ISSN 0005-1179 (print), ISSN 1608-3032 (online), Automation and Remote Control, 2023, Vol. 84, No. 7, pp. 823–830.

Machine Learning for Diagnosis of Diseases with Complete Gene Expression Profile

A. M. Mikhailov[*,a], **M. F. Karavai**[*,b], **V. A. Sivtsov**[*,c], **and M. A. Kurnikova**[**,d]

[*] *Trapeznikov Institute of Control Sciences, Russian Academy of Sciences, Moscow, Russia*
[**] *Dmitry Rogachev National Medical Research Center of Pediatric Hematology, Oncology and Immunology, Moscow, Russia*
e-mail: [a] *alxmikh@gmail.com,* [b] *mkaravay@yandex.ru,* [c] *TheDeGe@yandex.ru,* [d] *mish2109@yandex.ru*

Received July 19, 2022
Revised March 21, 2023
Accepted March 30, 2023

Abstract—This paper considers the use of machine learning for diagnosis of diseases that is based on the analysis of a complete gene expression profile. This distinguishes our study from other approaches that require a preliminary step of finding a limited number of relevant genes (tens or hundreds of genes). We conducted experiments with complete genetic expression profiles (20 531 genes) that we obtained after processing transcriptomes of 801 patients with known oncologic diagnoses (oncology of the lung, kidneys, breast, prostate, and colon). Using the indextron (instant learning index system) for a new purpose, i.e., for complete expression profile processing, provided diagnostic accuracy that is 99.75% in agreement with the results of histological verification.

Keywords: pattern recognition, machine learning, inverse patterns, gene expression profiles, diagnosis of diseases

DOI: 10.25728/arcRAS.2023.92.25.001

1. INTRODUCTION

The common denominator of existing approaches to the diagnosis of oncologic diseases based on gene expression profiles is the use of a limited set of genes; see, for example, methodological and review papers [1–6]. With this approach, it is also necessary to perform a preliminary analysis and identify individual genes or combinations of genes whose expression activity is most characteristic in the case of specific diseases. However, the diagnostic accuracy achieved does not exceed 95% on sets that include up to 90 genes [4], which is presumably due to the limited number of genes used. For example, in [5], this number is reduced to just 10 genes. Thus, the data analysis process consists of two stages. For example, in [6], in the first stage, researchers use the principal component analysis, selecting 103 genes. They carry out the final diagnosis in the second stage, where a Bayesian neural network provided 93.66% accuracy, a deep neural network provided 93.41%, and logistic regression — 92.82%. Furthermore, the third stage can take place. It would reject questionable results based on a decision threshold that allows to almost completely eliminate wrong diagnoses [6] at the cost of not diagnosing some cases, which is considered to be more acceptable than a wrong diagnosis.

This study examines the use of machine learning for disease diagnosis based on the analysis of the complete gene expression profile, i.e., the activity of all 20 531 known genes, which simplifies the process by eliminating the need for preselection of relevant genes. This paper discusses the details of the operation of the machine learning system used.

Fig. 1. HiSeq 2500 system.

In biology, the expression level estimates the transcriptional activity of a gene as the amount of messenger RNA (mRNA) it produces. Obtaining gene expression profiles is a minimally invasive or completely non-invasive procedure, such as saliva sampling, which determines the comfort of such a procedure for the patient. The quality of diagnosis depends on the completeness of the profile, i.e., the number of genes considered. However, the use of complete gene expression profiles is more of a future of medical diagnosis, as currently there is a lack of inexpensive equipment for mass application to obtain such profiles. Currently, we can obtain profiles using the following technologies, among others [7]:

— PCR tests;

— DNA microarrays;

— sequencing.

