



THE INSTANCE SELECTION METHOD FOR NEURO-FUZZY MODEL SYNTHESIS

Sergey A. Subbotin

Department of Program Tools, Zaporizhzhya National Technical University, Zhukovskiy str., 64,
Zaporizhzhya, 69063, Ukraine, subbotin@zntu.edu.ua, http://www.csit.narod.ru

Abstract: The problem of automation of neuro-fuzzy model synthesis based on instance set is solved. The method of instance selection for neuro-fuzzy model synthesis is proposed. The proposed method allows reducing the sample size and decreasing the requirements to computer resources. The method also performs transformation of the original multi-dimensional coordinate set to the one-dimensional axis, which is also discretized to improve the data generalization properties. The software implementing the proposed method is developed. The experiments were conducted to investigate the proposed method at the real problem solution. Obtained experimental results allow to recommend the proposed method for practical use. *Copyright © Research Institute for Intelligent Computer Systems, 2014. All rights reserved.*

Keywords: instance, neuro-fuzzy network, sample selection, data dimensionality reduction.

1. INTRODUCTION

The neuro-fuzzy networks are a paradigm of computational intelligence widely used for building of diagnostic and recognizing models [1-3]. The neuro-fuzzy network usually requires a training set of observations (instances) to build the model [3].

In practice, a number of problems faced with the need to process a large amount of available data which can not be loaded completely to the computer memory, as well as the fact that time of model building essentially depends on the training set volume.

Therefore, the actual problem is to reduce the volume of processed sample. It can be made through the allocation of training and test samples of smaller size from the available initial large sample. Known sampling methods are based on exhaustive search [4-7] and random search [5, 6, 8, 9].

The method of full exhaustive search [4] successively considers all possible subsamples – combinations of instances of the original sample and for each such subsample imply building a model or estimate of the value of a criterion which characterizes the ability of a subsample replace the original sample. Such method is applicable only for the initial sample of small dimension.

Methods of reduced search [5-7] imply cutting off part of unpromising solutions in the process of example combinations busting. They allow solving problems of large dimension compared to exhaustive search, but they are also slow.

The random search methods [5, 6, 8, 9] consider a limited number of combinations of instances, seeking to create new subsamples based on experience gained during the assessment of previous subsamples and use random or evolutionary search. These methods are applicable to samples of higher dimension, but no guarantee of an acceptable solution with a limited number of iterations of the search.

In general, we can conclude that known sampling methods suggests iterating over a large number of possible combinations of exemplars. This is for a large amount of initial sample leads to the so-called “combinatorial explosion”. These methods also require a set of criteria to assess the quality of partitions. Despite the existence of such criteria [10], in practice, for a large initial sample their use leads to significant costs of computer time.

The aim of this work was to develop a method that allows automatically splitting a large initial sample into training and test samples within the constraints of computer memory.

2. PROBLEM STATEMENT

Suppose we have an initial sample $X = \langle x, y \rangle$ that is a set of S instances describing dependence $y(x)$, $x = \{x^s\}$, $y = \{y^s\}$, $s = 1, 2, \dots, S$. The sample is characterized by a set of N input attributes (features) $\{x_j\}$, $j = 1, 2, \dots, N$, where j is a number of feature, and by output feature y . Each s -th instance can be represented as $\langle x^s, y^s \rangle$, $x^s = \{x_j^s\}$, where x_j^s is a value

of j -th input and y^s is the output feature of s -th instance (exemplar) of a sample, $y^s \in \{1, 2, \dots, K\}$, where K is a number of classes, $K > 1$.

Then the problem of the sample size reduction can be represented as the problem of the formation (separation) from the original sample $X = \langle x, y \rangle$ of sub-sample X^* , $X^* \subset X$, with a less volume $S^* < S$, with the most important properties of the original sample [3].

Since the class topology preservation is the most important for automation of diagnostic decisions making and for tasks of automatic classification the formed sub-sample must ensure the preservation of the original sample exemplars located at the class borders.

3. THE METHOD OF INSTANCE SELECTION

The most obvious basis for a method of the significant instance selection from the original data sample is a cluster analysis [11, 12], followed by determination of exemplars located on the borders of the classes [4, 6]. However, this method has a number of disadvantages.

