму автоматичної ідентифікації параметрів випадкових сигналів із заданими значеннями спектрально-кореляційних характеристик, що дозволить визначати поточні оцінки параметрів та їх зміну для корегування налаштувань адаптивних регуляторів.

3 Огляд літератури

Аналіз розвитку цифрових автоматичних систем (ЦАС) [1 - 16] показує, що протягом усього періоду їх існування відбувається безперервне підвищення вимог до стійкості, безпеки, якості, швидкодії та точності їх роботи, зростання числа виконуваних функцій, схемотехнічне та алгоритмічне ускладнення.

В результаті виникли проблеми, що важко реалізуються традиційними засобами. До таких проблем, перш за все, відноситься побудова оптимального варіанту проектованої системи, що задовольняє ряду проти шкодливих вимог, пов'язаних з впливом на систему детермінованих і випадкових корисних сигналів і перешкод (шумів), точне значення та зміна характеристик яких заздалегідь передбачити не можна . Передбачити можна лише у статистичному сенсі. Як приклад можна провести систему [16], що складається з генератора, навантаженого на велику кількість споживачів. Споживачі включаються і вимикаються випадково в часі. У цьому випадку струм генератора представляє випадкову функцію часу, а система регулювання генератор-регулятор напруги буде піддаватися випадковим впливам.

Вирішення цих проблем можливе при широкому використанні систем автоматичного проектування (САПР), найважливішими елементами яких є моделювання випадкових процесів із заданими значеннями параметрів їх статистичних характеристик і синтез структур адаптивних фільтрів, що автоматично ідентифікують ці параметри [9, 15, 17-33].

Автоматична ідентифікація статистичних параметрів випадкових сигналів та перешкод дозволяє враховувати їх зміну в оптимальних параметрах налаштування систем управління.

У загальному випадку функціональна схема алгоритму імітаційного цифрового моделювання може бути розбита на три основні блоки: алгоритм формування зовнішніх впливів та реалізацій випадкових корисних вхідних сигналів та випадкових перешкод (шумів) із заданими спектрально-кореляційними характеристиками; алгоритм функціонування синтезованої системи автоматичної ідентифікації поточних оцінок змінних параметрів статистичних характеристик цих впливів; алгоритм обробки та ідентифікації результатів цифрового моделювання.

4 Виклад основного матеріалу

Синтез структури безперервного формуючого адаптивного фільтра із заданою спектрально-кореляційною характеристикою проведено в [9]. Однак ця структура не враховує розрядності перетворювачів АЦП і ЦАП, а також умови стікості при переходе до цифрового фільтру. Для стаціонарного випадкового процесу (білого шуму) $S_f(\omega)=Q$ має постійний спектр Q у смузі частот ω від $-\omega_\beta$ до $+\omega_\beta$, отримано значення дисперсії Df , яке дорівнює

$$D_f = \frac{1}{2\pi}\int_{-\infty}^{\infty} S_f(\omega)d\omega = \frac{1}{2\pi}\int_{-\omega_\beta}^{\omega_\beta} Qd\omega = \frac{Q\Delta\omega}{2\pi} = \frac{Q\omega_\beta}{\pi}. \quad (4.1)$$

Графік спектральної щільності білого шуму з постійним спектром в обмеженій смузі частот наведено на рисунку 4.1.

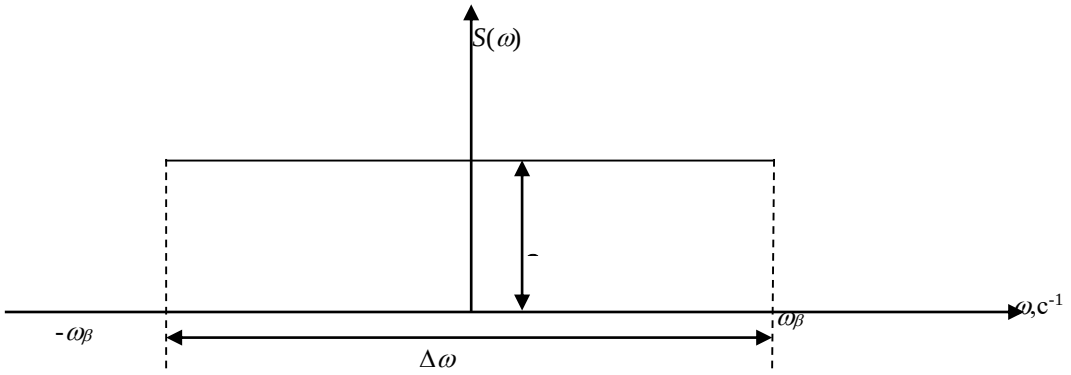


Рис.4.1 Графік спектральної щільності білого шуму в обмеженій смузі частот

Кореляційна функція визначається на основі інтегралу Фур'є:

$$R(\tau) = \frac{1}{2\pi}\int_{-\infty}^{\infty} S_f(\omega)e^{j\omega t}d\omega = \frac{1}{\pi}\int_{0}^{\omega_\beta} \mathrm{Q}\cos\omega\tau d\omega = \frac{Q}{\pi\tau}\sin\omega_\beta\tau\,. \quad (4.2)$$

Значення дисперсії (1) можна визначити виразом

$$D_f = R(0) = \lim_{\tau\to 0}\frac{Q}{\pi\tau}\sin\omega_\beta\tau = \frac{Q\omega_\beta}{\pi}. \quad (4.3)$$

Перевагою розглянутого уявлення випадкового процесу, як має обмежений спектр, у тому [3, 5, 9], що з нього дисперсія похідних всіх порядків обмежені. Це випливає з того, що дисперсія похідної порядку *m* дорівнює

$$D^{(m)} = \frac{1}{\pi}\int_{0}^{\omega_\beta} Q\omega_\beta^{2m}d\omega = \frac{Q\omega_\beta^{2m+1}}{\pi(2m+1)} = \frac{D\omega_\beta^{2m}}{2m+1}. \quad (4.4)$$

З [3, 5] випливає, що якщо задані, наприклад, допустимі значення дисперсії випадкового сигналу D іи його m-й похідний $D^{(m)}$, то з (4) можна знайти допустиме значення необхідної постійної часу формує фільтра $T_\beta = \frac{1}{\omega_\beta}$. Наприклад, для $D^{(1)}$ і D

$$T_\beta = \frac{1}{\omega_\beta} = \sqrt{\frac{D}{3D^{(1)}}}\,. \quad (4.5)$$

Якщо на вхід фільтра з передавальною функцією

$$H(s) = \frac{1}{Ts+1} \quad (4.6)$$

діють перешкоди у вигляді білого шуму з необмеженою смугою пропускання $S_0(\omega)=S_f(\omega)=Q$, то спектральна щільність вихідного сигналу фільтра за всіма частотами дорівнює

$$S_1(\omega) = \left|H(j\omega)\right|^2 S_0(\omega) = \frac{Q}{\left|T_\beta j\omega+1\right|^2} = \frac{Q}{T_\beta^2\omega^2+1}.$$

Інтегрування спектральної щільності вихідного сигналу за всіма частотами дає значення дисперсії вихідного сигналу:

$$D_1 = \frac{Q}{2\pi}\int_{-\infty}^{\infty}\frac{d\omega}{\left|T_\beta j\omega+1\right|^2} = \frac{Q}{2\pi T_\beta} = \frac{Q\omega_\beta}{\pi}. \quad (4.7)$$

Порівнюючи формули (1) і (7), отримуємо $D_1=D_f$. Таким чином, відфільтрований у смузі частот $\pm\omega_\beta$ випадковий корисний сигнал виходить пропусканням білого шуму з необмеженою смугою пропускання через фільтр із постійною часу

$$T_\beta = \frac{1}{\omega_\beta}.$$

Припущення, що на виході фільтра із заздалегідь невідомою передавальною функцією $H(p)$ необхідно отримати випадковий корисний сигнал із заданою, наприклад, з експоненційною кореляційною функцією

$$R_2(\tau) = D_2 e^{-\omega_\beta(\tau)}, \tag{4.8}$$

спектральна щільність вихідного випадкового корисного процесу знаходиться за інтегралом Фур'є

$$S_2(\omega) = \int_{-\infty}^{\infty} D_2 e^{-\omega_\beta(\tau)} e^{j\omega\tau} d\tau = \frac{2\omega_\beta D_2}{\omega_\beta^2+\omega^2} = \frac{2T_\beta D_2}{1+\omega^2 T_\beta^2} \quad . \tag{4.9}$$

Графік кореляційної функції білого шуму з обмеженою смугою пропускання наведено на рис. 2.

Такий випадковий процес можна отримати, якщо випадковий сигнал зі спектральною щільністю $S_0(\omega)$ (білий шум) пропустити через фільтр з частотною передатною функцією $\Phi(j\omega)$.

Тоді на виході фільтра з'явиться випадковий сигнал із спектральною щільністю $S_2(\omega)$:

$$S_2(\omega) = \left|\Phi(j\omega)\right|^2 S_0(\omega). \tag{4.10}$$

З цього виразу можна визначити модуль необхідної частотної передавальної функції фільтра за відомими спектральними щільностями вихідного та вхідного сигналів:

$$\left|\Phi(j\omega)\right| = \sqrt{\frac{S_2(\omega)}{S_0(\omega)}}. \tag{4.11}$$

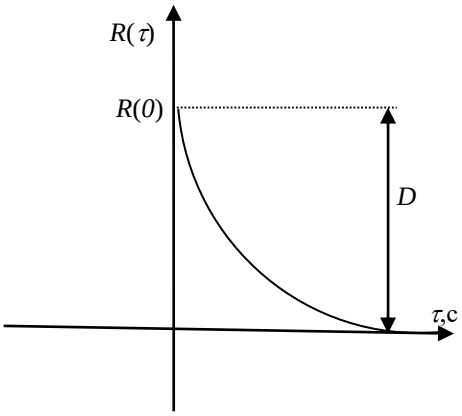


Рис. 4.2 Графік кореляційної функції білого шуму з обмеженою смугою пропускання

Після встановлення значень $S_2(\omega)$ і $S_0(\omega)$ із (9) і (7) отримаємо модуль шуканої частотної передавальної функції:

$$\left|\Phi(j\omega)\right| = \sqrt{\frac{2T_\beta D_2}{\left|T_\beta j\omega+1\right|^2 D_1}} = \frac{R}{T_\beta j\omega+1}, \tag{4.12}$$

$$\text{де } R = \sqrt{\frac{2T_\beta D_2}{D_1}}. \tag{4.13}$$

Цьому модулю частотної передавальної функції, що шукається, відповідає передатна функція фільтра

$$\Phi(s) = \frac{R}{T_\beta s+1}. \tag{4.14}$$

Отже, випадкові процеси з обмеженим білим шумом і експоненційною кореляційною функцією формуються з білого шуму за допомогою аперіодичних ланок (6) і (14) першого порядку з однаковими постійними часу T_β і різними статичними коефіцієнтами (6) і (12).

5 Синтез оптимальних параметрів налаштування фільтра заданої структури

Для того щоб випадковий корисний сигнал проходив через оптимальний фільтр з була мінімальною помилкою по корисному сигналу, необхідно, щоб спектральна щільність корисного сигналу $S_x(\omega)$ укладалася в смугу пропускання фільтра.

Для зменшення впливу перешкоди на систему необхідно, щоб її смуга пропускання не перевищувала смугу пропускання спектральної щільності корисного сигналу або автоматично змінювалася відповідно до заданого (оптимального) співвідношення сигнал/шум.

Нехай на вхід замкнутої системи (фільтра) з передавальною функцією в розімкнутому стані

$$S_x(\omega)\,K(s)=\frac{1}{Ts} \tag{5.15}$$

діє вхідний сигнал *x(t),* що є випадковою нестаціонарною функцією часу із спектральною щільністю

$$S_x(\omega)=\frac{R}{T_\beta^2\omega^2+1}, \tag{5.16}$$

та перешкода *f*(*t*), яка являє собою білий шум зі спектральною щільністю $S_f(\omega)=Q$, причому сигнали *x(t)* та *f(t)* статистично незалежні.

У силу незалежності *x(t)* та *f(t)* вираз для дисперсії помилки D_e, зване середнім ризиком *r* [3, 5], складається з двох доданків: одного, обумовленого дисперсією корисного сигналу D_{ex}, та другого, обумовленого перешкодою D_{ef}.

$$r=D_e=D_{ex}+D_{ef}=\frac{1}{\pi}\int_0^\infty\left|H_e(j\omega)\right|^2 S_x(\omega)d\omega+\frac{1}{\pi}\int_0^\infty\left|H(j\omega)\right|^2 S_f(\omega)d\omega, \tag{5.17}$$

де $H_e(\omega)$ и $H(\omega)$ – частотні передавальні функції помилково і замкнутої системи відповідно.

Для отримання поточної спектральної густини випадкового корисного сигналу

$$S_x(\omega)=\frac{R}{(1+jT_\beta\omega)(1-jT_\beta\omega)} \tag{5.18}$$

необхідно білий шум із спектральною щільністю *Q* пропустити через формуючий фільтр із передатною функцією [1-5, 9]

$$W_\phi(s)=\frac{R}{T_\beta s+1}. \tag{5.19}$$

У виразі (17) для *Dex* замість $H_e(j\omega)$ необхідно підставити

$$H_{ex}(j\omega)=H_e(j\omega)W_\phi(j\omega). \tag{5.20}$$

Знайдемо $H_e(s)=\dfrac{1}{1+K(s)}=\dfrac{Ts}{Ts+1}$ и $H(s)=\dfrac{K(s)}{1+K(s)}=\dfrac{1}{Ts+1}$. (5.21)

Підставляючи у вираз для $H_{ex}(p)$ значення $H_e(s)$ и $W_\phi(s)$, отримаємо

$$H_{ex}(s)=\frac{Ts}{(Ts+1)(T_\beta s+1)}=\frac{Ts}{TT_\beta s^2+(T+T_\beta)s+1}. \tag{5.22}$$

Підставляючи отримані вирази у вираз (17) отримаємо

$$r=D_e=D_{ex}+D_{ef}=\frac{R}{\pi}\int_0^\infty\left\{\left(\frac{Ts}{TT_\beta s^2+(T+T_\beta)s+1}\right)^2\Bigg|_{s=j\omega}\frac{d\omega}{}+\right.$$

$$\left.+\frac{Q}{\pi}\int_0^\infty\left(\frac{1}{1+Ts}\right)^2\Bigg|_{s=j\omega}\frac{d\omega}{}\right\}=\frac{RT}{2T_\beta(T+T_\beta)}+\frac{Q}{2T}. \tag{5.23}$$

Знайдемо оптимальну постійну часу *T*, що визначає мінімальне значення середнього ризику *r = De* min, прирівнюючи її похідну за параметром *T* нулю.

$$\frac{\partial r}{\partial T} = \frac{\partial D_e}{\partial T} = \frac{R(T+T_\beta) - RT}{2T_\beta (T+T_\beta)^2} - \frac{Q}{2T^2} = 0. \tag{5.24}$$

Навівши вираз (24) до спільного знаменника і прирівнявши чисельник нулю, отримаємо

$$T_\beta \left[(R-Q)T^2 - 2QT_\beta T - QT_\beta^2 \right] = 0. \tag{5.25}$$

Тоді

$$T^2 - 2\frac{Q}{R-Q} T_\beta T - \frac{Q}{R-Q} T_\beta^2 = 0, \tag{5.26}$$

звідки знайдемо оптимальне значення шуканої постійної часу

$$T_{onm} = \frac{T_\beta}{R-Q}\left(Q + \sqrt{RQ}\right) = \frac{T_\beta}{\sqrt{\frac{R}{Q}} - 1}, \tag{5.27}$$

підставляючи значення *Tonm* з (27) у формулу (23) отримаємо мінімальне значення дисперсії помилки *Demin=Deonm*, відповідне мінімальної дисперсії стекла за корисним сигналом і мінімальною дисперсією перешкоди на виході системи, що відповідає максимальній дисперсії перешкоди в сигналі помилки.