If the analysis involves the expression level of a relatively small number of genes, we can use the available and relatively inexpensive real-time quantitative PCR (qPCR) method. The second technology, DNA microarrays, provides a more accurate assessment of gene expression. However, when it comes to the expression profile of all genes, sequencing is currently required. One of the most popular platforms for high-throughput sequencing is Illumina equipment [8]. The HiSeq 2500 system utilizes next-generation SBS technology, which supports massively parallel sequencing using fluorescently labeled nucleotides, allowing the reading of individual bases as they are incorporated into growing DNA strands. Figure 1 shows an example of equipment that utilizes this method.

Illumina HiSeq performs two functions, such as genome sequencing and determination of expression level. The latter is achieved by reading not DNA but mRNA code. The expression level is inferred from the number of mRNA copies. Considering the decreasing cost of RNA sequencing, it is possible in the future to create simpler and less expensive equipment for installation in widely spread laboratories such as INVITRO. With the widespread availability of such equipment, prospective diagnostics could be discussed as the use of the complete expression profile automatically takes into account all possible **active** gene combinations, which are difficult to predict and that influence the onset of various diseases. Additionally, gene expression profiles serve as valuable materials not only in diagnostics but also in various scientific and clinical research.

The second component of the considered methodology is machine learning, which allows automatic learning from complete biological profiles without analyzing genetic combinations for subsequent classification and diagnosis. Artificial neural networks [9] are traditionally used for machine learning and have become a popular method applied to solve a variety of prediction and pattern recognition problems. However, training such networks takes a long time and requires significant computational resources, as it involves the calculation of a large number of coefficients to adapt the multilayer network architecture to a specific task. Nevertheless, we can significantly accelerate training and make it borderline instantaneous if we use an image indexing system [10, 11], similar to search engines like Google [12], where incremental learning is reduced to indexing new documents. In 1998, the term indextron was introduced [13] — a term used only as the name of the image indexing device, but not the name of the image indexing method. The difference between indextron and search engines lies in the fact that it inverts numerical data instead of textual documents. Section 3 discusses the specifics of inverting numerical images.

It should be noted that the approach used in [10] has similarities with the later proposed TF-IDF method [14], developed for document retrieval by computing the inverse document word frequencies. The TF-IDF method is utilized in search engines, where document name frequencies are calculated in fully inverted files. However, textual search engines do not work with numerical data as the document frequency of a word can change depending on the current set of documents, and the keyword itself might completely disappear due to even slight noise.

It should be mentioned that the indexing approach to pattern recognition is hardly employed in machine learning systems, where most methods utilize iterative learning, gradient descent, and a significant number of adaptive coefficients, which ultimately leads to slow learning. At the same time, the human brain can memorize new visual patterns at a glance. This study employ the method of numerical data indexing instead of iterative learning to achieve almost instantaneous learning when dealing with large volumes of data. Previous work on indexing non-textual numerical data can be found in [15], where the authors use such fast methods for image recognition in movies. However, the study [15] reduces the recognition of noisy numerical patterns representing images to converting the numerical patterns to textual form, followed by the use of existing methods for text information retrieval. The approach we use in this paper is applying the index recognition method [10, 11], which enables the indexing of noisy numerical patterns without intermediate conversion into textual form. It is also worth noting that the objective of this article is not to develop a new machine learning method, but to utilize the indexing recognition method for a new purpose, namely, for nearly instantaneous learning when working with large databases of biological data.

2. SOURCE DATA

In the experiment, we used a dataset called Gene Expression Cancer RNA-Seq [16] to compare the effectiveness of training and classifying data using a neural network and an indextron. This data set consists of 801 rows, each containing 20 531 floating point numbers (see Table 1). Each number with profile/gene coordinates represents the level of activity of the corresponding gene, measured in arbitrary units ranging from 0 (no activity or absence of the gene) to a maximum activity of 15. The numbers in each nth row represent the activity levels of the corresponding genes of the nth patient. The diagnoses of all patients in the dataset [15] are known and were determined using other clinical methods. In the experiment, we used the 401 odd rows for training, during which we provided the system with the diagnosis associated with the corresponding row. During testing, we used the 400 even rows, which were classified by assigning them to one of the five classes, and the class found was compared with the known diagnosis of the corresponding patient.