A first disadvantage of this method is its practical applicability primarily for small volume samples because of the need for calculating and storing matrices of distances between exemplars in a computer memory. Therefore, for a large sample of observations we propose to implement a consistent processing of exemplars such that did not need to keep distances between all exemplars, and thereby provide an economy of computer memory using.

A second disadvantage of this method is the difficulty of determining the exemplars located on the borders of the classes in a multi-dimensional feature space. Therefore, to avoid this disadvantage we propose to replace the multi-dimensional set of coordinates on a one-dimensional, which is also discretized to increase the generalizing properties of the method.

The third disadvantage of the method is the uncertainty of the number of clusters in the implementation of cluster analysis.

As a rule, in most tasks the user can not know in advance the number of clusters, and the automatic determination of the number of clusters requires a large number of partition variants sorting out, as well as a calculating and storing in the memory of the distance matrix between all exemplars.

To avoid this disadvantage we offer at the beginning to determine the limitations on the number of clusters and define the coordinates of their centers to coat the all areas of the feature space. Then we propose to perform recognition of exemplars of the original sample based on a set of

cluster centers by assigning to the cluster centers the numbers of the classes of closest to them exemplars. In cases when the collisions (situations when the center of cluster nearest to the recognized exemplar belongs to a class different from its) will occur we will form new clusters, recording at their center coordinates recognized exemplars that come into collision with the nearest cluster center.

After that, for the formed cluster set we propose to perform a join of all of the neighboring clusters belonging to the same class. This allows, on the one hand, providing initially a higher level of generalization of data in comparison with a cluster analysis with a reduction of the number of clusters, and, on the other hand, will accelerate the calculation in comparison with the cluster analysis with an increasing of the number of clusters.

The considered ideas are at the basis of the proposed method of sampling.

The stage of initialization: Make original sample X . Evaluate minimal x_j^{\min} and maximal x_j^{\max} values for each j -th feature, $j=1, 2, \dots, N$. Create Q centers of clusters $C^q = \{C_j^q\}$, $q=1, 2, \dots, Q$, $j=1, 2, \dots, N$, where C_j^q is the value of j -th feature for the center of q -th cluster: $x_j^{\min} \leq C_j^q \leq x_j^{\max}$. Note that $K \leq Q \ll S$.

In the simplest case each C_j^q may be set as a random number, but it seems as more convenient for this the using of a formula:

$$C_j^q = \begin{cases} \frac{1}{S^q} \sum_{s=1}^S \{x_j^s \mid y^s = q\}, q \leq K; \\ C_j^{q-K} + (rand_1 - rand_2)(x_j^{\max} - x_j^{\min}), \\ x_j^{\min} < C_j^{q-K} + (rand_1 - rand_2) \times \\ \times (x_j^{\max} - x_j^{\min}) < x_j^{\max}, K < q; \\ x_j^{\min} + rand_1(x_j^{\max} - x_j^{\min}), \text{ otherways,} \end{cases} \quad (1)$$

where $rand_1, rand_2$ are two different random numbers: $rand_1, rand_2 \in [0, 1]$, Q is set by the user or automatically choose by the formula:

$$Q = \begin{cases} K, K \geq \text{round}(\ln(S)); \\ \text{round}(\ln(S)), \text{round}(\ln(S)) > K, \end{cases} \quad (2)$$

where round is a function of rounding to the nearest integer.

After that set for each cluster C^q , $q=1, 2, \dots, Q$ the class number: $Y^q=0$, and also the number of exemplars in the original sample, that located in the q -th cluster: $S_q=0$.

The stage of sample partitioning into clusters. Looking through original sample for each exemplar x^s , $s=1, 2, \dots, S$ perform following actions:

– evaluate the distance from exemplar x^s to each cluster center C^q :

$$R(x^s, C^q) = R(C^q, x^s) = \sqrt{\sum_{j=1}^N (x_j^s - C_j^q)^2}, \quad (3)$$

where $q = 1, 2, \dots, Q$;

– find the number of cluster center nearest to the exemplar x^s :

$$q = \arg \min_{g=1,2,\dots,Q} \{R(x^s, C^g)\}, \quad (4)$$

– if the center does not has class label ($Y^q=0$), then set as its label the class number of exemplar x^s : $Y^q=y^s, S_q=1$;

– if the class of exemplar x^s and the class of cluster center C^q are equal ($Y^q=y^s$), then set: $S_q=S_q+1$;

– if the class of exemplar x^s and the class of cluster center C^q are not equal ($Y^q \neq y^s$), then add new cluster: $Q=Q+1, C^q=x^s, Y^q=y^s, S_q=1$.