Тоді отримаємо

$$D_{exonm} = \frac{RT_{onm}}{2T_\beta (T_{onm} + T_\beta)} = \frac{\sqrt{RQ}}{2T_\beta}, \tag{5.28}$$

$$D_{ef\,onm} = \frac{Q}{2T_{onm}} = \frac{\sqrt{QR} - Q}{2T_\beta}. \tag{5.29}$$

Таким чином, дисперсія помилки системи при оптимальному налаштуванні T_{onm} фільтра дорівнює:

$$r_{e\min} = D_{eonm} = D_{exonm} + D_{ef\,onm} = \frac{\sqrt{RQ} + \sqrt{RQ} - Q}{2T_\beta}. \tag{5.30}$$

Дисперсія вихідного корисного сигналу Dyx при оптимальному налаштуванні визначається за формулою

$$D_{yxonm} = \frac{1}{\pi}\int_0^\infty \left|H_{onm}(j\omega)\right|^2 \frac{R}{T_\beta^2\omega^2 + 1} d\omega = \frac{R}{2(T_{onm} + T_\beta)} =$$

$$= \frac{R}{2\left(\frac{T_\beta}{\sqrt{\frac{R}{Q}} - 1} + T_\beta\right)} = \frac{R\left(\sqrt{\frac{R}{Q}} - 1\right)}{2T_\beta\sqrt{\frac{R}{Q}}} = \frac{\sqrt{R}\left(\sqrt{R} - 1\right)}{2T_\beta\sqrt{Q}}. \tag{5.31}$$

Розділивши дисперсію вихідного сигналу фільтра по корисному сигналу *Dyx* на дисперсію перешкоди в сигналі помилки *Def* при оптимальному налаштуванні фільтра *T=Tonm* отримаємо коефіцієнт адаптації K_β, який характеризує оцінку поточного відношення рівнів корисного сигналу до перешкоди (відношення сигнал/шум). Зміна спектрально-кореляційних характеристик корисного сигналу та перешкоди викличе зміну коефіцієнта адаптації K_β та встановлення нового оптимального значення постійної часу адаптивного фільтра за формулою:

$$T_{опт} = \frac{T_\beta}{K_\beta - 1}, \tag{5.32}$$

де, $K_\beta = \frac{D_{ух}}{D_{ef}} = \frac{R\sqrt{Q}}{Q\sqrt{R}} = \sqrt{\frac{R}{Q}}$.

6 Реалізація структури та аналіз стійкості цифрових аналогів формують фільтрів із заданою спектрально-кореляційною характеристикою

Для визначення дисперсії Dxy і Def цифрових фільтрів використовуються відомі формули [3, 5]

$$D_{xy} = \lim_{N\to\infty} \frac{1}{N+1} \sum_{n=0}^{N} y^2[nT_k] = \sqrt{R} \quad . \tag{6.33}$$

і

$$D_{ef} = \lim_{N\to\infty} \frac{1}{N+1} \sum_{n=0}^{N} e^2[nT_k] = \sqrt{Q}, \tag{6.34}$$

де $y^2[nT_k]$ та $e^2[nT_k]$ – квадрати гратчастих функцій відліків вихідного сигналу y(t) та помилки e(t) у моменти часу nT_k, T_k – період вимірювання гратчастих функцій y[nTk] та $e[nT_k]$, $n=0,1,2\ldots\infty$.

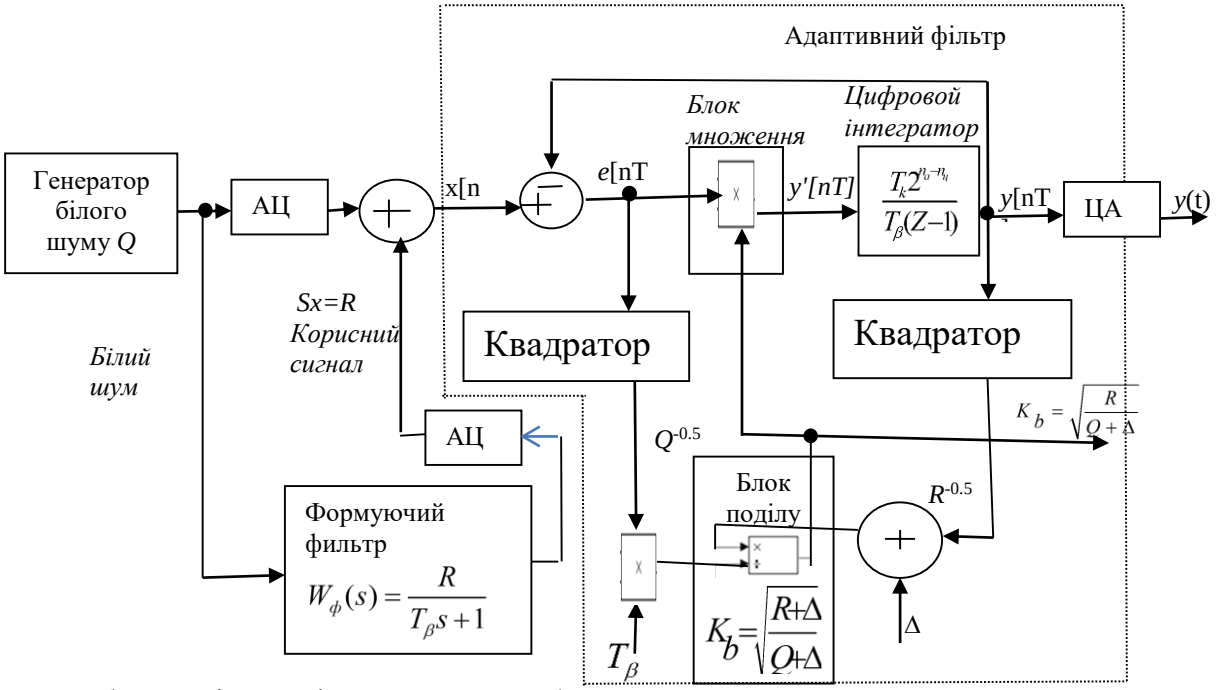
При переході до цифрових аналогів ланок структурних елементів фільтрів моделі навіть структурно стійкі ланки першого порядку можуть втрачати стійкість за певних співвідношеннях параметрів цих ланок і періодом дискретизації.

Аналіз стійкості цифрових фільтрів проведемо на прикладі дискретизації структурно стійкої безперервної аперіодичної ланки першого порядку.

Для реалізації цифро-аналогової імітаційної моделі адаптивного фільтра пропонується використовувати модель, структурна схема якої наведена на рис. 3.

$$W_p(Z) = \frac{T_k}{T_{опт}(Z-1)} 2^{n_a - n_ц} \tag{6.36}$$

.(



БП – блок поділу; І – інтегратор; БМ – блок множення; Δ - мала величина, що вводиться в дільник для виключення поділу на нуль

Рис. 6.3 Структурна схема пропонованої цифро-аналогової імітаційної моделі цифрового адаптивного фільтра

Вихідний сигнал блоку БП поділу дорівнює поточному значенню коефіцієнта адаптації K_β і може бути використаний для автопідстроювання параметрів налаштування цифрових регуляторів при зміні оптимального значення постійної часу адаптивного фільтра. Послідовним з'єднанням адаптивних фільтрів може бути реалізована структура адаптивного фільтра вищого порядку.

Передаточна функція розімкнутого цифрового фільтра з функцією передачі *K(s)* з екстраполятором *We(s)* (рисунок 4.3) має вигляд

$$W_p(s) = W_e(s)K(s)\mathrm{K}_{\text{а–ц}} = \frac{1-e^{-st}}{s}\cdot\frac{1}{T_{опт}s}2^{n_a-n_ц}, \qquad (6.18)$$

де n_a и $n_ц$ - розрядності перетворювачів АЦП і ЦАП,

$\mathrm{K}_{\text{а–ц}} = 2^{n_a-n_ц}$ - підсумковий статичний коефіцієнт передачі перетворювачів АЦП та ЦАП.

Z-перетворення цієї передавальної функції має вигляд

Z-передаточна функція замкнутого контуру (суматор. блок множення, інтегратор) цифро-аналогової імітаційної моделі цифрового адаптивного фільтра має вигляд

$$H(Z) = \frac{T_k 2^{n_a-n_ц}}{T_{опт}\left(Z-1+\dfrac{T_k 2^{n_a-n_ц}}{T_{опт}}\right)} = \frac{A(Z)}{B(Z)}. \qquad (6.19)$$

Для аналізу стійкості розглянемо характеристичне рівняння

$$B(Z) = Z-1+\frac{T_k 2^{n_a-n_ц}}{T_{опт}} = 0. \qquad (38)$$

Застосувавши до цього рівняння білінійне перетворення, отримаємо

$$B(W) = \frac{1+W}{1-W}-1+\frac{T_k 2^{n_a-n_ц}}{T_{опт}} = 0$$

і

$$\frac{T_k 2^{n_a-n_ц}}{T_{опт}}W + 2 - \frac{T_k 2^{n_a-n_ц}}{T_{опт}} = 0. \qquad (6.20)$$

По критерию Гурвица эта система устойчива, если выполняется условие положительности коэффициентов $a_0>0$ и $a_1>0$,

де $a_0 = \dfrac{T_k 2^{n_a-n_ц}}{T_{опт}}$, а $a_1 = 2-\dfrac{T_k 2^{n_a-n_ц}}{T_{опт}}$, то есть $0 < \dfrac{T_k 2^{n_a-n_ц}}{T_{опт}} < 2$.

Отже, для стійкості цифрового адаптивного фільтра необхідно, щоб постійні часу T_β, $T_{опт} = \dfrac{T_\beta}{K_\beta - 1}$, коефіцієнт адаптації $K_\beta = \dfrac{D_{ух}}{D_{ef}} = \sqrt{\dfrac{R}{Q}}$ перебували у таких межах $0 < \dfrac{1}{T_{опт}} < \dfrac{2}{T_k 2^{n_a-n_ц}}$ або

$$0 < K_\beta < \frac{2^{n_ц-n_a+1}T_\beta}{T_k} + 1 \qquad (6.21)$$

При прагненні до нуля складової оцінки дисперсії за корисним сигналом ($Q\to 0$), ця нерівність може бути порушена, і в цьому випадку в адаптивному фільтрі виникнуть нестійкі розбіжності коливання. Для їх усунення необхідно до величини Q додати Δ, що обмежує діапазон адаптації

коефіцієнта $K_\beta = \sqrt{\frac{R}{Q+\Delta}}$ при заданому відношенні $0 < \sqrt{\frac{R}{Q+\Delta}} < \frac{2^{n_ц - n_a + 1} T_\beta}{T_k} + 1$ з урахуванням розрядності перетворювачів АЦП та ЦАП.

Вихідний сигнал блоку БД поділу дорівнює поточному значенню K_β коефіцієнта адаптації і може бути використаний для автопідстроювання параметрів налаштування цифрових регуляторів при зміні оптимального значення постійної часу цифрового фільтру адаптивного фільтра.

Послідовним з'єднанням адаптивних фільтрів може бути реалізована структура адаптивного фільтра вищого порядку.

7. Програмна модель дослідження адаптивного цифрового фільтру (m-файл програми модювання) та графіки результатів моделювання

```
clear
clc

m_xam=1.0;        %Амплітуда вхідного сигналу фільтра
m_xch=2.0;        %Частота вхідного сигналу фільтра
m_t=0.001;
m_mod=10.0;
m_ti=0.01;
m_d=0.1;
m_m=1.0;

NI=m_mod/m_t;
% m_d=Δ----------Цикл моделювання----------
n=0:0.001:9.999;

Y=0;
YP=0;
E=0;
F=0;
Y1=0;
A=rand(NI,1);
B=rand(NI,1);
V(1:NI,1)=sqrt((-2)*log(A)).*cos(2*pi*B);
filter_out(NI,1)=0;
forN1=1:NI
F(N1,1)=V(N1,1);
if N1==1
E(N1,1)=X(N1,1)-filter_out(N1,1)+F(N1,1);
else
E(N1,1)=X(N1,1)-filter_out(N1-1,1)+F(N1,1);
end
EA(N1,1)=E(N1,1)*E(N1,1);
ED(N1,1)=EA(N1,1)+m_d;
if N1 ==1
YA(N1,1)=filter_out(N1,1)^2+m_d;
else
YA(N1,1)=filter_out(N1-1,1)^2+m_d;
end
Y1(N1,1)=E(N1,1)*YA(N1,1)/(ED(N1,1));
Kb(N1,1)=YA(N1,1)/(m_d+ED(N1,1));
if N1>1
filter_out(N1,1)=(0.5*m_t/m_ti)*(Y1(N1,1)+Y1(N1-
1,1))+filter_out(N1-1,1);
end
```

```
end
```

Результати моделювання на програмной моделі адаптивного цифрового фільтру (m-файл програми модювання), які наведені на графіках моделювання наступні:

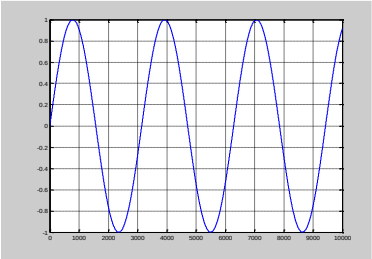
Вхідний сигнал X=1sin(2n)

Вхідний сигнал завад **F(N1,1)=±3.5*sqrt((-2)*log(A)).*cos(2*pi*B), де A=rand(NI,1), B=rand(NI,1);**

Вихідний сигнал Y с завадою **filter_out±E(N1,1)**=1±(від 0.1 до 1.5)sin(2n), тобто рівень завад зменьшується при зростанні амплитуди сигналу

Вихідний сигнал змін коефіцієнта адаптації ***Kb≈від 2.3 до 35***

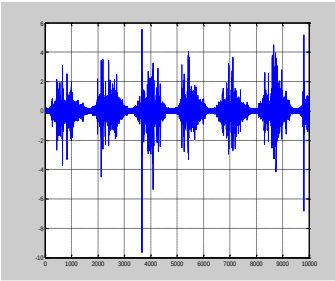
```
X=m_xam*sin(m_xch*n)
;
plot(X)
grid on
```



```
X=X';
```

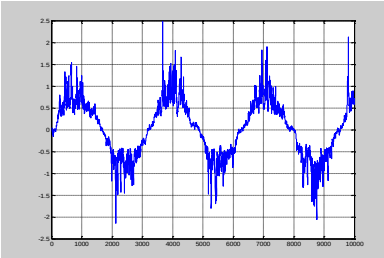
%Вхідний сигнал

```
plot(Y1)
grid
```



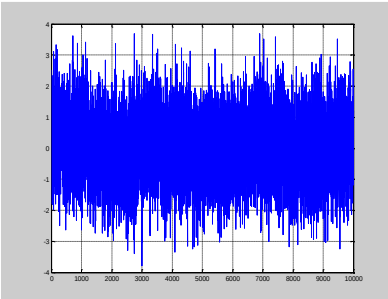
% Сигнал похидної вхідного сигналу Y1

```
plot(filter_out)
grid
```



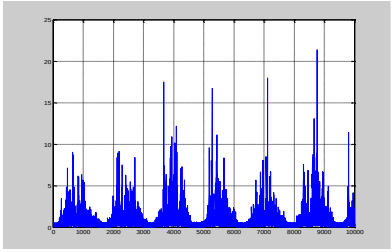
%Вихідний сигнал с завадою filter_out±E(N1,1)

```
plot(E)
grid
```



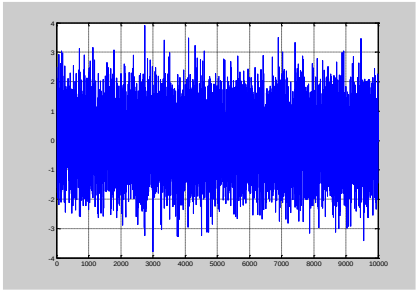
%Вихідний сигнал помилки E(N1,1)≈F(N1,1)

```
plot(Kb)
grid
```



%Вихідний сигнал змін коефіцієнта адаптації *Kb*

```
plot(F)
grid
```



%Вхідний сигнал завад *F*

8. Висновки

У цій статті було обґрунтовано метод автоматичної ідентифікації дисперсії випадкових корисних сигналів та випадкових перешкод із заданими значеннями спектрально-кореляційних характеристик, що до-зволяє визначати поточні оцінки дисперсії та їх зміну для довільних випадкових впливів з неві-домими характеристиками. Автоматична ідентифікація статистичних параметрів випадкових сигналів і перешкод у формуючому адаптивному фільтрі дозволяє враховувати їх зміну в оптимальних параметрах налаштування систем управління. Отримано формули розрахунку оптимального значення постійного часу адаптивного фільтра в залежності від коефіцієнта адаптації *Kb*, що характеризують оцінку поточного відношення рівнів корисного сигналу та перешкоди. Показано, що параметричний вихід адаптивного цифрового фільтра можна використовувати при автоматичному коригуванні параметрів регулятора в контурі регулювання системи управління в обмеженому діапазоні змін амплітудних і частотних характеристик корисного сигналу та перешкод в діапазоні адаптації коефіцієнта адаптації при заданих співвідношеннях або з урахуванням розрядності перетворювачів АЦП та ЦАП, що розширює сферу застосування запропонованого методу ідентифікації. Результати моделювання на програмній моделі адаптивного цифрового фільтру (m-файл програми модування) та графіки результатів моделювання показало високий коефіцієнт придушення перешкод у всьому діапазоні

його зміни та зміну рівня завад на виході АФ в залежності від рівня вхідного сигналу. При збільшенні рівня вхідних завад рівень вихідних завад змішується. При послідовному з'єднанні АФ загальний коефіцієнт адаптації Kb*Kb*…автоматично збільшується.