Table 1. Gene activity of patients

Profile	Gene 0	Gene 1	Gene 2	…	Gene 20 529	Gene 20 530	Class (diagnosis)
0	0.0	2.017	3.266	…	5.287	0.0	3
1	0.0	0.592	1.588	…	2.094	0.0	1
…	…	…	…	…	…	…	…
800	0.0	2.325	3.806	…	4.551	0.08	5

3. RECOGNITION PROBLEM STATEMENT AND SOLUTION METHOD

In this study, we use the index recognition method, proposed and considered in [10] and improved in [11] and other works, for a new purpose, namely diagnosing diseases by classifying gene expression profiles. We discuss the specifics of this method application in this problem below.

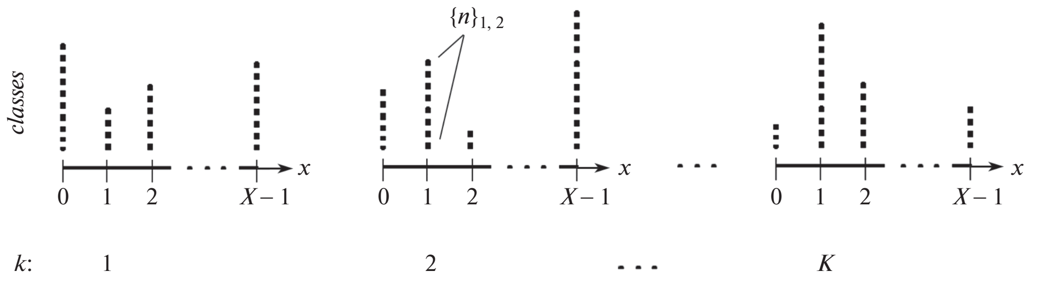

Fig. 2. Inverse patterns shown as K column groups.

Let all variables be integers, and let there be N patterns, where each pattern is represented by a K-dimensional feature vector

$$\mathbf{x_n} = (x_{n,1}, \ldots, x_{n,k}, \ldots, x_{n,K}), \quad n = 1, \ldots, N. \tag{1}$$

Here, Chebyshev distance between any two patterns $\mathbf{x_p}$ and $\mathbf{x_q}$ $(p, q \leqslant N)$ is greater than a certain predetermined number R, i.e. $|\mathbf{x_p} - \mathbf{x_q}| > R$, and variable $x_{n,k}$ $(0 \leqslant x_{n,k} < X)$ is a value of kth feature of the vector that represents nth pattern. Inequality $|\mathbf{x_p} - \mathbf{x_q}| > R$ means that for vectors $\mathbf{x_p}$, $\mathbf{x_q}$ there exists at least one dimension k such that $|x_{p,k} - x_{q,k}| > R$. In this case, every nth vector represents the nth class, which includes all vectors $\mathbf{x}$ such that $|\mathbf{x} - \mathbf{x_n}| \leqslant R$. Value R determines the size of the class and is called a radius of generalization.

Classification problem. For given vector $\mathbf{x} = (x_1, \ldots, x_k)$, find class n such that $|\mathbf{x} - \mathbf{x_n}| \leqslant R$. If such a class does not exist then add a new vector $\mathbf{x_{N+1}} = \mathbf{x}$ to the list (1) and increase the number of classes by 1.

Obviously, we can solve this problem by comparing the given vector $\mathbf{x}$ with vectors that represent classes, i.e., by exhaustive enumeration of classes. However, solving the classification problem with the inverse pattern method allows a significantly accelerated search, especially in the case of a large number of classes.

Consider solving the classification problem with inverse patterns. When using this method an unknown pattern $\mathbf{x}$ belongs to class m such that

$$m : H_R(m|\mathbf{x}) = \max_{n=1}^{N} H_R(n|\mathbf{x}).$$

Here, $H_R(m|\mathbf{x})$ is a histogram of class names contained in inverse patterns of features of vector $\mathbf{x}$. Thus, $H_R(m|\mathbf{x})$ is a conditional histogram of classes, as it depends on vector $\mathbf{x}$.