The stage of cluster set reduction. Looking through the set of formed cluster centers perform such actions for each q -th cluster, $q=1,2, \dots, Q$:

– if the number of exemplars located in it is equal to zero ($S_q = 0$), then delete q -th cluster, and correct the number of clusters: $Q=Q-1$, and also renumber the elements $\{Y^q\}$ and $\{S_q\}$, then go to next cluster processing;

– find distances from the q -th cluster to all another clusters:

$$R(C^q, C^g) = R(C^g, C^q) = \sqrt{\sum_{j=1}^N (C_j^g - C_j^q)^2}, \quad (5)$$

where $g = 1, 2, \dots, Q, g \neq q$;

– find the cluster C^p nearest to q -th cluster:

$$p = \arg \min_{g=1,2,\dots,Q} \{R(C^q, C^g)\}, \quad (6)$$

– if $Y^q=Y^p$, then join q -th and p -th clusters by the formula:

$$C_j^q = \frac{C_j^q + C_j^p}{2}, j=1, 2, \dots, N, \quad (7)$$

or with account their exemplar numbers by the formula:

$$C_j^q = \frac{S_q C_j^q + S_p C_j^p}{S_q + S_p}, j=1, 2, \dots, N, \quad (8)$$

after that correct the number of clusters: $Q=Q-1$, and also renumber elements $\{Y^q\}$ and $\{S_q\}$, than go to next cluster processing.

The stage of collision resolution and cluster addition: Set $S_q=0$. Looking through the original sample for each exemplar $x^s, s=1, 2, \dots, S$ make such actions:

– evaluate distances from it to centers of each cluster:

$$R(x^s, C^q) = R(C^q, x^s) = \sqrt{\sum_{j=1}^N (x_j^s - C_j^q)^2}, \quad (9)$$

where $q = 1, 2, \dots, Q$;

– find the number of cluster center closest to exemplar x^s :

$$q = \arg \min_{g=1,2,\dots,Q} \{R(x^s, C^g)\}, \quad (10)$$

– if it belongs to the same class that exemplar x^s ($Y^q=y^s$), then set as a coordinate of an exemplar on the generalized axe: $x_*^s=q, y_*^s=Y^q, S_q=S_q+1$;

– if the class of exemplar x^s and the class of nearest to it cluster C^q does not equal ($Y^q \neq y^s$), then add new cluster: $Q=Q+1, C^q=x^s, Y^q=y^s, S_q=1$, and set as a coordinate of an exemplar on the generalized axe: $x_*^s=q, y_*^s=Y^q$.

The stage of evaluation of individual feature informativeness: At the beginning let renumber exemplars $\{x_*^s, y_*^s\}$ on the generalized axe in the order of increase of value of x_* . Then sequentially process exemplars on the generalized axe $x_*^s, s=1,2, \dots, S$:

– find minimal and maximal values of each feature in the original feature space for exemplars with equal coordinates on the generalized axe separately for each class:

$$x_j^{\min,q} = \min_{p=1,2,\dots,S} \{x_j^p \mid y^p = q, x_*^p = x_*^q\}, \quad (11)$$

$$x_j^{\max,q} = \max_{p=1,2,\dots,S} \{x_j^p \mid y^p = q, x_*^p = x_*^q\}, \quad (12)$$

– for each j -th feature, $j=1,2, \dots, N$, evaluate the number of exemplars of each q -th class S_j^{q+} and the number of exemplars of other classes S_j^{q-} , that located by j -th feature in the interval of its values for q -th class in the original feature space for specific coordinate on at the generalized axe:

$$S_j^{q+} = \sum_{s=1}^S \sum_{p=s}^S \left\{ \left| \begin{array}{l} x_j^{\min,q} \leq x_j^s \leq x_j^{\max,q}, x_j^{\min,q} \leq x_j^p \leq x_j^{\max,q} \\ y_*^s = q, y_*^p = q, x_*^s = x_*^p \end{array} \right. \right\}, \quad (13)$$

$$S_j^{q-} = \sum_{s=1}^S \sum_{p=s+1}^S \left\{ \left| \begin{array}{l} x_j^{\min,q} \leq x_j^s \leq x_j^{\max,q}, x_j^{\min,q} \leq x_j^p \leq x_j^{\max,q} \\ y_*^s = q, y_*^p \neq q, x_*^s = x_*^p \end{array} \right. \right\}, \quad (14)$$