СПИСОК ЛІТЕРАТУРИ

1. Небылов А.В. Гарантирование точности управления. -М.: Наука. Физматлит, 1988.-304с.

2. Исследование цифровых автоматических систем. Лабораторный практикум. \ Под ред. А.В. Небылова. - СПб.: Издательство С.-Петербургского университета, 1996.- 192с.

3. Бесекерский В.А. Цифровые автоматические системы.- М.: Наука,1976.-576 с. https://b.eruditor.link/file/2129497/

4. Острем К., Витенмарк Б. Системы управления с ЭВМ: Пер. с англ. - М.: Мир, 1987.-480 с. https://lib-bkm.ru/12515

5. Микропроцессорные системы автоматического управления. \ Под ред. В.А.Бескерского.- Л.: Машиностроение, 1988.-365 с. https://libarch.nmu.org.ua/handle/GenofondUA/48188

6. Изерман Р. Цифровые системы управления М.: Мир, 1984.-541с. https://lib-bkm.ru/12325

7. Проектирование цифровых устройств на однокристальных микропроцессорах.\ В.В. Сташин, А.В. Урусов, О.Ф. Мологонцева.- М.: Энергоатомиздат, 1990.-224 с. https://electronics.lnu.edu.ua/course/mikroprotsesorna-tekhnika-153-mikro-ta-nanosystemna-tekhnika

8. Куо Б. Теория и проектирование цифровых систем управления. - М.: Машиностроение, 1986.449 с. https://b.eruditor.link/file/18241/

9. Сытник Б.Т. Синтез структуры и моделирования адаптивных цифровых фильтров и систем управления с нестационарными характеристиками. Часть 1./Сытник, В.Б. Сытник//Інформаційні керуючі системи залізничного транспорту. – 2003. – N6. – С. 18 – 24. http://jiks.kart.edu.ua/article/view/265542

10. Солодовников В.А. Микропроцессорные автоматические системы регулирования. -М.:Высшая школа,1991г.-255с.

11. Алексеенко А.Г., Галицын А.Д., Иванников А.Д. Проектирование радиоэлектронной аппаратуры на микропроцессорах: Программирование, типовые решения, методы отладки. -М.: Радио и связь, 1984. -272с.

12. Романенко В.Д., Игнатенко Б.В. Адаптивное управление технологическими процессами на базе микро-ЕОМ: Учеб. пособ. -К.: Выща школа, 1990. -334с.

13. Загарий Г.И., Шубладзе А.М. Синтез систем управления на основе критерия максимальной степени устойчивости. -М.: Энергоатомиздат, 1988. -104с.
https://www.researchgate.net/signup.SignUp.html

14. Пат. № 4063 Україна Пристрій для вимірювання параметрів інерційних ланок / винахідники: Г. І. Загарій, Б. Т. Ситнік, І. В. Гусєв, А. В. Мамонов, Б. С. Левочко, П. В. Гусєв; володілець: Харківська державна академія залізничного транспорту; заявл. 07.02.1990; опубл. 27.12.1994, Бюл. № 6-І. http://lib.kart.edu.ua/handle/123456789/8478

15. Пат. № 11427 Україна Адаптивний фільтр / винахідники : Г. І. Загарій, Б. Т. Ситнік, Б. С. Левочко, А. В. Мамонов, І. В. Гусєв, П. В. Гусєв, В. С. Коновалов, В. Г. Пороцкій ; володілець : Харківська державна академія залізничного транспорту; заявл. 13.03.1989; опубл. 25.12.1996, Бюл. № 4. http://lib.kart.edu.ua/handle/123456789/8478

16. Вилькевич Б.И. Автоматическое управление электрической передачей и электрические схемы тепловозов. -М.: Транспорт, 1987. -272с.
https://libarch.nmu.org.ua/handle/GenofondUA/70858

17. Антонью А. Цифровые фильтры: анализ и проектирование. – М.: Радио и связь, 1983. – 320 с. https://skylots.org/6562863709/Antonyu+A+Cifrovye+filtry+Analiz+i+proektirovanie

18. Бендат Дж., Пирсол А. Прикладной анализ случайных данных. – М.: Мир, 1989. – 540 с. https://book-i-nist.com/view.php?book_id=18544

19. Блейхут Р. Быстрые алгоритмы цифровой обработки сигналов. – М.: Мир, 1989. – 448 с. http://bamper.info/972-blejhut_r_bystrye_algoritmy_cifrovoj_obrabotki_si.html

20. Гольденберг Л.М. и др. Цифровая обработка сигналов: Справочник. - М.: Радио и связь, 1985.- 312 с. http://booksshare.net/index.php?id1=4&category=physics&author=goldenber-lm&book=1985

21. Гольденберг Л.М. и др. Цифровая обработка сигналов: Учебное пособие для вузов. - М.: Радио и связь, 1990.- 256 с. https://studizba.com/files/show/djvu/3622-1-gol-denberg-l-m-matyushkin-b-d-polyak-m.html

22. Гутников В.С. Фильтрация измерительных сигналов. – Л.: Энергоатомиздат, 1990. – 192 с. https://bon.ua/ru/obyavlenie/gutnikov-v-s-filtraciya-izmeritelnyh-signalov-e3edab

23. Даджион Д., Мерсеро Р. Цифровая обработка многомерных сигналов. – М.: Мир, 1988. – 488 с. https://libarch.nmu.org.ua/handle/GenofondUA/58914

24. Дмитриев В.И. Прикладная теория информации: Учебник для студентов вузов. - М.: Высшая школа, 1989.- 325 с. https://studfile.net/preview/953348/

25. Купер Дж., Макгиллем А. Вероятностные методы анализа сигналов и систем. – М.: Мир, 1989. – 376 с. http://booksshare.net/index.php?id1=4&category=biol&author=kuper-dg&book=1989

26. Макс Ж. Методы и техника обработки сигналов при физических измерениях: В 2-х томах. - М.: Мир, 1983. https://b.eruditor.link/file/130058/

27. Оппенгейм А.В., Шафер Р.В. Цифровая обработка сигналов. – М.: Связь, 1979. – 416 с. https://b.eruditor.link/file/3758471/

28. Рабинер Л., Гоулд Б.Теория и применение цифровой обработки сигналов. – М.: Мир, 1978. – 848 с. https://www.geokniga.org/books/10680

29. Хемминг Р.В. Цифровые фильтры. – М.: Недра, 1987. – 221 с. http://elib.kstu.kz/lib/document/IBIS/1215BA04-CE80-417A-A46C-3D28F47B1CC0/

30. Васильев Д.В. Радиотехнические цепи и сигналы: Учебное пособие для вузов. - М.: Радио и связь, 1982. - 528 с. https://b.eruditor.link/file/243537/

31. Зиновьев А.Л., Филиппов Л.И. Введение в теорию сигналов и цепей: Учебное пособие для вузов. - М.: Высшая школа, 1975. - 264 с. https://crafta.ua/lots/6537702886-zinovev-al-filippov-li-vvedenie-v-teoriyu-signalov-i-cepyay

32. Адаптивные фильтры. /Под ред. К.Ф.Н.Коуэна и П.М.Гранта. – М.: Мир, 1988, 392 с. http://repository.vsau.org/getfile.php/5343.pdf

33. Айфичер Э., Джервис Б. Цифровая обработка сигналов. Практический подход. / М., "Вильямс", 2004, 992 с. https://studizba.com/files/show/djvu/2295-1-ayficher-e-dzhervis-b-cifrovaya.html

REFERENCES

1. Nebylov A.V. Guaranteed precision control. -M.: Science. Fizmatlit, 1988.-304s.

2. Research of digital automatic systems. Laboratory workshop. \ Ed. A.V. Nebylova. - St. Petersburg: St. Petersburg University Publishing House, 1996.- 192 p..

3. Besekersky V.A. Digital automatic systems. - M.: Nauka, 1976.-576 p..

4. Ostrem K., Vitenmark B. Computer control systems: Transl. from English - M.: Mir, 1987.-480 p.

5. Microprocessor automatic control systems. \ Ed. V.A. Beskersky.- L.: Mechanical Engineering, 1988.-365 p.

6. Izerman R. Digital control systems M.: Mir, 1984.-541s.

7. Design of digital devices on single-chip microprocessors.\ V.V. Stashin, A.V. Urusov, O.F. Mologontseva.- M.: Energoatomizdat, 1990.-224 p.

8. Kuo B. Theory and design of digital control systems. - M.: Mechanical Engineering, 1986. – 449 p.

9. Sytnik B.T. Synthesis of structure and modeling of adaptive digital filters and control systems with non-stationary characteristics. Part 1. / Sytnik, V.B. Sytnyk //I nformation core systems of health transport. – 2003. – N6. – C. 18 – 24.

10. Solodovnikov V.A. Microprocessor automatic control systems. -M.: Higher School, 1991.-255 p.

11. Alekseenko A.G., Galitsyn A.D., Ivannikov A.D. Design of electronic equipment on microprocessors: Programming, standard solutions, debugging methods. -M.: Radio and communication, 1984. -272 p.

12. Romanenko V.D., Ignatenko B.V. Adaptive control of technological processes based on micro-EOM: Textbook. allowance -K.: Vyshcha School, 1990. -334 p.

13. Zagariy G.I., Shubladze A.M. Synthesis of control systems based on the criterion of the maximum degree of stability. -M.: Energoatomizdat, 1988. -104 p.

14. Пат. № 4063 Україна Пристрій для вимірювання параметрів інерційних ланок / винахідники: Г. І. Загарій, Б. Т. Ситнік, І. В. Гусєв, А. В. Мамонов, Б. С. Левочко, П. В. Гусєв; володілець: Харківська державна академія залізничного транспорту; заявл. 07.02.1990; опубл. 27.12.1994, Бюл. № 6-І. http://lib.kart.edu.ua/handle/123456789/8478

15. Пат. № 11427 Україна Адаптивний фільтр / винахідники : Г. І. Загарій, Б. Т. Ситнік, Б. С. Левочко, А. В. Мамонов, І. В. Гусєв, П. В. Гусєв, В. С. Коновалов, В. Г. Пороцкій ; володілець : Харківська державна академія залізничного транспорту; заявл. 13.03.1989; опубл. 25.12.1996, Бюл. № 4. http://lib.kart.edu.ua/handle/123456789/8478

16. Vilkevich B.I. Automatic control of electrical transmission and electrical circuits of diesel locomotives. -M.: Transport, 1987. -272 p.

17. Anthony A. Digital filters: analysis and design. – M.: Radio and Communications, 1983. – 320 p.

18. Bendat J., Peirsol A. Applied analysis of random data. – M.: Mir, 1989. – 540 p.

19. Bleikhut R. Fast algorithms for digital signal processing. – M.: Mir, 1989. – 448 p.

20. Goldenberg L.M. and others. Digital signal processing: Handbook. - M.: Radio and communication, 1985.- 312 p.

21. Goldenberg L.M. and others. Digital signal processing: Textbook for universities. - M.: Radio and communication, 1990.- 256 p.

22. Gutnikov V.S. Filtering of measurement signals. – L.: Energoatomizdat, 1990. – 192 p.

23. Dajion D., Mersereau R. Digital processing of multidimensional signals. – M.: Mir, 1988. – 488 p.

24. Dmitriev V.I. Applied information theory: Textbook for university students. - M.: Higher School, 1989.- 325 p.

25. Cooper J., McGillem A. Probabilistic methods for analyzing signals and systems. – M.: Mir, 1989. – 376 p.

26. Max J. Methods and technology of signal processing in physical measurements: In 2 volumes. - M.: Mir, 1983.

27. Oppenheim A.V., Shafer R.V. Digital signal processing. – M.: Svyaz, 1979. – 416 p.

28. Rabiner L., Gould B. Theory and application of digital signal processing. – M.: Mir, 1978. – 848 p.

29. Hemming R.V. Digital filters. – M.: Nedra, 1987. – 221 p.

30. Vasiliev D.V. Radio engineering circuits and signals: Textbook for universities. - M.: Radio and communication, 1982. - 528 p.

31. Zinoviev A.L., Filippov L.I. Introduction to the theory of signals and circuits: A textbook for universities. - M.: Higher School, 1975. - 264 p.

32. Adaptive filters. /Ed. C. F. N. Cowan and P. M. Grant. – M.: Mir, 1988, 392 p.

33. Ayficher E., Jervis B. Digital signal processing. Practical approach. / M., "Williams", 2004, 992 p.

Miroshnyk Anatolii *graduate student of the department "Automation and control in technical systems, ACTS", National Technical University, Kharkiv Polytechnic Institute, Kharkiv, Kirpychova St.,2, 61002*
e-mail: anatolii.miroshnyk@nure.ua
https://orcid.org/0000000157029611

Kachanov Petro *Doctor of Technical Sciences, Professor, Professor of the Department of Higher Education automation and control in technical systems of the National Technical University "Kharkiv Polytechnic University", Kharkiv, Kirpychova St.,2, 61002*
e-mail: petro.kachanov@khpi.edu.ua
https://orcid.org/0000-0002-7532-5913

Sytnik Borys *associate professor, associate professor of the Department of Information Technologies of the Ukrainian State University of Railway Transport, Kharkiv, Feuerbacha Square, 7, Ukraine, 61000.*
e-mail:bts12021947@gmail.com
http://orcid.org/0000-0002-9664-5617

Structure synthesis and modeling of adaptive digital shaping filters

The paper substantiates the method of automatic identification of the variance of random useful signals and random interference with given values of spectral-correlation characteristics, which allows determining the current estimates of the variance and their change for arbitrary random influences with unknown characteristics. It is shown that the parametric output of the adaptive digital filter can be used to automatically adjust the controller parameters in the control loop of the control system in the adaptation range of the adaptation coefficient, taking into account the bit depth of the ADC and DAC converters, which expands the scope of the proposed identification method.
Relevance. The relevance of the work lies in the possibility of synthesizing the structure and modeling of adaptive digital shaping filters.
Research methods. The main research method is the method of automatic identification of the variance of random useful signals and random interference with given values of spectral-correlation characteristics, which allows determining the current estimates of the variance and their change for arbitrary random influences with unknown characteristics.
Results. It is shown that the parametric output of the adaptive digital filter can be used to automatically adjust the controller parameters in the control loop of the control system in the adaptation range of the adaptation coefficient, taking into account the bit depth of the ADC and DAC converters, which expands the scope of the proposed identification method.
Conclusions. A method for automatic identification of the variance of random useful signals and random interference with given values of spectral-correlation characteristics has been substantiated, which allows determining current estimates of the variance and their change for arbitrary random influences with unknown characteristics. Automatic identification of the statistical parameters of random signals and interference in the forming adaptive filter allows taking into account their change in the optimal parameters of control system tuning. The formulas for calculating the optimal value of the adaptive filter constant time depending on the adaptation coefficient Kb, which characterize the assessment of the current ratio of the levels of the useful signal and the interference, are obtained. It is shown that the parametric output of the adaptive digital filter can be used to automatically adjust the controller parameters in the control loop of the control system in a limited range of changes in the amplitude and frequency characteristics of the useful signal and interference in the adaptation range of the adaptation coefficient at specified ratios or taking into account the bit depth of the ADC and DAC converters, which expands the scope of the proposed identification method. The simulation results on the software model of the adaptive digital filter (m-file of the modulation program) and the graphs of the simulation results showed a high interference suppression coefficient in the entire range of its change and a change in the interference level at the AF output depending on the input signal level. When the input noise level increases, the output noise level decreases. When the AFs are connected in series, the overall adaptation coefficient Kb*Kb*...automatically increases.
Keywords: *structural-parametric identification, index identification models, adaptive control system, high-speed motion, interference immunity, adaptive adaptive filters, regulator adaptive filters.*

Надійшла у першій редакції 28.08.2023, в останній - 22.09.2023.
The first version has been received on 28.08.2023, the final version - on 22.09.2023.