Note that if radius of generalization R is equal to zero, then the number of classes would be equal to number N of vectors in the list (1). If the number of actual externally determined classes is less than N then we use table function $class(n)$, $n = 1, \ldots, N$ that settles the mapping between the classes in the list (1) and external classes which are always known during supervised learning. Studies [10, 11] determine inverse patterns of features such as certain sets of class names. We index such sets with two-dimensional indices. Each two-dimensional index is determined by a pair of numbers (x, k), where x is a feature value, and k is a measurement number. The set of names of $\{n\}$ classes in the left part of equality (2) is an inverse pattern of feature x in measurement k.

$$\{n\}_{x,k} = \{n : x_{n,k} = x\}, \quad k = 1, \ldots, K, \quad x = 0, 1, \ldots, X-1. \tag{2}$$

Figure 2 graphically illustrates the concept of inverse patterns. There, they are represented by columns of points, heights of which equal energies of the corresponding inverse patterns. As mentioned above, the elements of columns are class names. Due to the fact that inverse sets have two-dimensional indices, we split all columns into K groups where each group k can contain up to X columns and each column up to N classes. Here, X is a feature value range. What remains to be done to solve the classification problem with inverse patterns is to find the class histogram $H_R(n|\mathbf{x})$. Let the input pattern be presented by vector $\mathbf{x} = (x_1, \dots, x_k)$. Then we can find the histogram of classes contained in inverse patterns using the following algorithm:

$$\forall n \in \{n\}_{x(k)+r,k}, \qquad H_R(n|\mathbf{x}) = H_R(n|\mathbf{x}) + 1. \tag{3}$$

Here $r = -R, \dots, -1, 0, 1, \dots, R,\ k = 1, \dots, K,\ x(k+r), k = x_{k+r,k}$. Therefore, the classification criterion presented above is a class histogram, the maximum position of which we have to find. To understand the algorithm we need to use the concept of "inverse patterns." Expression (2) defines it and is equipped with the necessary commentaries. Inverse patterns that contain sets of pattern classes act as input information for loop (3). This loop histograms classes and yields the required histogram as its output. The position of its maximum corresponds to the desired class. In terms of programming, this is a loop on macrocolumns r, measurements k, and classes n that are contained in inverse patterns.

During the indextron training, if the histogram maximum for input $\mathbf{x}$ is lower than a certain chosen threshold then the number of classes is increased by 1, $N = N + 1$, and this new number gets stored in corresponding inverse patterns

$$\{n\}_{x(k),k} = \{n\}_{x(k),k} \bigcup N, \qquad k = 1, \dots, K.$$

The initial condition is always $N = 0$. However, we do not calculate the class histogram (3) if the training uses a zero-valued radius of generalization $R = 0$. At the same time, every iteration the number of classes increases by 1. The study [17] presents a graphical illustration of this algorithm.

During recognition, if the histogram maximum for input $\mathbf{x}$ is lower than a certain threshold, then the input patterns remain unrecognized.

Note that the histogram maximum equals the dimension K of pattern $\mathbf{x}$. It is possible to show that it is derived from the following properties of inverse patterns: the columns of each group

- contain strictly N different class names: $\sum\limits_{x=0}^{X-1} |\{n\}_{x,k}| = N$,
- do not intersect: $\{n\}_{x,k} \cap \{n\}_{y,k} = \varnothing$.

During learning and recognizing a pattern, we use the value of its features x_k, $k = 1, \dots, K$ as an address of a column located in the kth dimension. At the same time, we coerce the real feature values x into an integer range of [0–255]:

$$x = 255(x - x_{\min})/(x_{\max} - x_{\min}).$$

Experience suggests that such continuous value sampling usually does not reduce the accuracy of classification. Table 2 presents an example of normalization of Table 1 data.