After that for each j -th feature, $j=1, 2, \dots, N$:

– evaluate weight (individual evaluation of informativeness) of j -th feature for q -th cluster, $q=1, 2, \dots, Q$:

$$w_j^q = \begin{cases} 1, S_j^{q-} = 0; \\ \frac{S_j^{q+}}{S_j^{q-}}, S_j^{q-} > 0; \end{cases} \quad (15)$$

– define weight (individual evaluation of informativeness) of j -th feature for all set of clusters:

$$w_j = \max_{q=1,2,\dots,K} \{w_j^q\}, \quad (16)$$

$$\text{or } w_j = \frac{\max_{q=1,2,\dots,K} \{w_j^q\}}{\max_{j=1,2,\dots,N} \{w_j^q\}}, \quad (17)$$

$$\text{or } w_j = \frac{\frac{1}{K} \sum_{q=1}^K w_j^q}{\max_{j=1,2,\dots,N} \{w_j^q\}}, \quad (18)$$

The stage of sample partitioning on training and test samples include such actions:

– create the initial training X^* and test X' samples: $X^* = \emptyset, X' = \emptyset$;

– find all exemplars in the initial feature space with equal coordinates on the generalized axe but different class numbers and put them into a formed training set X^* :

$$X^* = \bigcup_{p=1}^s \{ \langle x^p, y^p \rangle \mid \langle x^p, y^p \rangle \notin X^*, x_*^p = x_*^s, y^p \neq y^s \}, \quad (18)$$

where $s=1, 2, \dots, S$;

– find all exemplars in the initial feature space with equal coordinates on the generalized axe and equal class numbers and put into a formed training set one of them, which is more closely located to the center of corresponding cluster:

$$X^* = X^* \cup \langle x^q, y^q \rangle, \quad (20)$$

where $q = \arg \min_{g=1,2,\dots,Q} \{R(x^p, C^g) \mid \langle x^p, y^p \rangle \in \Omega^s\}$,

$$R_w(x^s, C^q) = R_w(C^q, x^s) = \sqrt{\sum_{j=1}^N w_j (x_j^s - C_j^q)^2},$$

$$\Omega^s = \bigcup_{p=1}^S \{ \langle x^p, y^p \rangle \mid \langle x^p, y^p \rangle \notin X^*, x_*^p = x_*^s, y^p = y^s \},$$

$s=1, 2, \dots, S$;

– move the rest of exemplars to the test set: $X' = X \setminus X^*$.

The developed sampling method allows selecting training and testing samples from the large volume original sample. It performs a small number of passes through the original sample and does not require downloading it to a computer memory and storage in the memory of matrix of distances between exemplars of the original sample.

The additional result of the method is a set of formed cluster centers of coordinates, which can be used to define the subsequent construction of diagnostic and recognition models.

The received individual estimates of feature informativeness also allow considering the proposed method not only as a method of sampling, but also as a method of feature informativeness evaluation. The use of feature informativeness estimates is possible in the methods of feature selection, as well as in some methods of diagnostic and recognition model building.

4. THE COMPLEXITY ANALYSIS OF INSTANCE SELECTION METHOD

To evaluate the temporal and spatial complexity of the proposed method we will proceed from its implementation based on a computer with sequential computations, and the dimension of the memory we will estimate in the cells containing real numbers. The estimates of complexity will be giving in so-called “soft” form, where there is no suppression of terms of lower order by the terms of large orders.

For the initialization phase the time complexity will be $O(6NS+2Q)$, and space complexity – $O(NS+NQ+2N+2Q)$. For the stage of the cluster formation the time complexity will be $O(2SQN+4SQ)$, and space complexity – $O(NQ)$. For the stage of cluster set reduction the time complexity will be $O(Q+2Q^2N+Q^2+QN)$, and space complexity – $O(Q^2)$. For the stage of the collision resolution and cluster adding the time complexity will be $O(SNQ+SQ+5S)$, and space complexity – $O(NQ+3S)$. For the stage of feature individual informativeness evaluation the time complexity will be $O(2SNQ+20NS^2+2N+KN+KN^2)$, and space complexity – $O(2NQ+2NQ+NQ+N)$. For the stage of original sample partitioning to the training and test samples the time complexity will be $O(4S^2+4S^2+4QN+Q)$, and space complexity – $O(2S+2S+Q)$.