DOI: https://doi.org/10.26565/2304-6201-2022-54-01

УДК (UDC) 004.93

Малига Ігор Євгенійович *аспірант*
Харківський Національний Університет ім. В.Н. Каразіна, майдан Свободи 4, Харків, Україна, 61022
e-mail: igormalyga@gmail.com;
https://orcid.org/0000-0002-5708-7739

Шматков Сергій Ігорович *д.т.н., професор; завідуючий кафедри теоретичної теоретичної та прикладної системотехніки*
Харківський Національний Університет ім. В.Н. Каразіна, майдан Свободи 4, Харків, Україна, 61022
e-mail: s.shmatkov@karazin.ua
https://orcid.org/0000-0002-6328-988X
Scopus Author ID: 57203141869

Аналіз впливу різних векторних представлень слів на точність класифікації текстових даних

Актуальність. Зростання обсягу доступної текстової інформації в Інтернеті та інших джерелах створює необхідність у вдосконаленні методів обробки тексту для ефективного аналізу та використання цих даних. Векторне представлення слів визначається як ключовий елемент у цьому контексті, оскільки воно дозволяє перетворювати слова у числові вектори, зберігаючи семантичні відносини. З розвитком сучасних методів машинного навчання, особливо глибокого навчання, векторні представлення слів стали важливим елементом для покращення результатів моделей в обробці текстових даних. Такі моделі вимагають якісних та семантично насичених векторних представлень.Усе це визначає актуальність вивчення впливу різних векторних представлень слів на обробку текстових даних та виявлення оптимальних методів для конкретних завдань.

Мета: Мета даної статті полягає в систематичному аналізі впливу різних методів векторизації слів на результати обробки текстових даних. Дослідження спрямоване на визначення оптимальних підходів до векторної репрезентації слів для покращення ефективності та точності моделей обробки тексту в різноманітних завданнях штучного інтелекту та машинного навчання.

Методи дослідження. Аналіз, експеримент.

Результати. Виявлено, що, не дивлячись на значний прогрес у технологіях машинного навчання, проблеми семантики та контексту при обробці текстових даних все ще мають місце. Вони впливають на якість і точність рішень, прийнятих системами, заснованими на машинному навчанні, що може привести до неправильного аналізу і викривлення даних. Виявлено, що навіть сучасні моделі на основі трансформерів можуть зіткнутися з викликами розуміння семантики та контексту, особливо у складних і багатозначних сценаріях.

Висновки. На основі проведеного дослідження було зроблено висновки, що проблема семантики та контексту в обробці текстових даних є суттєвою і вимагає подальшого вивчення. Існуючі методи і технології, хоча і показують високі результати в деяких задачах, можуть бути недостатніми в інших, особливо складних, ситуаціях. Пропонується продовжити дослідження в цій області, розробляти нові методи і підходи, які б можливо, будуть здатні ефективно вирішувати ці проблеми. Також важливим є вивчення того, як різні контекстуальні фактори впливають на семантику текстових даних та як ці впливи можна врахувати при проектуванні та використанні систем машинного навчання.

Ключові слова: *Машинне навчання, обробка природної мови, семантика, контекст, текстові дані, нейронні мережі, трансформери, BERT, GPT-3, аналіз даних, аналіз сентименту, семантичний аналіз.*

Як цитувати: Малига І. Є., Шматков С. І. *Аналіз впливу різних векторних представлень слів на точність класифікації текстових даних. Вісник Харківського національного університету імені В.Н.Каразіна, сер. «Математичне моделювання. Інформаційні технології. Автоматизовані системи управління»*. 2023. вип. 59. С.49-55. https://doi.org/10.26565/2304-6201-2023-59-05

How to quote: Malyga I.E., Shmatkov S.I., *Analysis of the influence of different word vector representations on the accuracy of text data classification, Bulletin of V.N. Karazin Kharkiv National University, series Mathematical modelling. Information technology. Automated control systems*, vol. 59, pp.49-55, 2023.https://doi.org/10.26565/2304-6201-2022-53-01 [In Ukrainian].

1 Вступ

З розширенням обсягів текстової інформації у сучасному цифровому світі виникає важлива задача оптимізації обробки цих даних. Одним із ключових аспектів цього процесу є використання векторних представлень слів. В контексті штучного інтелекту та машинного навчання, де точність та ефективність моделей залежать від репрезентації слів, розуміння впливу різних методів векторизації стає надзвичайно актуальним завданням. У цій статті ми систематично аналізуємо вплив різних векторних представлень слів на результати обробки текстових даних, визначаючи оптимальні підходи для різноманітних завдань у сфері обробки тексту.

2 Постановка проблеми в загальному вигляді та її зв'язок із важливими науковими чи технічними завданнями. Огляд публікацій з цієї проблеми.

У сучасному високотехнологічному середовищі, де обсяг та різноманітність текстової інформації динамічно зростають, виникає нагальна потреба вдосконалення методів обробки текстових даних. Ключовою проблемою є вибір оптимального методу векторизації слів, який дозволяє представити слова у векторній формі для подальшого використання в алгоритмах обробки природної мови (Natural Language Processing, NLP). Вірність цього представлення безпосередньо впливає на точність та ефективність моделей NLP. Наукові дослідження демонструють, що якість векторних представлень слів має вирішальне значення для результатів завдань обробки тексту. Один із прикладів — у роботі "Efficient Estimation of Word Representations in Vector Space" (Mikolov et al., 2013), де Word2Vec надав широкий простір для розвитку методів векторизації слів. Проте, із зростанням кількості доступних методів, виникає необхідність визначення стратегій вибору та налаштування їх для різних завдань обробки тексту.

Обговорення також зосереджується на зростаючому об'ємі текстової інформації у віртуальному просторі, який збільшує необхідність вибору оптимальних методів для розв'язання завдань обробки тексту в реальному часі. Зараз ця проблема набуває ще більшого значення, оскільки вимагає розробки ефективних та точних стратегій векторизації слів для високопродуктивних систем NLP.

Актуальність даної проблематики визначається високим попитом на точні та ефективні системи обробки тексту у різноманітних сферах. Такі системи використовуються від підтримки прийняття рішень у бізнесі до автоматизації інтеракції із користувачем в інтелектуальних асистентах. Таким чином, наукове дослідження впливу різних методів векторизації слів на результати обробки тексту є стратегічно важливим для розвитку та оптимізації сучасних систем NLP. Дана проблема пов’язана з наступними науковими та технічними завданнями:

1. Розробка ефективних методів векторизації. Перше ключове наукове завдання - розробка методів векторизації слів, які б забезпечували ефективні та точні результати для різноманітних текстових даних. Це включає в себе вивчення та розробку нових алгоритмів, які враховують семантичні та синтаксичні властивості текстів.
2. Адаптація до мовних та культурних особливостей тексту. Дана задача стосується адаптації методів векторизації до мовних різниць та культурних особливостей. Наявність універсальних моделей, які можуть ефективно працювати в різних лінгвістичних умовах, є великим викликом.
3. Оптимізація алгоритмів та архітектур глибокого навчання. Ефективність та швидкодія важливі для застосувань у реальному часі та обробці великих обсягів даних.
4. Створення універсальних методів векторизації. Метою цбого завданнярозробка методів векторизації, які можуть адаптуватися до різних мов, жанрів та видів текстів. Це включає в себе створення моделей, що здатні працювати на текстах різних дисциплін та контекстів.

Дослідження в області переносу знань та адаптації методів векторизації для різних мовніх умов зазнає значного розвитку. Публікація "Cross-Lingual Word Embeddings" від Forsyth та Ropkins стала ключовим внеском у розумінні того, як можна застосовувати існуючі моделі для різних мов. Вони розглядають важливі аспекти переносу знань у векторних представленнях слів, що виявляється критичним у розвитку універсальних методів.

Останніми часами спостерігається зростання інтересу до застосування глибокого навчання для векторизації слів. У статті "Deep Learning Approaches for Word Embeddings" Бенджіо та Сакура розглядають використання рекурентних та трансформерних нейронних мереж для отримання векторних представлень слів. Вони аналізують переваги цих підходів та їхній вплив на точність та універсальність аналізу текстових даних.

Ключовим етапом у розумінні поточних викликів у векторній репрезентації слів є стаття "Challenges and Future Directions in Word Embeddings Research" від Лін та Ян. Вони докладно розглядають критичні аспекти існуючих методів та вказують на прогалини, які потребують уваги. Крім того, вони звертають увагу на важливість роботи з мовним різноманіттям та множинністю стилів використання мови.

3 Виділення невирішених раніше частин загальної проблеми, котрим присвячується означена стаття, з обґрунтуванням актуальності рішення. Дослідження за темою інших авторів

Хоча векторні представлення слів, такі як Word2Vec, GloVe та FastText, вже активно використовуються в галузі обробки природної мови (Natural Language Processing, NLP), існують певні аспекти, які залишаються недостатньо дослідженими. Одним з таких аспектів є вплив цих представлень на конкретні типи NLP задач, зокрема на задачі, пов'язані з фінтеком, юридичними текстами, та медичними записами. Ці області вимагають високої точності та специфічного розуміння мови, що робить їх особливо чутливими до вибору векторного представлення.

З недавнім появою контекстно-залежних моделей, таких як BERT та GPT, виникає питання про взаємодію та порівняння ефективності цих сучасних підходів із традиційними векторними представленнями. Цей аспект залишається відносно недослідженим, зокрема в контексті адаптації цих моделей до специфічних застосувань.

Враховуючи стрімкий розвиток технологій NLP та постійне зростання обсягу даних, які потребують обробки, важливо розуміти, як різні векторні представлення можуть впливати на результати обробки цих даних. Це особливо актуально в таких критичних галузях, як фінанси, право та медицина, де вибір найбільш ефективного представлення може мати значний вплив на точність та надійність систем NLP. Крім того, розуміння взаємодії між традиційними векторними представленнями та новітніми контекстно-залежними моделями може відкрити нові напрямки в дослідженні та розробці більш продуктивних та точних систем обробки мови.

Дослідження інших авторів по даній темі:

1. Word2Vec і GloVe: Праці Мікалова та співавторів (2013) по Word2Vec та Пеннінгтона та співавторів (2014) по GloVe заклали основи для розуміння контекстуальних зв'язків у текстах.
2. FastText: Бозіде та співавторів (2016) представили FastText, який розширив підход Word2Vec, забезпечуючи краще розуміння морфології слів.
3. Спеціалізовані Домени: Недостатньо досліджена область, але роботи таких авторів, як Ченг та співавторів (2016) в медичному NLP, показують потенціал специфічних підходів.

4 Формулювання мети статті, постановка завдання.

Головна мета цієї статті полягає у глибокому аналізі впливу різних векторних представлень слів, таких як Word2Vec, GloVe, та FastText, на ефективність обробки текстових даних, з особливим акцентом на домен відгуків на оголошення у мережі. Стаття також має на меті порівняти ці методи з сучасними контекстно-залежними моделями, наприклад BERT та GPT, для оцінки їхньої відносної ефективності у цій конкретній сфері.

Для досягнення цієї мети, стаття передбачає виконання наступних завдань:

1. Експериментальна Верифікація: Провести експерименти, використовуючи реальні набори даних відгуків, щоб підтвердити теоретичні висновки та визначити найбільш ефективні підходи для конкретного домену.
2. Аналіз отриманих результатів. На основі отриманих результатів провести їх аналіз та дати пояснення щодо них.
3. Розробка Рекомендацій: На основі отриманих результатів сформулювати рекомендації щодо оптимального вибору векторних представлень для обробки відгуків на оголошення в мережі.

Завершення цих завдань дозволить отримати детальне розуміння впливу векторних представлень слів на аналіз відгуків та внесе важливий вклад у подальший розвиток галузі обробки природної мови.

5 Виклад основного матеріалу з повним обґрунтуванням отриманих наукових результатів.

5.1 Опис процесу тестування та підготовка даних

У рамках нашого дослідження, ми зосередили увагу на оцінці точності трьох популярних методів векторизації: Word2Vec, GloVe та FastText. Для аналізу ми використали датасет "Large Movie Review Dataset v1.0", що є відомим і широко використовуваним у дослідженнях з обробки природної мови. Цей датасет містить велику кількість позитивних та негативних відгуків на фільми, що робить його ідеальним для оцінки ефективності методів векторизації у задачах класифікації сентименту. Даний датасет обраний через його велику кількість зразків текстів реальних відгуків, що дозволяє провести всебічне тестування моделей, вибірка налічує понад 30000 прикладів.

5.1.1 Очищення та нормалізація даних

Відгуки були очищені від нерелевантних символів, HTML-тегів, знаків пунктуації, а також була проведена нижньорегістрова конвертація. Очищення та нормалізація даних є важливими етапами в процесі обробки текстових даних, особливо при роботі з машинним навчанням та аналізом природної мови. Ці процедури допомагають покращити якість даних та їхню готовність для подальшої обробки. Загальний підхід до очищення даних виглядає наступним чином:

1. Видалення непотрібних символів. Це включає видалення зайвих пробілів, табуляцій, символів нового рядка, а також інших неалфавітних символів, які не несуть важливої інформації для аналізу тексту.
2. Видалення HTML-тегів та URL. Якщо датасет містить HTML-теги або URL, їх слід видалити, оскільки вони можуть вплинути на аналіз тексту.
3. Видалення спеціальних символів та знаків пунктуації: Знаки пунктуації, як-от коми, крапки, лапки тощо, часто видаляються, оскільки вони можуть не нести семантичного значення в контексті деяких задач NLP.

Процес нормалізації даних:

1. Перетворення тексту в нижній регістр. Це допомагає уникнути дублювання слів через різницю в регістрах (наприклад, "Сова" та "сова").
2. Видалення стоп-слі.: Стоп-слова (наприклад, "і", "у", "на") часто видаляються, оскільки вони можуть бути занадто частими та не нести важливої інформації для аналізу.
3. Стемінг та лематизація. Стемінг зменшує слова до їх кореневої форми, тоді як лематизація перетворює слова в їх словникову форму. Обидва ці методи допомагають уніфікувати різні форми слова.
4. Токенізаці.: Перетворення тексту на набір токенів (слів), що є необхідним для багатьох методів NLP.

Для класифікації даних було використано метод логістичної регресії як базової моделі класифікації. Даний метод був обраний через його ефективність та простоту у задачах бінарної класифікації.

Методи векторизіції над якими проводилось тестування та їх особливості:

1. Word2Vec: Метод, заснований на нейронних мережах, що генерує векторні представлення слів, враховуючи їх контекст у великих текстових корпусах.
2. GloVe (Global Vectors for Word Representation): Цей метод зосереджується на агрегації глобальної статистики співвідношення слів у корпусі для виведення векторних представлень.
3. FastText: Підхід, розроблений Facebook, що враховує не тільки слова, але й їх внутрішню структуру (наприклад, нграми), дозволяючи краще обробляти рідкісні слова та слова з помилками.

5.2 Оцінка точності

Оцінка точності у дослідженні методів векторизації (Word2Vec, GloVe, FastText) на датасеті "Large Movie Review Dataset v1.0" проводилася з використанням стандартних підходів у машинному навчанні та обробці природної мови.Датасет спочатку був розділений на навчальну

та тестову вибірки у співвідношенні 80/20. Точність (Accuracy) визначається як відношення кількості правильно класифікованих зразків (істинно позитивних та істинно негативних) до загальної кількості зразків у тестовій вибірці:

$$Точність = \frac{Істинно\ позитивні + Істинно\ негативні}{Загальна\ калькість\ зразків}$$

Також існують альтернативні способи оцінки точності роботи моделей:

1. F1-Скор. Комбінує точність та повноту, є корисним при нерівномірному розподілі класів. Однак, для задач з балансованим розподілом класів, як у нашому випадку, загальна точність може бути більш інтуїтивно зрозумілою.
2. ROC AUC. Вимірює здатність моделі відрізняти класи, але може бути менш прямолінійним для інтерпретації у випадку простих бінарних класифікаційних задач.

5.3 Результати

Результати нашого дослідження показали наступну ефективність векторизації слів у задачі класифікації відгуків:

Word2Vec: Метод показав точність класифікації 81%. Цей метод базується на нейронних мережах та використовує контекстну інформацію для створення векторних представлень слів. Він є популярним в сфері обробки природної мови і відомий своєю здатністю виявляти семантичні відношення між словами. Цей результат вказує на високу ефективність Word2Vec у визначенні семантичних відносин між словами, але також підкреслює обмеження методу в розумінні більш тонких контекстуальних нюансів.

GloVe: З точністю 87% GloVe перевищив Word2Vec. Такий результат можна пояснити більш ефективним аналізом глобальних статистичних відносин у корпусі, що допомогло краще уловити семантичні властивості слів. показав найвищу точність.. Він базується на глобальних статистиках співвідношень між словами у корпусі тексту. Вектори, створені за допомогою GloVe, добре відображають семантичні зв'язки між словами та допомагають в уникненні проблеми "зникнення слів" (word dropout), яка може виникнути при використанні Word2Vec.

FastText: З точністю 85%, FastText також показав сильні результати, переважно завдяки своїй здатності до глибшого аналізу структури слів. Це особливо ефективно для мов, де формування слів має велике значення. Він розширює підхід Word2Vec, додавши здатність векторизувати слова, що складаються із підслів. Це дозволяє FastText краще розрізняти слова з суфіксами та префіксами та використовувати морфологічну інформацію для створення векторних представлень.