Therefore, the features of each received K-dimensional pattern isolate K columns — one column in each group. The issue is that, usually, for numerical features, the column intersection appears

Table 2. Normalized data

Profile	Gene 0	Gene 1	Gene 2	⋯	Gene 20 529	Gene 20 530	Class (diagnosis)
0	0	82	137	…	112	0	3
1	0	24	66	…	36	0	1
…	…	…	…	…	…	…	…
800	0	95	160	…	95	0	5

to be empty due to measurement errors and pattern deformations alter the column addresses. For this reason, we introduce macrocolumns, i.e., along with the column with address x we consider its neighbors with addresses $x \in [-R, R]$. The possibility of using macrocolumns is the result of inverse pattern properties mentioned above. As noted previously, we find the column intersections by calculating a histogram that determines the frequencies of class appearances. At the same time, the input pattern belongs to the most frequently appearing class.

4. SOFTWARE AND HARDWARE IMPLEMENTATION OF INDEXTRON

We implemented the parallel version of the algorithm in Python and ran it on the Nvidia GeForce GTX 1660 Super GPU. This graphics card has 44 multiprocessors and 1408 cores, allowing for parallelization of one process into 1408 parallel subprocesses. In the problem considered, training can be parallelized into 20 531 processes, corresponding to the number of all genes in the profile. The maximum possible parallelization is $20\,531 * (2R + 1)$ processes, as for each gene, all values within the generalization radius can be checked in parallel.

We achieved parallelization using the *cuda* module from the *numba* library. To process threads by the GPU multiprocessors, the threads were divided into blocks using the following formula:

$$\mathit{blocks\_per\_grid} = (\mathit{number\_of\_iterations}| + (\mathit{threads\_per\_block} - 1))//\mathit{threads\_per\_block},$$

where *number_of_iterations* is the number of parallel iterations.

Before the execution of the program, we copied the arrays of input data from the main memory to the GPU memory using the *cuda.to_device*() function, and upon completion of the program, we copied the execution results back to the main memory using the *device_array.copy_to_host*() function.

We used a decorator to parallelize the write and read functions. Within the parallelized functions, we call a function returning the index of the thread on which the corresponding parallel iteration should be executed.

5. RESULTS

Table 3 presents the accuracy and the number of operations for the indextron and the neural network training for the disease diagnosis problem based on data from 801 patients.

Table 3. Results comparison.

	Neural network	Indextron
Accuracy (%)	99.5	99.75
Number of operations	46 bln addition/ multiplication operations	8 mil memory write operations

The indextron training time on a single-core 1.6 GHz laptop is 0.43 s. Training time of the four-layer neural network on the same hardware increases, according to the increase in the number of operations, $46 * 109/8 * 106 = 5750$ times. Note that the architecture of the indextron is ideal for parallel implementation. It allowed us to achieve almost instant learning in 75 ms for the problem considered. Generalization radii R in the training and recognition modes are 0 and 84 correspondingly, i.e., 33% of the feature variation range (0–255).

6. CONCLUSION

1. The simplicity of the indextron algorithm, which only requires writing K integers to memory for training each pattern and has classification complexity of $O(hK)$ operations for reading and

summation, allows the creation of large gene expression databases to diagnose various diseases (h is the average column height, where $h = N/X$).

2. With the wide availability of equipment for finding gene expression profiles, it becomes possible to create a search diagnostic system similar to Google for mass use, where queries are in numerical form rather than text.

3. The experiments carried out confirm the existence of such a possibility, as fast training with the addition of new data to the database, even at the software level, takes only $0.43/800 = 0.00054$ s, and the accuracy of diagnosing five types of cancer was 99.75%. However, diagnosing many other diseases will require the creation of a large number of different databases.