However, we do not take into account the amount of memory for the generated training and test samples that can be stored in RAM or in the external computer memory. Obviously that the total amount of memory for storing the generated samples do not exceed the dimension of the original sample NS .

The total time complexity of all stages of the method is $O(6NS+4Q+5SQN+5SQ+5S+2Q^2N+$

$+Q^2+5QN+20NS^2+2N+KN+KN^2+8S^2$), and space complexity – $O(NS+8NQ+3N+3Q+Q^2+7S)$.

To simplify the analytical assessments we will take relations of parameters that are reasonable from a practical point of view: $n=NS$, $K=2$, $N \approx 0,25S \approx 2\sqrt{n}$, $Q \approx 0,25S \approx 0,0625N \approx 0,125\sqrt{n}$.

As a result, taking into account the assumptions and rounding we get concerning to problem input dimensionality n the complexity estimations of the proposed method: temporal is $O(162,0625 n\sqrt{n}+148,25n+56\sqrt{n})$ and space is $O(3,015625n+68\sqrt{n})$.

5. EXPERIMENTS AND RESULTS

To verify the practical applicability of the proposed method it was implemented as a program. Program implementing proposed method was used to solve practical problems of diagnosis and pattern recognition [13–15].

The characteristics of the original samples for the problem solution as well as the results of experiments on the proposed method investigation are given in the Table 1. Here n_{tr} is a dimension of the generated training set.

Table 1. Characteristics of the original samples and the results of experiments on the formation of samples

Problem	N	S	K	n	n_{tr}/n
Prediction of increase of the surface strength of the blades of gas turbine engines (with the discrete output) [13]	12	59	2	708	0,39
Automatic vehicle classification on the image (by the original and constructed features) [14]	4122	1062	3	4377564	0,23
Recognition of cultural and weed agricultural plants [15]	256	3226	2	825856	0,12

The data set for task of prediction of the surface strength increase of the gas turbine engines blades were collected on the Motors Sich JSC air engine building plant and characterized by the blade material and the hardening process parameters. After the blade hardening the hardening coefficient for each item were obtained. It reflects the increasing of blade surface strength. The blades were separated on classes according to their hardening coefficients [13].

The data set for task of automatic vehicle classification on the image (by the original and constructed features) were collected on Italian autostrade and consists from 64x64 pixel gray scale images of vehicles, which were used for constructive

features (image moments, characteristics of symmetry etc.) computing. Each image was manually associated with corresponding vehicle type (car, truck and bus) [14].

The data set for the task of plant recognition was consists from spectral characteristics of agricultural plants and mostly typical weeds and also provides a plant class [15].

The conducted experiments confirmed the efficiency of the proposed method and software implementing it.

For each data set the proposed method selects a subsample, for which dimension reduction ratios n_{tr}/n is shown in table. As it can be seen from the table, the proposed method significantly reduces the dimension of the training data and saves this time on the subsequent model construction, and enhances the generalizing properties of the synthesized models with respect to the dimension of the original data samples.

The formed subsamples were used for model building using feed-forward artificial neural networks trained by the Levenberg-Marquardt method [2]. Each network was trained separately on the full original and on the reduced training sets. After the training models were tested by recognition the whole initial data set. The results of model testing shows that error of a model built on a reduced sample did not differ significantly from the error of a model built on the entire original sample (difference of errors was estimated on average 3%, and errors were errors were acceptable for problems). This demonstrates the effectiveness of the proposed method for sampling.

The dimension of the original sample and the complexity of the class separation in corresponding problem limit the effectiveness of the proposed method. Obviously, the higher the complexity of the separation of classes, the more instances of the original sample need to be included in the formed subsample. On the other hand, the greater the original sample, the more redundant examples it may contain, and, accordingly, the greater the effect of the proposed method using. Conversely, the smaller the original sample, the less of the redundant instances it contains, and the less the effect of the proposed method.