Висновки

Аналіз показав, що всі три методи ефективні для векторизації тексту у задачах класифікації сентименту. Однак, GloVe виявився найбільш точним у нашому дослідженні, що може бути пов'язано з його здатністю агрегувати широку статистичну інформацію про слова. FastText також продемонстрував сильні результати, особливо у випадках, де важливе розуміння внутрішньої структури слова. Word2Vec, хоч і показав нижчу точність порівняно з іншими методами, все ще залишається важливим інструментом у сфері обробки природної мови.

5 Висновки

У цій статті ми систематично проаналізували вплив різних методів векторизації слів - Word2Vec, GloVe, FastText - на обробку текстових даних. Кожен з цих методів має свої особливості та переваги в контексті різних задач обробки природної мови.

Дослідження показало, що GloVe демонструє вищу точність порівняно з Word2Vec та FastText для конкретної задачі класифікації сентименту на датасеті "Large Movie Review Dataset v1.0". Це підкреслює важливість вибору відповідного методу векторизації, виходячи з конкретних потреб та характеристик датасету.

Одним з ключових висновків є те, що проблеми семантики та контексту при обробці текстових даних залишаються значущими і вимагають подальшого вивчення. Це особливо важливо для складних сценаріїв, де розуміння глибшого смислу та контексту має вирішальне значення.

Виявлено, що існуючі методи, хоча й ефективні в деяких сценаріях, можуть бути недостатніми для інших, більш складних випадків. Це підкреслює необхідність розробки нових методів та підходів, що зможуть більш ефективно вирішувати проблеми семантики та контексту.

Це дослідження має практичне значення для розробників систем машинного навчання, оскільки воно виділяє ключові проблеми, з якими вони можуть стикатися, та надає напрямки для подальших досліджень.

Стаття відкриває перспективи для подальших досліджень, зокрема у розвитку нових методів машинного навчання, які більше зосереджені на семантиці та контексті, та у пошуку способів інтеграції цих аспектів у існуючі моделі.

Підсумовуючи, дана стаття робить внесок у розуміння впливу різних методів векторизації слів на обробку текстових даних. Отримані результати та аналіз надають цінні інсайти як для теоретичних, так і для практичних аспектів у сфері обробки природної мови та машинного навчання.

СПИСОК ЛІТЕРАТУРИ

1. Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient Estimation of Word Representations in Vector Space. Available at: https://arxiv.org/abs/1301.3781.
2. Bahdanau, D., Cho, K., & Bengio, Y. (2014). Neural Machine Translation by Jointly Learning to Align and Translate. Available at: https://arxiv.org/abs/1409.0473.
3. Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2018). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. Available at: https://www.aclweb.org/anthology/N19-1423/.
4. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is All you Need. Available at: https://papers.nips.cc/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html.
5. Pennington, J., Socher, R., & Manning, C. (2014). Glove: Global Vectors for Word Representation. Available at: https://www.aclweb.org/anthology/D14-1162/.
6. Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., & Dean, J. (2013). Distributed Representations of Words and Phrases and their Compositionality. Available at: https://papers.nips.cc/paper/2013/hash/9aa42b31882ec039965f3c4923ce901b-Abstract.html.
7. Davidov, D., Tsur, O., & Rappoport, A. (2010). Semi-supervised recognition of sarcastic sentences in Twitter and Amazon. Available at: https://aclanthology.org/W10-2914/.
8. Blodgett, S. L., Green, L., & O'Connor, B. (2018). Demographic Dialectal Variation in Social Media: A Case Study of African-American English. Available at: https://aclanthology.org/D16-1120/.
9. Howard, J., & Ruder, S. (2018). Universal Language Model Fine-tuning for Text Classification. Available at: https://www.aclweb.org/anthology/P18-1031/.
10. Peters, M. E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., & Zettlemoyer, L. (2018). Deep contextualized word representations. Available at: https://www.aclweb.org/anthology/N18-1202/.
11. Huang, P. S., He, X., Gao, J., Deng, L., Acero, A., & Heck, L. (2013). Learning Deep Structured Semantic Models for Web Search using Clickthrough Data. Available at: https://posenhuang.github.io/papers/cikm2013_DSSM_fullversion.pdf.
12. Xu C., McAuley J., (2018). The Importance of Generation Order in Language Modeling. Available at: https://www.aclweb.org/anthology/D18-1324/.
13. Suzuki M., Matsuo Y., (2020). A survey of multimodal deep generative models. Available at: https://arxiv.org/abs/2207.02127.
14. Felbo, B., Mislove, A., Søgaard, A., Rahwan, I., & Lehmann, S. (2017). Using Millions of Emoji Occurrences to Learn Any-domain Representations for Detecting Sentiment, Emotion and Sarcasm. Available at: https://www.aclweb.org/anthology/D17-1169/.
15. Reyes A., Rosso P., (2016). Mining Subjective Knowledge from Customer Reviews: A Specific Case of Irony Detection. Available at: https://aclanthology.org/W11-1715.pdf.
16. Yang, Z., Yang, D., Dyer, C., He, X., Smola, A., & Hovy, E. (2016). Hierarchical Attention Networks for Document Classification.. Available at: https://www.aclweb.org/anthology/N16-1174/.
17. Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I. (2019). Better language models and their implications. Available at: https://openai.com/blog/better-language-models/.

18. Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D. M., Wu, J., Winter, C., ... Amodei, D. (2020). Language Models are Few-Shot Learners Available at: https://proceedings.neurips.cc/paper/2020/file/1457c0d6bf5478631ec67e564d04505b-Paper.pdf.
19. Wang, A., Singh, A., Michael, J., Hill, F., Levy, O., & Bowman, S. R. (2019). GLUE: A Multi-Task Benchmark and Analysis Platform for Natural Language Understanding. Available at: https://openreview.net/pdf?id=rJ4km2R5t7.
20. Lu, X., Xiong, C., Parikh, A. P., & Socher, R. (2019). ViLBERT: Pretraining Task-Agnostic Visiolinguistic Representations for Vision-and-Language Tasks. Available at: https://arxiv.org/abs/1908.02265.

Malyha Ihor Yevheniyovych
postgraduate
Kharkiv National University named after V.N. Karazin,4 Svobody Square, Kharkiv, Ukraine, 61022
e-mail: igormalyga@gmail.com;
https://orcid.org/0000-0002-5708-7739

Shmatkov Serhiy Ihorovych
doctor of science, professor; Head of the Department of Theoretical Theoretical and Applied System Engineering
Kharkiv National University named after V.N. Karazin,4 Svobody Square, Kharkiv, Ukraine, 61022
e-mail: s.shmatkov@karazin.ua
Scopus Author ID: 57203141869

Analysis of the influence of different word vector representations on the accuracy of text data classification

Relevance. The growing amount of available textual information from the Internet and other sources creates the need to improve text processing methods for efficient analysis and use of this data. The vector representation of words is defined as a key element in this context, as it allows transforming words into numerical vectors while preserving semantic relations. With the development of modern machine learning methods, especially deep learning, words vector representations have become an important element for improving the results of models in text data processing. Such models require high-quality and semantically rich vector representations. All this determines the relevance of studying the impact of different vector representations of words on text data processing and identifying optimal methods for specific tasks.

Objective: The purpose of this paper is to systematically analyze the impact of different word vectorization methods on the results of text data processing. The study aims to identify optimal approaches to word vector representation to improve the efficiency and accuracy of text processing models in various artificial intelligence and machine learning tasks.

Research methods. Analysis, experiment.

Results. It has been found that despite significant progress in machine learning technologies, the problem of semantics and context in text data processing still exists. This problem affects the quality and accuracy of decisions made by machine learning-based systems, which can lead to incorrect analysis and data distortion. It has been found that even modern transformer-based models may face challenges in understanding semantics and context, especially in complex and ambiguous scenarios.
Conclusions. Based on the study, it was concluded that the problem of semantics and context in text data processing is significant and requires further study. Existing methods and technologies, although showing good results in some tasks, may be insufficient in other, especially complex, situations. It is proposed to continue research in this area, to develop new methods and approaches that might be able to effectively solve these problems. It is also important to study how different contextual factors affect the semantics of textual data and how these influences can be taken into account when designing and using machine learning systems.
***Keywords**: Machine learning, natural language processing, semantics, context, text data, neural networks, transformers, BERT, GPT-3, data mining, sentiment analysis, semantic analysis.*

Надійшла у першій редакції 18.05.2023, в останній - 19.08.2023.
The first version has been received on 18.05.2023, the final version - on 19.08.2023.

DOI: https://doi.org/10.26565/2304-6201-2023-59-06

УДК 65.0(075.8)

Модель мультипаралельної обробки інформації мережевого планування

Толстолузький Євген Дмитрович
аспірант;
Харківський національний університет радіоелектроніки,
проспект Науки, 14, Харків, Україна, 61166
e-mail: evventol@gmail.com
https://orcid.org/0000-0002-2039-0267

У сучасному світі інформаційні технології відіграють все більш важливу роль. Це призводить до зростання кількості IT-проектів. IT-проект - це проект, який має чіткі терміни та мету створення унікального та якісного продукту за встановлені терміни. IT-проекти складаються з різноманітних технологій, обчислювальних та комунікаційних процесів, інформаційних та людських ресурсів. Для ефективного управління такими проектами використовується поняття управління проектами. Управління проектами включає в себе створення та коригування планів, контроль та розподіл ресурсів і задач, створення балансу між проектними обмеженнями. Чим триваліший проект, тим більше ризиків виникає під час його виконання та впровадження. Ці фактори можуть впливати на час розробки проекту, його прибуток, витрачені ресурси, а також втрати та витрати у разі непередбачених ситуацій. Для спеціалістів, які працюють над створенням IT-проектів, будь-які незаплановані питання та витрати можуть стати великою проблемою. Тому розробка нових автоматизованих рішень для управління проектами є актуальним питанням. Такі програмні моделі можуть допомогти мінімізувати витрати часу та розрахувати можливі ризики. Одним з етапів, який можна автоматизувати під час планування робіт та ресурсів, є побудова візуальної моделі виконання робіт у вигляді мережевого графу. У даній роботі розглядається можливість автоматизування процесу побудови мережевого графу з використанням методів мультипаралельної обробки інформації. Даний вид розрахунків може збільшити виграш у часі виконання проекту, налагодити механізм паралельного виконання поставлених задач, а також мінімізувати можливі ризики.

Ключові слова: *проект, автоматизація, менеджер з проектів, мультипаралельна обробка інформації.*

Як цитувати: Толстолузький Є. Д. Модель мультипаралельної обробки інформації мережевого планування. *Вісник Харківського національного університету імені В.Н. Каразіна, серія Математичне моделювання. Інформаційні технології. Автоматизовані системи управління*. 2023. вип. 59. С.56-62. https://doi.org/10.26565/2304-6201-2023-59-06

How to quote: Y. Tolstoluzkyi, “Model of multiparallel information processing for network planning” *Bulletin of V. N. Karazin Kharkiv National University, series Mathematical modeling. Information technology. Automated control systems*, vol. 59, pp. 56-62, 2023. https://doi.org/10.26565/2304-6201-2023-59-01 [In Ukrainian].

1 Вступ

Багато компаній, як великих, так і малих, стикаються з проблемою розподілу ресурсів. Ця проблема може бути вирішена за допомогою процесу планування. Планування дозволяє визначити терміни виконання робіт проекту, кількість необхідних людських та інших ресурсів, а також визначити ризики.

Використання комп'ютерних технологій для планування забезпечує більшу надійність та зменшує кількість людських помилок. Крім того, багато процесів у проекті виконуються паралельно, тому для розрахунку та побудови моделі раціонально використовувати паралельні технології.

Тому тема роботи, пов'язана з розробкою моделі мультипаралельної обробки інформації мережевого планування, є актуальною.

2 Постановка задачі

Планування будь-якого проекту починається з чітко визначеного набору завдань. Для цього необхідно проаналізувати продукт, який планується створити. На першому етапі розробляється MVP (minimum viable product, мінімально життєздатний продукт) – це базова версія продукту, що

вирішує одну з основних задач потенційного клієнта. Після цього можна приступати до створення завдань, які необхідно виконати для реалізації проекту.

Від якості та точності оцінки часу та складності завдань залежить успіх проекту. Якщо завдання будуть переоцінені за часом або недооцінені за складністю, це може призвести до затримок у виконанні проекту або навіть до його провалу.

Для моделі, яка розробляється, дані повинні бути сформовані у відповідності з вимогами. Якщо дані не будуть відповідати вимогам, модель не зможе побудувати мережевий граф та оцінити ризики та терміни виконання.

Вимоги до листа:

- Завдання записуються одне за одним без розривів.
- Показники завдань займають чітке місце у таблиці, як показано на рисунку 1.
- Максимальна кількість задач у листі обмежена максимальною ємністю змінних у мові C++ (приблизно 20000 задач).

Структура таблиці складена відповідно до методології PERT (Project Evaluation and Review Technique). Ця методологія була обрана, оскільки вона найкращим чином застосована для аналізу часу та визначення ризиків проекту, використовується для проектів різної величини та є простою для програмної реалізації.

Методи мультипаралельної обробки інформації мають потужний апарат для розпаралелювання процесів. Це дозволяє автоматично розраховувати та будувати паралельні граф-схеми (мережеві графи) різної глибини, від одногілкових до максимально розпаралелених. Крім того, ці методи дозволяють вибрати найкращий варіант розпаралелювання для кожного проекту.[1-3]

Індекси	Задачі	Описання	Оптимістичний час	Песимістичний час	Фаза виконання	Оцінка важкості	Зв'язки
1	Розробка дизайну головної		14	25		3	0
2	Створення стилю та розмітки головного сайту		4	10		2	1
3	Додавання функцій на сторінку		7	15		4	2
4	Створення бази знань		3	5		2	0
5	Підключення бази знань до сайту		5	9		3	4
6	Оптимізація та організація бази знань		15	28		5	4
7	Заповнення бази знань даними о користувачах і їх ролях		10	20		4	5
8	Додання авторизації на сайт		8	14		4	7
9	Створення мобільної версії сайту		5	18		3	8,3

Рис. 1. Приклад таблиці параметрів мережевого графа

3 Опис моделі

На рисунку 2 представлена модель мультипаралельної обробки інформації мережевого планування у нотації IDEF0. Ця нотація була обрана, оскільки вона призначена для чіткого відображення процесів, залежності кожної роботи від інших та сторонніх вимог. Нотація IDEF0 дозволяє відобразити не часову послідовність, а відношення між роботами.

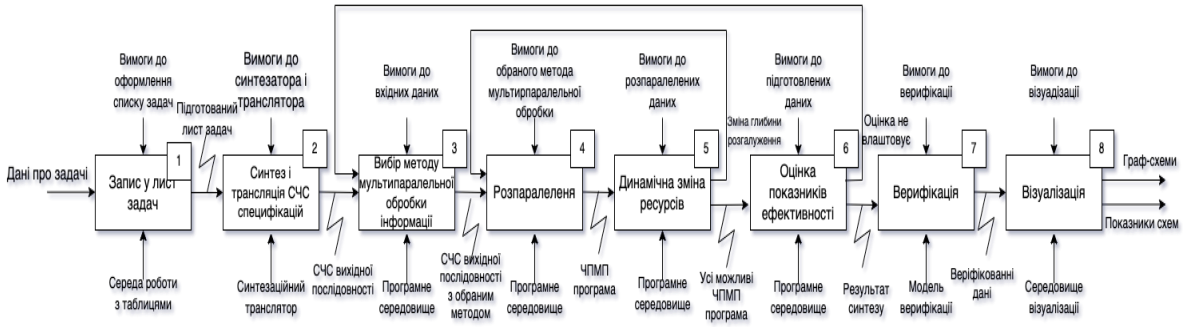


Рис. 2. Модель в системі нотації IDEF0

Розроблена модель містить наступні блоки:

1. блок для запису у лист задач;
2. блок для синтезу і трансляції СЧС (семантико-числових специфікацій);
3. блок для вибору методу мультипаралельної обробки інформації;
4. блок для розпаралелювання;
5. блок для оцінки показників ефективності;
6. блок для динамічної зміни ресурсів;
7. блок для верифікації;
8. блок для візуалізації.

Наведемо більш детальний опис моделі.

Блок 1: структурне планування.