REFERENCES

1. Khan, J., Wei, J., Ringner, M., et al. Classification and Diagnostic Prediction of Cancers Using Gene Expression Profiling and Artificial Neural Networks, *Nat Med.*, 2001, June 7(6): 673–9. https://doi.org/10.1038/89044

2. Kumar, A. and Halder, A., Greedy Fussy Vaguely Quantified Rough Approach for Cancer Relevant Gene Selection from Gene Expression Data, *Soft Comput.*, 2022, vol. 26, pp. 13567–13581. https://doi.org/10.1007/s00500-022-07312-4

3. Houssein, E., Hassan, H., Mustafa, al-sayed, et. al., Gene Selection for Microarray Cancer Classification based on Manta Rays Foraging Optimization and Support Vector Machines, *Arabian Journal for Science and Engineering*, 2022, vol. 47, pp. 2555–2572. https://doi.org/10/1007/s13369-021-06101-8

4. Zheng, Y., Sun, Y., Kuai, Y., et al., Gene Expression Profiling for the Diagnosis of Multiple Primary Malignant Tumors, *Cancer Cell Int.*, 2021, vol. 21, Article no. 47. https://doi.org/10.1186/s12935-021-01748-8

5. Ye, Q., Wang, Q., Qi, P., et. al., Development and Validation of a 90-Gene Real-Time PCR Assay for Tumor Origin Identification, *Symposium MXW*, 2018.

6. Joshi, P. and Dhar, R., EpICC: A Bayesian Neural Network Model with Uncertainty Correction for a More Accurate Classification of Cancer, *Sci. Rep*, 12, 2022, Article no. 14628. https://doi.org/10.1038/s41598-022-18874-6

7. Steiling, K. and Christenson, S., Tools for Genetics and Genomics: Gene Expression Profiling, *UpToDate*, 2021. Retrieved from https://www.uptodate.com/contents/ tools-for-genetics-and-genomics-gene-expression-profiling

8. St. Petersburg University Research Park. High-throughput complete genome sequencing system, 2023 https://researchpark.spbu.ru/equipment-biobank-rus/ equipment-biobank-genom-rus/equipment-biobank-ngsseq-rus/1762-biobank-hiseq-2500-sequencing-system-rus

9. IBM. What are neural networks? // Retrieved from https://www.ibm.com/cloud/learn/neural-networks

10. Mikhailov, A. and Pok, Y.M., Artificial Neural Cortex, *Smart Engineer. Syst. Design.*, 2001, vol. 11, ASME PRESS, N.Y. P. 113–120.

11. Mikhailov, A. and Karavay, M., Pattern Inversion as a Pattern Recognition Method for Machine Learning, *Cornell University*, 2021. Retrieved from https://arxiv.org/abs/2108.10242

12. Brin, S. and Page, L., The Anatomy of a Large-Scale Hypertextual Web Search Engine, *Comput. Networks ISDN Syst.*, 1998, vol. 30, nos. 1–7. Stanford University, Stanford, CA, 94305, USA. Retrieved from https://doi.org/10.1016/S069-7552(98)00110-X

13. Mikhailov, A., Indextron, *Artificial Neural Networks in Engineering Conf. (ANNIE 1998)*, St. Louis, Missouri, Nov. 4–7, 1998. Proceedings Vol. 8: ANNIE 1998, Publisher: ASME Press, ISBN: 0791800822

14. Jones, K., A Statistical Interpretation of Term Specificity and Its Application in Retrieval, *J. Document.*, MCB Univer.: MCB Univer. Press, 2004, vol. 60, no. 5. pp. 493–502. ISSN 0022-0418

15. Sivic, J. and Zisserman, A., Efficient Visual Search of Videos Cast as Text Retrieval, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2009, vol. 31, no. 4. https://doi.org/10.1109/TPAMI.2008.111

16. UCI. Machine learning repository // Retrieved from https://archive.ics.uci.edu/ml/datasets/gene+expression+cancer+RNA-Seq

17. Mikhailov, A. and Karavay, M., Indextron, *Proceedings of the 10th International Conference on Pattern Recognition Application and Methods*, 4–6 Feb 2021, Vienna, V.1-978-989-758-486-2, pp. 143–149. https://doi.org/10.5220/0010180301430149

This paper was recommended for publication by O.N. Granichin, a member of the Editorial Board