The developed method for sampling also defines the individual evaluation of feature informativeness. As a result dimensionality reduction of training samples and model complexity synthesized on their base can be achieved.

5. CONCLUSION

The problem of autonomously partitioning the original sample into training and test samples that create instances of diagnostic and recognizing models have been addressed in the paper.

The scientific novelty of the new method preserves, in a generated sub-sample, the most important topological properties of the original sample without the need to load the original sample into computer memory. As a consequence, multiple passes over the original sample is not required. This significantly reduces the sample size, and decreases the requirements for computer resources.

The practical significance of the presented results is the determination of estimates of temporal and spatial complexities, the software implementation of the sampling method, and experiments performed on real problems to demonstrate the capability of the proposed method. The results of experiments allow recommending the proposed method for use in practice.

Further research could be directed towards the development of effective implementation of the proposed method for multi-processor (multi-core) computers working in parallel mode.

6. REFERENCES

- [1] D. Ruan, *Intelligent hybrid systems: fuzzy logic, neural networks, and genetic algorithms*, Springer, Berlin, 2012, 354 p.
- [2] S. Sumathi, S. Paneerselvam, *Computational intelligence paradigms: theory & applications using MATLAB*, CRC Press, Boca Raton, 2010, 851 p.
- [3] S. Subbotin, An. Oleinik, E. Gofman, S. Zaitsev, Al. Oleynik, *Intelligent information technologies of automated diagnosis and pattern recognition systems design*, Smith Company Ltd., Kharkov, 2012, 318 p. (in Russian).
- [4] A. Chaudhuri, H. Stenger, *Survey sampling theory and methods*, Chapman & Hall, New York, 2005, 416 p.
- [5] S. Subbotin, Methods of sampling based on exhaustive and evolutionary search. *Automatic Control and Computer Sciences*, (47) (2013), pp. 113-121.
- [6] P. Lavrakas, *Encyclopedia of survey research methods*, Sage Publications, Thousand Oaks, 2008, 1072 p.
- [7] H. Bernard, *Social research methods: qualitative and quantitative approaches*, Sage Publications, Thousand Oaks, 2006, 659 p.
- [8] S. Ghosh, *Multivariate analysis, design of experiments, and survey sampling*, Marcel Dekker Inc., New York, 1999, 698 p.
- [9] M. Plutowski, *Selecting training exemplars for neural network learning*, PhD Thesis in computer science and engineering, University of California, San Diego, 1994, 135 p.
- [10] S. Subbotin, The training set quality measures for neural network learning, *Optical Memory and Neural Networks (Information Optics)*, (19) 2 (2010), pp. 126-139.
- [11] B. Everitt, *Cluster Analysis*, John Wiley & Sons Ltd., Chichester, 2011, 346 p.
- [12] J. Abonyi, B. Feil, *Cluster analysis for data mining and system identification*, Birkhäuser, Basel, 2007, 303 p.
- [13] A. Boguslaev, Al. Oleinik, An. Oleinik, D. Pavlenko, S. Subbotin, *Progressive technologies of modeling, optimization and intelligent automation of aviation engine lifecycle stages*, Motor Sich JSC, Zaporozhye, 2009, 468 p. (in Russian).
- [14] S. Subbotin, K. Boichenko, Automatic system of vehicle detection and recognition on the image, *Software Products and Systems*, (1) (2010), pp. 114-116. (in Russian).
- [15] V. Dubrovin, S. Subbotin, S. Morshchavka, D. Piza, The plant recognition on remote sensing results by the feed-forward neural networks, *Intelligent engineering systems through artificial neural networks*. Vol. 10 – Smart engineering systems design: neural networks, fuzzy logic, evolutionary programming, data mining, and complex systems, ASME Press, New York, 2000, pp. 697-702.



Sergey A. Subbotin, Dr. hab. Sc. received in Kharkiv National University of Radio electronics in 2014, Docent (Associate Professor) received in Zaporizhzhya National Technical University in Ukraine in 2007. He is laureate of Award of President of Ukraine (2004), laureate of the Award of Parliament of Ukraine (2010), Cabinet of Ministers of Ukraine grant-holder (2001–2002). He is now Professor of Department of Program Tools and Deputy Editor-in-Chief of scientific journal "Radio Electronics, Computer Science, Control". His research activities include computational intelligence, technical diagnosis, pattern recognition, machine learning.