У цьому етапі відбувається обговорення проекту з замовником, розбиття проекту на завдання, визначення термінів та бюджету для кожного завдання. Крім того, відбувається попередня оцінка завдань, їх тривалості, залежності між ними та важливості кожного завдання. Цю роботу можна виконати в будь-якому середовищі, яке працює з таблицями, наприклад Excel. Після оформлення листа завдань, ці дані передаються у блок 2 синтезу та трансляції СЧС. Важливо відзначити, що лист завдань повинен бути формалізованим згідно з вимогами його оформлення. Тобто, у блоці 1 відбувається формування формалізованого листа завдань, який відповідає ідеям замовника.[4]

Блок 2: синтез та трансляція СЧС.

У цьому блоці відбувається створення програми на мові С++, згідно з формалізованим листом задач. Після цього програма подається на синтезаційний транслятор, який генерує таблиці СЧС. Ці таблиці використовуються методами мультипаралельної обробки інформації для конвертації послідовного алгоритму у паралельний. Тобто, блок 2 приймає лист задач та перетворює його на СЧС таблиці, які необхідні для розпаралелювання алгоритму.

Блок 3: вибір методу мультипаралельної обробки інформації.

На цьому етапі обирається метод, який буде використовуватися для розпаралелювання алгоритму. У моделі можуть використовуватися такі методи мультипаралельної обробки інформації:

- метод суміщення незалежних робіт;
- конвеєрний метод;
- декомпозиційний метод;
- метод суміші алгоритмів.

Ці методи розміщені по мірі ускладнення. Наступний метод обирається, якщо обраний метод має незадовільні показники на етапі оцінки. З точки зору планування робіт класичні методи паралельної обробки інформації можуть бути інтерпретовані наступним чином.

Метод суміщення незалежних робіт. Цей метод полягає в тому, що незалежні завдання обробляються паралельно. Тобто, якщо завдання не мають послідовних зв'язків, то вони можуть виконуватися одночасно. Для того, щоб застосувати цей метод, необхідно виконати такі умови: завдання повинні бути незалежними, попередня задача повинна бути завершена до моменту початку виконання наступної задачі. У цьому методі спеціаліст працює над задачами у одній гілці від початку до кінця. Тобто, для кожної окремої гілки потрібен свій спеціаліст.

Конвеєрний метод. Цей метод полягає в тому, що завдання розбивається на фрагменти, які обробляються послідовно. При цьому, кожний фрагмент може виконуватися різними

спеціалістами, залежно від їхньої кваліфікації. Наприклад, якщо на проекті працюють спеціалісти різних рівнів кваліфікації (Junior, Middle, Senior, Architector), то один і той самий фрагмент проекту може бути виконаний різними працівниками по-різному та за різний проміжок часу. У цьому випадку, більш кваліфікований працівник виконує найбільш тривалі фрагменти проекту, у той час коли найлегший, або найкоротший фрагмент виконує менш кваліфікований спеціаліст.

Декомпозиційний метод. Цей метод є покращеною версією конвеєрного методу. Він полягає в тому, що завдання розбивається на частини, які можуть виконуватися різними спеціалістами. При цьому, спеціалісти можуть обмінюватися інформацією та допомагати один одному. Наприклад, якщо є складна задача, над якою працює менш кваліфікований співробітник, то йому в допомогу кожен день на деякий час допомагає більш кваліфікована людина. Ця допомога може бути як поясненням задачі, постановкою конкретних кроків, поясненням незрозумілих моментів, або допомогою в реалізації роботи. Таким чином, декомпозиційний метод дозволяє значно прискорити виконання складних задач, навіть якщо на них працюють спеціалісти з різним рівнем кваліфікації. Цей метод особливо ефективний для великих проектів, де існує багато складних задач.

Метод суміші алгоритмів. Цей метод є поєднанням методу суміщення незалежних робіт та декомпозиційного методу. У цьому методі головна ідея полягає в тому, щоб максимально ефективно використовувати ресурси працівників, щоб уникнути простоїв. Для цього методу необхідно розрахувати максимальне навантаження на співробітника та комбінованість працівників над задачами. Тобто, необхідно визначити, які завдання можуть виконуватися паралельно, а які послідовно. Таким чином, метод суміші дозволяє значно скоротити час виконання проекту, навіть якщо в ньому є складні задачі.

Блок 4: розпаралелювання.

На цьому етапі відбувається:

- побудова множини задач;
- розбиття множини задач на підмножини, які формуються на різних часових ярусах;
- розрахунок моментів початку та закінчення задач;
- прорахунок можливості розгалуження та запуску різних задач на одному і тому ж самому часовому відрізку;
- попередня оцінка складності та тривалості реалізації стислої моделі;
- формування стислої СЧС моделі;
- розрахунок часу та складності виконання.

Тобто, блок 4 приймаючи таблиці СЧС перетворює їх на часопараметризовану модель у середовищі програмування, згідно з обраним методом мультипаралельної обробки інформації.

Блок 5: оцінка показників ефективності.

На цьому етапі відбувається оцінка показників ефективності часопараметризованої мультипаралельної моделі (ЧПММ). Оцінка проводиться за такими показниками:

- час виконання;
- складність виконання;
- критичний шлях;
- показники ризиків.

Оцінка проводиться за допомогою методології PERT.

Якщо оцінки ефективності не влаштовують, то відбувається повторне розпаралелювання, але користуючись іншим методом мультипаралельної обробки. Цей процес може повторюватися кілька разів, поки не будуть отримані задовільні оцінки ефективності. У випадку коли показники усіх методів не задовольняють вимогам, виводиться найкращий з можливих варіантів методу та розпаралелення.

Блок 6: динамічної зміни ресурсів.

На етапі 6 відбувається динамічна зміна ресурсів для оптимізації продуктивності ЧПММ. Для цього використовуються семантико-числові параметри, які враховують зміни ресурсу цифрових систем у динаміці виконання задач. Блок розраховує можливу глибину розгалуження для кожного проекту. Цей показник передається на блок розпаралелення, який робить розгалуження задач у відповідності з даними. На блок оцінки ефективності подаються усі моделі, від послідовної до максимально розгалуженої. Це дозволяє оцінити ефективність різних варіантів розгалуження та вибрати найкращий. [5]

Блок 7: верифікація.

На цьому етапі відбувається перевірка коректності та відповідності результатам формального синтезу ЧПММ.

У якості вхідних даних верифікатора можуть використовуватися:

- структури СЧС моделей;
- часові структури СЧС;
- графічні специфікації часопараметризованих моделей.

Верифікатор виконує такі завдання:

- компіляційна верифікація структур семантико-числової специфікації графа та послідовної програми;
- верифікація структур семантико-числових специфікацій часопараметризованої мультипаралельної моделі;
- перевірка відповідності показників ефективності заданим вимогам та обмеженням.

Блок 8: візуалізація.

На цьому етапі відбувається побудова візуальної складової моделі.

Візуалізатор приймає підготовлені дані та на їхній основі будує такі візуальні елементи:

- часові графи проекту (Рис.3-6);
- дані розрахунків ризиків (Рис.7);
- дані розрахунків ймовірної тривалості проекту (Рис.8).

Ці візуальні елементи дозволяють зрозуміти структуру та поведінку моделі. На наступних рисунках представленні результати візуалізації.

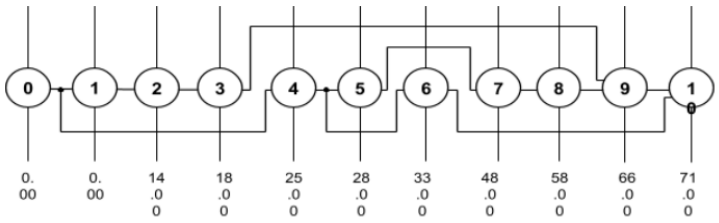


Рис. 3 – Часова послідовна мережева модель планування виконання робіт при мінімальному часі реалізації задач.

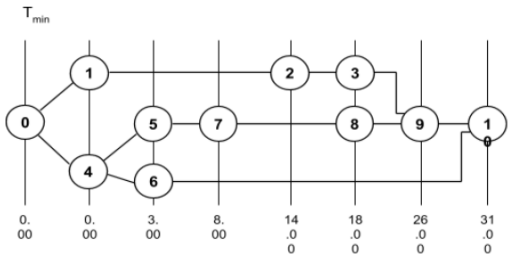


Рис. 4. Часова паралельна мережева модель планування виконання робіт при мінімальному часі реалізації задач.

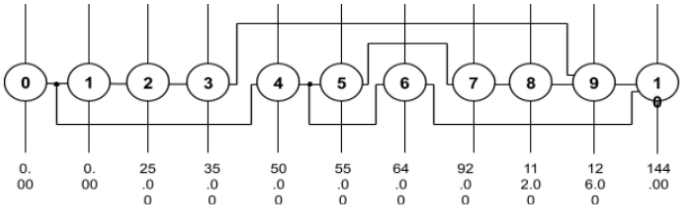


Рис. 5. Часова послідовна мережева модель планування виконання робіт при максимальному часі реалізації задач.

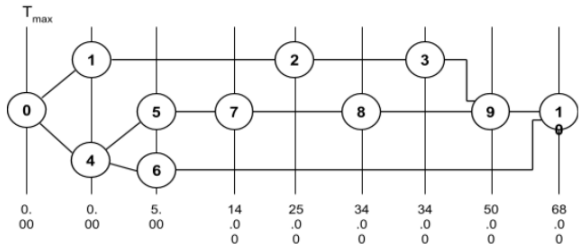


Рис. 6. Часова мережева паралельна модель планування виконання робіт при максимальному часі реалізації задач.

i	j	t^{o}_{i-j}	t^{mv}_{i-j}	t^{p}_{i-j}	t^{cp}_{i-j}	σ^{2}_{i-j}
0	1	14	20	25	19.8	3.36
1	2	4	6	10	6.3	1
2	3	7	10	15	10.3	1.7
0	4	3	4	5	4	0.1
4	5	5	7	9	7	0.4
4	6	15	22	28	21.8	4.6
5	7	10	16	20	15.6	2.7
7	8	8	11	14	11	1
8	9	5	13	18	12.5	4.6
3	9	5	12	18	11.8	4.6

Рис. 7 Вихідні дані середнього часу та середньоквадратичної дисперсії

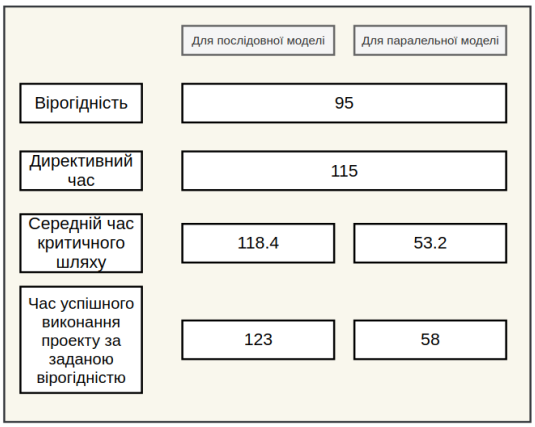


Рис. 8. Вихідні дані

4 Висновки

Розроблена модель мультипаралельної обробки інформації мережевого планування дозволяє майже повністю нівелювати негативний вплив людей на етапі мережевого планування, – особливо, при формуванні мережевого графу, тому, що автоматично побудована схема є більш точною.

Порівнюючи мережеве планування із застосуванням методів мультипаралельної обробки зі звичайним мережевим плануванням можна відмітити візуалізацію чіткої часової граф схеми, на якій відображено у який час буде закінчена та чи інша задача, орієнтуючись на працівників, що працюють паралельно.

СПИСОК ЛІТЕРАТУРИ

1. Гергель, В.П., Стронгін, Р.Г. Основи паралельних обчислень для багатопроцесорних обчислювальних систем. - Н.Новгород, ННГУ. 2003.
2. Воєводін В.В., Воєводін Вл.В. Паралельні обчислення. - СПб.: БХВ-Петербург.2002.
3. Немнюгін С. Моделі та засоби програмування для багатопроцесорних систем - СПб.: БХВ-Петербург. 2009.
4. Хьюз К., Хьюз Т. Паралельне і розподілене програмування на С++: Пер. з англ. - М.: Видавничий дім "Вільямс", 2004. - 672с.
5. Поляков Г.А., Шматков С.І., Толстолужська О.Г., Толстолужський Д.А. Синтез і аналіз паралельних процесів в адаптивних часопараметризованих обчислювальних системах. - Х.: ХНУ імені В.Н. Каразіна, 2012, -672с.

REFERENCES

1. Gergel, V.P., Strongin, R.G. Fundamentals of Parallel Computing for Multiprocessor Computing Systems. - Nizhny Novgorod, UNN. 2003. [In Ukrainian].
2. Voevodin V.V., Voevodin V.V. Parallel computing. - SPb .: BHV-Petersburg. 2002. [In Ukrainian].
3. Nemnyugin S. Models and programming tools for multiprocessor systems - SPb .: BHV-Petersburg. 2009. [In Ukrainian].
4. Hughes K., Hughes T. Parallel and distributed programming in C ++ .: Per. from English - M .: Publishing house "Williams", 2004. - 672s. [In Ukrainian].
5. Polyakov G.A., Shmatkov S.I., Tolstoluzhskaya E.G., Tolstoluzhsky D.A. Synthesis and analysis of parallel processes in adaptive time-parameterized computing systems. - Kh .: KhNU named after V.N. Karazin, 2012.-672s. [In Ukrainian].

Model of multiparallel information processing for network planning

Tolstoluzkyi Yevhen
PhD student;
Kharkiv National University of Radio Electronics,
14 Nauky Avenue, Kharkiv, Ukraine, 61166

The information technologies are playing an increasingly important role in the modern world. This leads to an increase in the number of IT projects. An IT project is a project that has clear deadlines and the goal of creating a unique and high-quality product within the set timeframe. IT projects consist of various technologies, computing and communication processes, information and human resources. To effectively manage such projects, the concept of project management has been formalized. Project management involves creating and adjusting plans, controlling and allocating resources and tasks, and creating a balance between project constraints. The longer the project, the more risks arise during its execution and implementation. These factors can affect the project's development time, profit, resources spent, as well as losses and costs in the event of unforeseen situations. For specialists working on IT projects, any unplanned issues and costs can present serious challenge. That's why the development of new automated project management solutions is a pressing issue. Such software models can help to minimize time and calculate possible risks. One of the stages that can be automated during the planning process is the construction of a visual model of work in the form of a network graph. This paper considers the possibility of automating the process of building a network graph using multiparallel information processing methods. This type of calculation can increase the gain in project execution time, establish a mechanism for parallel execution of tasks, and minimize possible risks.

Key words: *project, automation, project manager,* multiparallel information processing methods.

Надійшла у першій редакції 11.06.2023, в останній - 07.09.2023.
The first version has been received on 11.06.2023, the final version - on 07.09.2023.

DOI: https://doi.org/10.26565/2304-6201-2023-59-07

УДК (UDC) 004.67:004.8

Узлов Дмитро Юрійович
к.т.н., доцент закладу вищої освіти кафедри теоретичної та прикладної інформатики
Харківський національний університет імені В .Н. Каразіна, майдан Свободи, 4, Харків, Україна, 61022
e-mail: dmytro.uzlov@karazin.ua
https://orcid.org/0000-0003-3308-424X

Морозова Анастасія Геннадіївна
к.т.н., старший викладач закладу вищої освіти кафедри теоретичної та прикладної інформатики
Харківський національний університет імені В .Н. Каразіна, майдан Свободи, 4, Харків, Україна, 61022
e-mail: a.morozova@karazin.ua
https://orcid.org/0000-0003-2143-7992

Кузнєцова Вікторія Олександрівна
к.ф-м.н., доцент закладу вищої освіти кафедри вищої математики та інформатики
Харківський національний університет імені В .Н. Каразіна, майдан Свободи, 4, Харків, Україна, 61022
e-mail: vkuznietcova@karazin.ua
https://orcid.org/0000-0003-3882-1333

Руккас Кирило Маркович
д.т.н, доцент, професор закладу вищої освіти кафедри теоретичної та прикладної інформатики
Харківський національний університет імені В .Н. Каразіна, майдан Свободи, 4, Харків, Україна, 61022
e-mail: rukkas@karazin.ua
https://orcid.org/0000-0002-7614-0793

Використання нейронних мереж для масштабування табличних даних тренувальних dataset

У роботі запропоновано метод збільшення табличних даних тренувальних dataset за допомогою нейронних мереж, описано архітектуру таких мереж.
Актуальність. На даний час існує проблема недостатньої кількості вихідних даних для навчання моделей штучного інтелекту, що призводить до значної похибки моделювання. Робота присвячено розробці підходів до генерації штучних табличних даних, які можна використовувати надалі для моделей штучного інтелекту.
Мета. Метою роботи було проаналізувати методи та алгоритми для збільшення training dataset для табличних даних за допомогою нейронних мереж.
Методи дослідження. Основним методом дослідження є процес підбору параметрів алгоритму генерації штучних даних та вибір оптимальних параметрів архітектури нейронної мережі.
Результати. Використання нейронних мереж для масштабування табличних даних тренувальних dataset підтвердило працездатність запропонованого підходу. Результати налаштування алгоритму та вибір оптимальних параметрів нейронної мережі показали, що згенеровані штучні дані найбільше нагадують початкові по критеріям середнього значення, максимального, мінімального та залежності між даними.
Висновки. Вирішено задачу масштабування табличних даних тренувальних dataset за допомогою нейронних мереж. Такий підхід дозволяє значно спростити процес навчання нейронних мереж. Наукова новизна даної роботи полягає в розробці підходів і методів збільшення табличних даних з використанням штучного інтелекту та deep learning.
Ключові слова: *нейронні мережі, database, табличні дані, data augmentation, training dataset, штучний інтелект, deep learning.*

Як цитувати: Узлов Д. Ю., Морозова А. Г., Кузнєцова В. О., Руккас К. М. Використання нейронних мереж для масштабування табличних даних тренувальних dataset. *Вісник Харківського національного університету імені В. Н. Каразіна, серія Математичне моделювання. Інформаційні технології. Автоматизовані системи управління*. 2023. вип. 59. С.63-71. https://doi.org/10.26565/2304-6201-2023-59-07

How to quote: D. Uzlov, A. Morozova, V. Kuznietcova, K. Rukkas, “Scaling tabular data of training datasets with neural networks” *Bulletin of V. N. Karazin Kharkiv National University, series*

Mathematical modelling. Information technology. Automated control systems, vol. 59, pp.63-71, 2023. https://doi.org/10.26565/2304-6201-2023-59-07

1 Вступ

Data augmentation (збільшення даних) – це техніка штучного створення нових даних з наявних даних і значного збільшення різноманітності даних, доступних для навчання моделей. Це робиться шляхом застосування предметно-орієнтованих методів до підмножини навчальних даних. Оскільки продуктивність моделі сильно залежить від якості та кількості набору даних, використання синтетично згенерованих даних може певною мірою допомогти покращити продуктивність моделі [1]. Data augmentation техніки широко використовуються для графічних даних (Image Augmentation) [2], а також текстових даних (Text Data Augmentation) [3]. Для збільшення SQL даних або табличних даних не існує стандартних технік та методів. Сьогодні прийняття рішення на основі статистики є суттєвою для ряду задач, тому використання методів збільшення табличних даних та використання Neural Networks та Machine Learning для таких даних є актуальною задачею.

Використання штучно згенерованих даних вирішує такі проблеми як:

1. Необхідність великої кількості досліджень та збору даних (опитування користувачів сервісу, проведення тестових випробувань, тощо).
2. Аналіз зібраних даних.
3. Відкидання неправдивих даних (навмання заповнені бюлетені, помилки через технічні причини, тощо).
4. Оцифрування даних.

Техніка Data augmentation може бути використана в будь-якій компанії, яка застосовує у своїх дослідженнях штучний інтелект та табличні дані. Вона має зменшити витрати на збір даних тим самим пришвидшити впровадження нових змін в проекті, а також покращити якість навчання моделі.

Техніка Data augmentation може бути корисним в будь-якому проекті, що спеціалізується на роботі з табличними даними та штучним інтелектом, та особливо у випадках, коли є необхідність у великій кількості досліджень та збору даних, але їх за якоїсь причини важко зібрати у необхідній кількості.

Метод масштабування табличних даних повинен генерувати нові дані базуючись на вхідному dataset. Він може працювати з невеликою кількістю вхідних рядків та генерувати задану кількість нових. Нові рядки базуються здебільшого на середньому значенні вхідних рядків та копіюють їх тип розподілу відносно інших стовпчиків таблиці. Тим самим згенеровані дані «візуально» здаються схожими на ті, на яких проводилось тренування моделі.

2 Постановка задачі

Штучний інтелект набуває все більшої популярності останнім часом та прогнозується збільшення популярності у майбутньому. Через це питання збільшення training dataset для табличних даних є актуальним у сьогодення, основним завданням якого є аналіз вхідного dataset та генерація нових, схожих табличних даних. Для досягнення поставленої мети, були сформульовані наступні задачі:

1. Проаналізувати наявні бібліотеки для збільшення табличних даних.
2. Проаналізувати методи роботи бібліотеки з даними та за можливості покращити якість роботи алгоритму з ними, підібравши найкращі параметри для роботи алгоритму. А саме:
    a. скалери;
    b. алгоритми оптимізації для моделі глибокого навчання;
    c. топологія нейронної мережі.
3. Вдосконалити роботу бібліотеки для роботи з типами даних string та int.
4. Протестувати роботу бібліотеки на dataset з різними видами розподілу. У якості dataset використовувати реальні дані, а не згенеровані.

Вибір бібліотеки для збільшення табличних даних

В роботі розглянуто можливість використання різних бібліотеки для генерування нових табличних даних із вхідного dataset. Критерієм вибору бібліотеки була задача максимально

зберегти початкові характеристики даних, а саме математичне очікування, дисперсію та залежність між стовпцями.

Бібліотека deep_tabular_augmentation надає абсолютну свободу користувачу щодо налаштування вхідних параметрів, що використовуються для генерування нових даних [9]. Бібліотека дозволяє самостійно обирати, досліджувати та змінювати вхідні параметри для отримання бажаного результату.

Бібліотека RandomForestClassifier з sklearn дозволяє вказувати тільки вхідний dataset без можливості самостійно впливати на генерацію даних [10].

Пакет ydata_synthetic надає значно більше свободи користувачу у порівнянні з RandomForestClassifier [11]. Він надає більше можливостей впливати на зміну даних, але все ж таки менше ніж при використанні deep_tabular_augmentation.

Для генерації табличних даних тренувальних dataset було обрано бібліотеку deep_tabular_augmentation.

3 Алгорит збільшення табличних даних

Налаштування бібліотеки deep_tabular_augmentation для масштабування табличних даних тренувальних dataset складається з декількох етапів:

1. Підготовка та масштабування ознак.
2. Розбиття даних.
3. Визначення топології нейронної мережі
4. Вибір оптимізатора
5. Визначення кількості епох для навчання моделі

Масштабування ознак.

Перший етап – це Масштабування ознак (або Нормалізація даних). Перш за все dataset необхідно підготувати для роботи з ним. Для цього використовують масштабування ознак. Так як значення у даних можуть сильно різнитися між собою та мати різні діапазони, модель може давати хибні результати. Тому дані потрібно нормалізувати. Для цього використовують різні скалери з sklearn. В залежності від типу розподілу даних необхідно обрати відповідний скалер. Наприклад, MinMaxScaler та StandardScaler гарно працюють з числовими даними. Водночас StandardScaler використовують для нормального розподілу, MinMaxScaler за відсутності нормального розподілу та коли варто вказати на чітку відстань між значеннями. Для більшості dataset якості роботи StandardScaler достатньо. Варто також зазначити, що деякі скалери, наприклад Normalizer неможливо використовувати з бібліотекою deep_tabular_augmentation через те, що розробник не впровадив необхідні для цього зміни в свій модуль. Порівняння скалерів наведено у Таблиці 1.

Таблиця 1. Порівняння скалерів

Скалер	Стійкий до викидів	З чіткою межею даних	Межа невідома
StandardScaler	-	-	+
MinMaxScaler	-	+	-
MaxAbsScaler	+	+	-
RobustScaler	+	-	+
PowerTransformer	+-	+-	+

Розбиття даних.

Наступним етап є Розбиття даних. Дані розбиваються на дві частини: для тренування та валідації моделі. Попередньо рядки перемішують між собою. Таким чином дані для валідації використовуються для того, щоб зрозуміти наскільки навчена модель, а також це допомагає виявити проблеми Underfitting та Overfitting. Зазвичай для розбиття даних використовується train_test_split() з бібліотеки sklearn.model_selection, але для більш специфічних завдань можна звернути увагу на split_df() з mlprepare. Низькі значення функції втрат можуть вказувати на те, що модель гарно навчилась генерувати нові дані, або на те, що вона перенавчена. Було протестовано наступні варіанти розбиття даних валідацію: 5%, 8%, 10%, 12% та 15%.

В більшості випадків найнижчі значення функції втрат, при яких зберігається залежність між даними, та модель не перенавчається, досягаються при виділенні 10% даних на валідацію.

Топологія нейронної мережі.

Після масштабування ознак необхідно вказати, за допомогою якої топології нейронної мережі будуть опрацьовуватися дані. Ця топологія вказує на зв'язки між вузлами (нейронами) в мережі. Для вирішення задачі найбільш цікавими є наступні три типи мережі, топології яких наведено на рисунку 3.1.:

- Auto Encoder (AE)
- Variational AE (VAE)
- Sparse AE (SAE)

Загалом вони використовуються для класифікації та кластеризації ознак. VAE на відміну від AE приділяє більше уваги на зв'язок між даними у той час, коли AE намагається їх узагальнити. SAE схожий на VAE, але також здатний знаходити приховані шаблони групування даних. На практиці це виявляється у тому, що SAE виділяє значно більше даних, що знаходяться значно далі від основного скупчення. Наприклад, якщо взяти нормальний розподіл, то SAE буде також виділяти точки, що знаходяться біля 0, у порівнянні з AE та VAE, котрі виділяють лише дані близькі до середнього значення. Тож було вирішено зупинитися на SAE тому, що для задачі масштабування табличних даних важливо вказати всі дані, а не лише близькі до середніх значень.

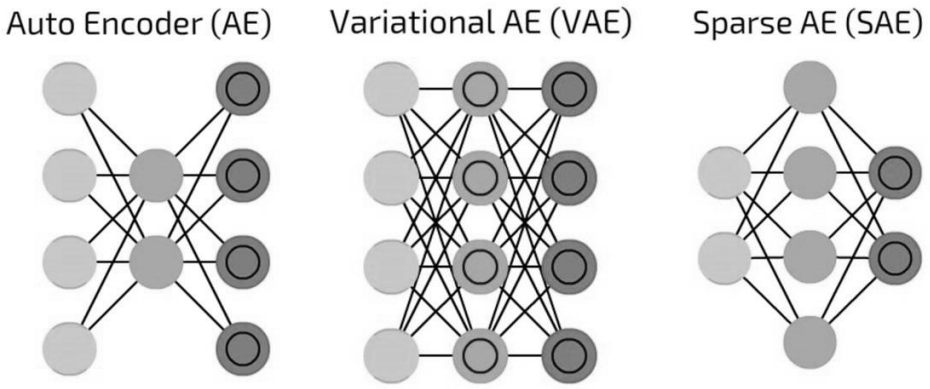


Рис.3.1. Топології нейронної мережі

Найближчі до початкових даних результати роботи алгоритму було отримано, коли слої топології виглядають наступним чином: 6, 20, 6. Тобто 6 нейронів у вхідному і вихідному слої та 20 у прихованому. При цьому, ці зміни майже не впливають на зміну середнього значення чи дисперсії даних, а змінюється здебільшого лише залежність даних між собою та вірогідність отримати мінімальних та максимальних значень одночасно в декількох стовпчиках таблиці.

Також кількість нейронів має сенс змінювати в залежності від вхідних даних. Емпірично було виявлено, що збільшення 1 та 3 слою підвищує «влучність» алгоритму, згенеровані дані стають ближчими до середніх значень розподілу. Збільшення або зменшення прихованого (другого) слою відповідно збільшує або зменшення кількість даних, що знаходяться на значному віддаленні від середнього значення. А ось зменшення першого та третього слою майже не має практичного сенсу, у більшості випадків на різних даних алгоритм все частіше дає похибки та помилково визначає залежність між розподілом даних. Наприклад, коли розподіл даних схожий на графік функції $y = a^x$ алгоритм може надати $y = \frac{1}{x}$.

Оптимізатор навчання нейронних мереж.

Наступний етап підготовки до data augmentation – це обрання алгоритму оптимізації навчання нейронної мережі, який використовується для налаштування ваги нейронної мережі в процесі її навчання. Він визначає, які значення ваги потрібно використовувати для мінімізації функції втрат, яка вимірює помилку передбачення моделі на навчальному наборі даних. Функція втрат – це функція, яка характеризує втрати при неправильному прийнятті рішень на основі спостережених даних. Тобто це метод оцінки того, наскільки добре алгоритм моделює вказаний набір даних, наскільки гарно алгоритм працює з заданим набором. [6]

Найкраще у дослідженнях себе проявили оптимізатори Adam (Адаптивне оцінювання моментів) та RMSProp (Пропагація кореня середньоквадратичного значення), та зовсім погано SGD (Стохастичний градієнтний спуск), незважаючи на його високу популярність. Хоча й Adam має більшу обчислювальну складність, ніж RMSprop, через необхідність обчислення додаткових моментів градієнта, але дані, що генеруються з його допомогою, у більшості випадках більш схожі на початкові у порівнянні з RMSProp. А саме min, max значення ближче до вхідних даних, залежність між даними більш схожа на залежність вхідних даних та менша ймовірність помилково вказати хибну залежність між даними.

Кількість епох при навчанні моделі.

Визначення цього параметру значною мірою залежить від самого датасету. У всіх розглянутих випадках найкращі результати отримувались при значенні 100-300 епох. При збільшенні цього значення модель починає перенавчатися та згенеровані нею дані все більше дублюють середнє значення початкового dataset, при зменшенні зростає ймовірність похибки алгоритму та неправильно виявленої залежності між даними.

Вдосконалення алгоритму при роботі з типами даних string та integer.

Кластеризація, або кластерний аналіз — це статистична процедура, задача якої полягає в розбитті вибірки об'єктів на підмножини, що не перетинаються і називаються кластерами [7]. Отже типовою задачею кластеризації є розбиття даних на основі їх подібності. Бібліотека deep_tabular_augmentation не вміє працювати зі строковими типами даних, хоча якщо строкові дані розбити на невелику кількість кластерів, то виходить, що генерувати нові дані спираючись на порядковий номер кластеру має сенс. Звісно, що такими діями не вийде згенерувати новий текст, а можна лише використовувати старий, а також такі дії будуть мати сенс лише у випадках, коли кількість даних більша за кількість кластерів. Тож було вирішено додати до алгоритму можливість генерувати дані зі строковим типом.

Також deep_tabular_augmentation завжди генерує дані з плаваючою точкою, навіть якщо вхідні дані мають цілий тип (integer). Цю особливість також було виправлено.

Види розподілу початкових даних та результати роботи алгоритму

Для тестування методів бібліотеки deep_tabular_augmentation та data augmentation були використані різні dataset, отримані із реальних даних на сайті kaggle.com. Нижче наведено результати роботи методів бібліотеки для різних видів розподілів.

Приклад 1. Нормальний розподіл.

На рисунку 3.2. зображені вхідні дані, а на рисунку 3.3. – згенеровані.

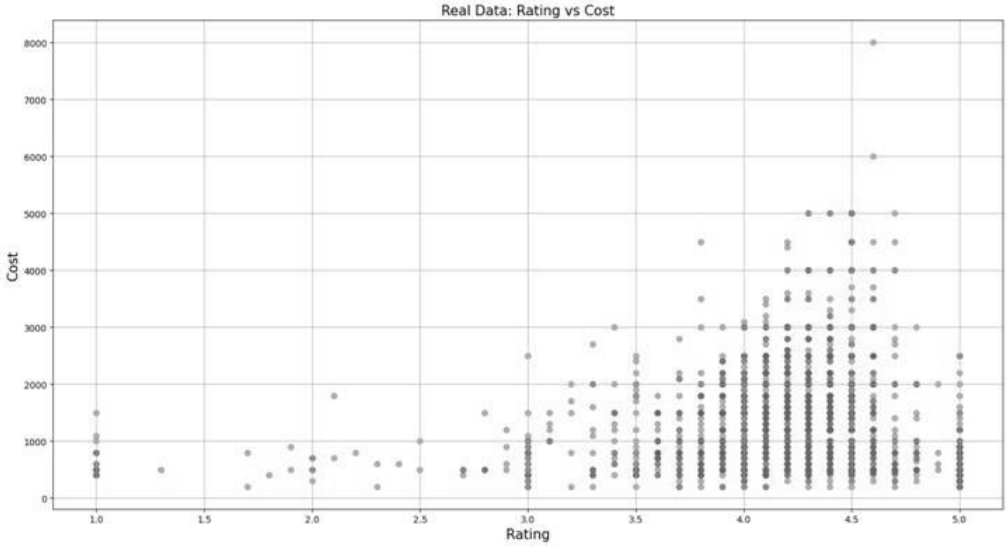


Рис.3.2. Вхідні дані

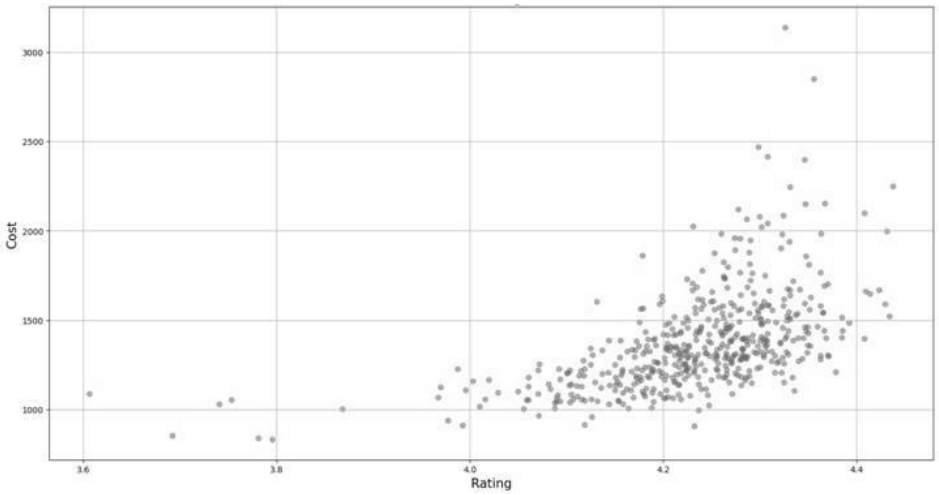


Рис.3.3. Згенеровані дані

На рисунках видно, що зберігається загальна форма розподілу та середнє значення. Варто зазначити, що у вхідних даних наявний шаг по осі X (Rating), тобто дані розташовані на певному віддалені один від одного, при цьому цей шаг не зберігається у згенерованих даних. Дані були згенеровані з використанням StandardScaler. Також, було застосовано інший вид скайлера, а саме MinMaxScaler, який застосовують для вказання чіткої відстані між даними. Але його застосування у даному випадку не виправдало себе, бо відстань він вказував недоречно. Тому було вирішено вказувати шаг між даними вже після генерації, оброблюючи дані, а не з застосуванням скалеру.

Приклад 2. Лінійний розподіл.

Наступний тип розподілу – лінійний. На рисунку 3.4. зображені вхідні дані, які один відносно одного розташовані по прямій лінії. На рисунку 3.5. – результат роботи алгоритму. Видно, як алгоритм доповнив пустоти. Також між даними у вхідному dataset зберігається математична залежність: по осі X – кількість років, по осі Y – кількість місяців, тому Y завжди у 12 разів більше за X, але у згенерованих даних ця математична точність втрачається.

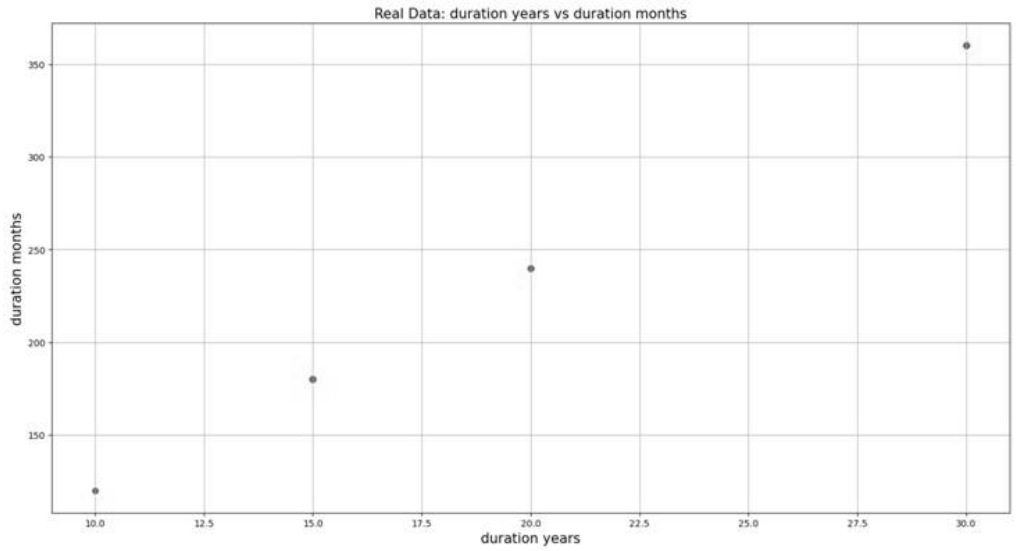


Рис.3.4. Вхідні дані

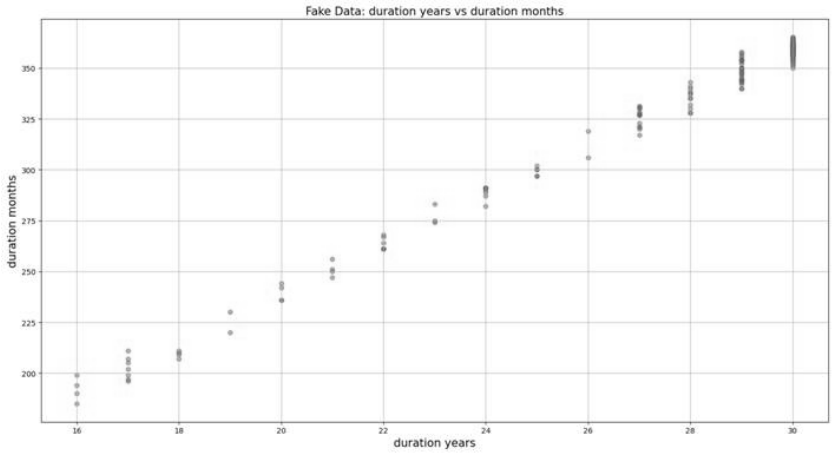


Рис.3.5. Згенеровані дані

Приклад 3.

Ще один приклад розподілу наведено на рисунку 3.6. та відповідні згенеровані дані на рисунку 3.7.

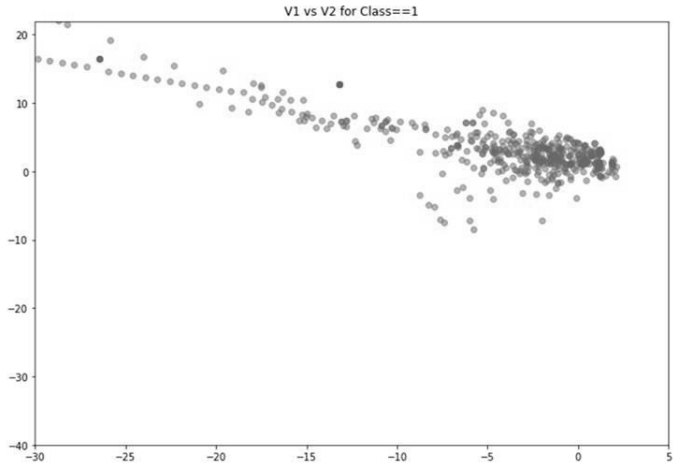


Рис.3.6. Вхідні дані

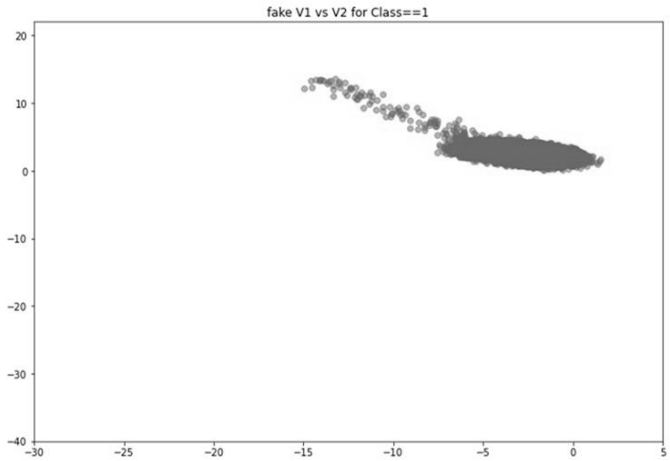


Рис.3.7. Згенеровані дані

Недоліки бібліотеки deep_tabular_augmentation.

Найбільш суттєвим недоліком є майже повна відсутність дисперсії у згенерованих даних. Дані дійсно мають схоже середнє значення, але наприклад, коли вхідні дані мають нормальний

розподіл, згенеровані візуально більше схожі на криву лінію, чи в деяких випадках дві криві лінії, що перетинаються. Приклад вхідних та згенерованих даних наведено на рисунку 3.8.

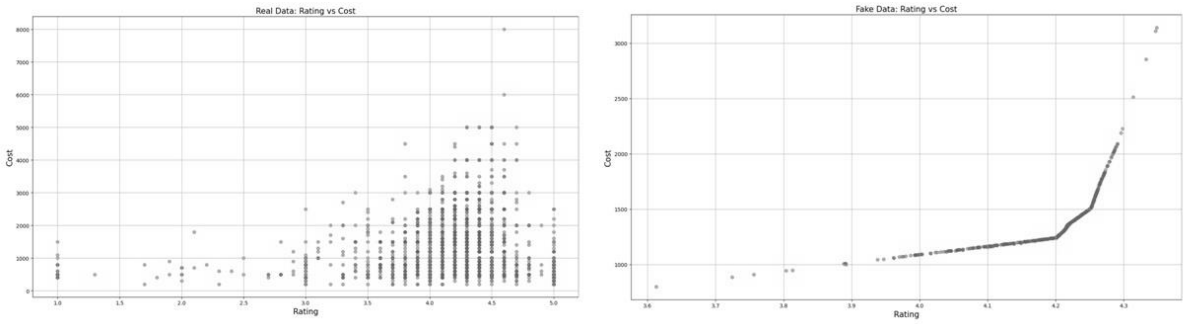


Рис.3.8.Ліворуч – вхідні дані, праворуч – згенеровані дані

Для вирішення цієї проблеми розробник радить трохи зміщувати дані, на відстань рівну 10% від дисперсії. У результаті змішування можна отримати приблизно наступний графік, наведений на рисунку 3.9.

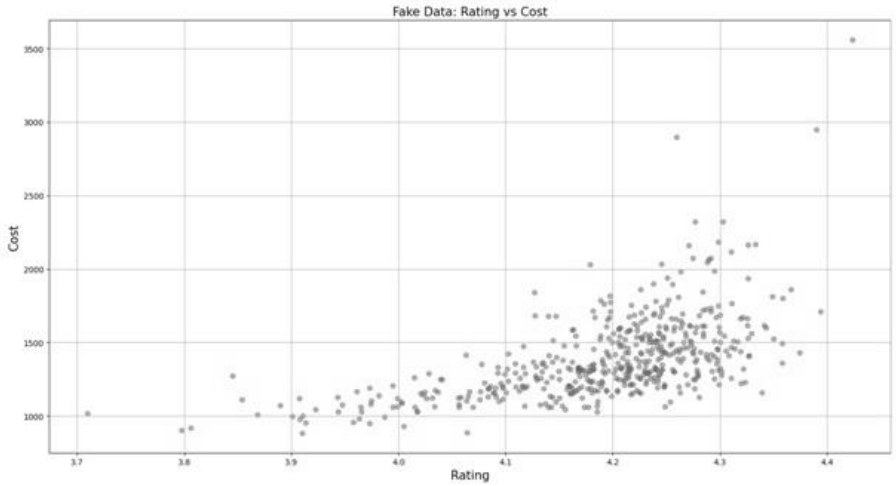


Рис.3.9. Згенеровані дані зі зміщенням 10% від дисперсії

Звісно, що в результаті ми ніколи не матимемо однакові дисперсії, а в спробі змістить дані на відстань рівну дисперсії ми можемо отримати у чисельному вигляді майже однакові середні значення та дисперсію, у порівнянні з початковими даними, але візуально це вже зовсім не схоже на нормальний розподіл, тобто залежність між даними втрачається.

4 Висновки

В роботі було проаналізовано методи та алгоритми для збільшення training dataset для табличних даних. Для досягнення цієї мети біло обрано бібліотеку deep_tabular_augmentation та проаналізував функції, що в ній використовуються. Під час аналізу було підібрано діапазони значень вхідних параметрів, при яких досягається найвища точність роботи алгоритму та згенеровані дані найбільше нагадують початкові по критеріям середнього значення, максимального, мінімального та спостерігається залежність між даними. Були підібрані наступні параметри:

- функція втрат;
- скалери;
- топологія нейронної мережі;
- оптимізатор навчання нейронних мереж;
- оптимальна кількість епох.

Також були помічені та проаналізовані недоліки модулю та покращена якість роботи алгоритму при роботі з типами даних string та integer.

СПИСОК ЛІТЕРАТУРИ

1. Abinaya Mahendiran, Vedanth Subramaniam. Data Augmentation Techniques for Tabular Data. Mphasis. https://www.mphasis.com/content/dam/mphasis-com/global/en/home/innovation/next-lab/Mphasis_Data-Augmentation-for-Tabular-Data_Whitepaper.pdf
2. Luis Perez, Jason Wang. The Effectiveness of Data Augmentation in Image Classification using Deep Learning. arXiv:1712.04621, 2017. https://arxiv.org/pdf/1712.04621
3. Shorten, C., Khoshgoftaar, T.M. & Furht, B. Text Data Augmentation for Deep Learning. J Big Data 8, 101 (2021). https://doi.org/10.1186/s40537-021-00492-0
4. Qizhe Xie, Zihang Dai, Eduard Hovy, Minh-Thang Luong, Quoc V. Le. Unsupervised Data Augmentation for Consistency Training. arXiv:1904.12848v6, 2020. https://arxiv.org/pdf/1904.12848v6
5. E. Jannik Bjerrum. SMILES Enumeration as Data Augmentation for Neural Network Modeling of Molecules. ArXive-prints, Mar. 2017
6. Alhassan Mumuni, Fuseini Mumuni. Data augmentation: A comprehensive survey of modern approaches. https://doi.org/10.1016/j.array.2022.100258
7. Ioffe S., Szegedy C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. International conference on machine learning, PMLR (2015), pp. 448-456
8. Agnieszka Mikolajczyk, Michal Grochowski. Data augmentation for improving deep learning in image classification problem. 2018 International Interdisciplinary PhD Workshop (IIPhDW). DOI:10.1109/IIPHDW.2018.8388338
9. https://github.com/lschmiddey/deep_tabular_augmentation
10. https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html
11. https://docs.synthetic.ydata.ai/1.4

Uzlov Dmytro	*Doctor of Philosophy, Associate professor of theoretical and applied computer science department, V. N. Karazin Kharkiv National University, Svobody Sq., 4, Kharkiv, Ukraine, 61022*
Morozova Anastasiia	*Doctor of Philosophy, Senior lecturer of theoretical and applied computer science department, V. N. Karazin Kharkiv National University, Svobody Sq., 4, Kharkiv, Ukraine, 61022*
Kuznietcova Victoriya	*Doctor of Philosophy, Associate professor of higher mathematics and computer sciences department, V. N. Karazin Kharkiv National University, Svobody Sq., 4, Kharkiv, Ukraine, 61022*
Rukkas Kyrylo	*Doctor of Technical Sciences, Associate professor, Professor of theoretical and applied computer science department, V. N. Karazin Kharkiv National University, Svobody Sq., 4, Kharkiv, Ukraine, 61022*

Scaling tabular data of training datasets with neural networks

The paper proposes a method of scaling the tabular data of the training dataset using neural networks, describes the architecture of such networks.

Relevance. Presently, there is a problem of insufficient amount of raw data for training artificial intelligence models, which leads to significant modeling error. The work is devoted to the development of approaches to the generation of artificial tabular data, which can be used in the future for artificial intelligence models.

Goal. The purpose of the work was to analyze methods and algorithms for scaling the training dataset for tabular data using neural networks.

Research methods. The main research method is the process of selecting the parameters of the artificial data generation algorithm and choosing the optimal parameters of the neural network architecture.

The results. Using neural networks for scaling the tabular data of the training dataset confirmed the efficiency of the proposed approach. The results of the algorithm adjustment and the selection of the optimal parameters of the neural network showed that the generated artificial data most resemble the initial ones in terms of the criteria of average value, maximum, minimum and dependence between data.

Conclusions. The task of scaling the tabular data of the training dataset using neural networks has been solved. This approach makes it possible to significantly simplify the process of learning neural networks. The scientific novelty of this work lies in the development of approaches and methods for increasing tabular data using artificial intelligence and deep learning.

Keywords: *neural networks, database, tabular data, data augmentation, training dataset, artificial intelligence, deep learning.*

Надійшла у першій редакції 09.07.2023, в останній - 17.08.2023.
The first version has been received on 09.07.2023, the final version - on 17.08.2023.