Randomized Machine Learning: Statement, Solution, Applications

Yu. S. Popkov*, Yu. A. Dubnov*, A. Yu. Popkov*

*Institute for Systems Analysis
Federal Research Center “Computer Science and Control” of Russian Academy of Sciences
Moscow, Russia
Email: popkov@isa.ru

This work is partially supported by Russian Fund for Basic Research (project no. 16-07-00743)

Abstract—In this paper we propose a new machine learning concept called randomized machine learning, in which model parameters are assumed random and data are assumed to contain random errors. Distinction of this approach from “classical” machine learning is that optimal estimation deals with the probability density functions of random parameters and the “worst” probability density of random data errors. As the optimality criterion of estimation, randomized machine learning employs the generalized information entropy maximized on a set described by the system of empirical balances. We apply this approach to text classification and dynamic regression problems. The results illustrate capabilities of the approach.

Keywords—randomized machine learning; entropy; optimality

I. INTRODUCTION

The resources and processing power of computers have steady growth, facilitating accumulation and storage of huge amounts of data, in natural or digital formats. There appear giant data centers intended for accumulation and storage of these data. And the following question arises immediately: which operations can be applied to data, except their storage? A very attractive idea is endeavoring to extract new knowledge from data. For instance, in 1992 G. Piatetsky-Shapiro formulated the concept of Data Mining (DM) [1], [2] which was to integrate the methods developed in different areas of computer science and methods of mathematical statistics and probability theory within a user-friendly computing environment. Statement of the fundamental problems associated with qualitative and, perhaps, quantitative description of knowledge and adequate data arrays was left to a researcher. The pioneering experiments with Data Mining revealed a whole bunch of negative features of data (errors, reliability, temporal irregularity, etc.) casting doubt on the resulting knowledge. These features become more substantial in case of much heterogeneous data. Data validation in the wide sense forms a serious challenge.

In 2008, Nature editor C. Lynch prepared a special issue [3] dedicated to the big data problem, formulating the concept of Big Data (BD). It was assumed to develop a computing environment aimed at processing methods for the large arrays of data. For the time being, technological aspects of the DM and BD has been essentially developed.

A certain part of the scientific community was fascinated by the data problem, even announcing a new branch of science (known as Data Science) that covers all research relating to data usage.

The physical core of all concepts involving data is a supercomputer implementing computational algorithms for big data arrays handling and knowledge extraction. To succeed, one has to make some hypotheses and learn the computer using mentioned data arrays. Therefore, Machine Learning (ML) procedures and algorithms play a key role in DM and BD concepts. In contrast to these concepts, ML has more than sixty-year history and rich experience in solution of many problems. The first publications on the subject dates back to 1957 when F. Rosenblatt created an algorithm called “the perceptron” [4], implemented as the computing machine “Perceptron MARK I”. Rosenblatt’s work laid the foundations of research on neural networks. The notion of empirical risk, a basic one for ML procedures, was introduced by Ya. Tsypkin in the monograph [5]. The method of potential functions for classification and pattern recognition problems was published in 1970, see the book [6].

Modern ML concept is based on the deterministic parametrization of models and estimation using data arrays with hypothetical properties. Estimation quality is characterized by empirical risk (ER) functions, and minimization yields the corresponding optimal estimates in terms of the chosen ER; the details can be found in [7]–[9].

Generally, the real problems solved by ML procedures are immersed in some uncertain environment. If the matter concerns data, they are obtained with errors, omissions, dubious validity. Model design and parametrization represents a subjective non-formalizable process depending on the individual knowledge of a researcher. Therefore, mass application of ML procedures causes rather high uncertainty. Empirical risk minimization gives parameter estimates for an existing data array and an adopted parameterized model; in other words, these estimates are conditional. There is no clarity about behavior with other data arrays or another parametrization model.

The above circumstances call for uncertainty simulation somehow. Here a general trend is stochastic simulation. We will use it for parameterized models and data. Such an approach means that model parameters are assumed random (randomized in an appropriate way) and data are assumed to
contain random errors. The procedures enjoying these properties will be called Randomized Machine Learning (RML) procedures. Their distinction from the existing ML procedures is that, within RML procedures, optimal estimation deals with the probability density functions (PDF) of random parameters and the “worst” PDF of random data errors, instead of the parameters proper. As the optimality criterion of estimation, RML employs the generalized information entropy maximized on a set described by the system of empirical balances.

To use the model “learned” by ML procedure, it is necessary to replace its parameters with the corresponding estimates yielded by empirical risk minimization and then calculate its output. In RML procedure, the “learned” model is equipped with the optimal PDF of the parameters and the “worst” PDF of the noises simulating data errors. The output of the randomized model is an ensemble of random vectors or random trajectories that are “best” under maximum uncertainty.

II. MODERN STRUCTURE OF ML PROCEDURE

ML procedures are designed via system integration of many subject domains from natural sciences and humanities; here the role of integrator is played by the computer-oriented branches of mathematics. Perhaps, it would not be an overstatement that the general formulation of the machine learning problem involves the data–models–algorithms triad, see [8]–[13]. To a large extent, all elements of this triad rely on the corresponding subject domains, predetermining their specifics illustrated below using some examples. However, despite the above specifics, it seems fruitful to structure the general concept of ML procedure (see Fig. 1) and the existing statements of ML problems varying in form, yet being similar in essence.

a) Data block (D): There is a set of data composed of two groups, namely, the “input” data arrays \( X \in X \) and the “output” data arrays \( Y \in Y \). These data are generated by a real object and, by assumption, there exists a connection between the “input” and “output” data in terms of an unknown functional \( \Omega \). For instance, the functional \( \Omega \) describes an industrial object (identification, dynamic regression), a “learner” associating each set of input data with a set of output data (classification, clustering), or may have other forms depending on a problem under consideration [14]–[16].

The above arrays are formed by \( s \) observations of the input vectors \( x \in R^n \) and the output vectors \( y \in R^m \). In the sequel, the input arrays will be characterized by a matrix \( X \) with the columns \( x^{(j)}, j = \frac{1}{s}, \ldots, s \) or by matrices \( X^{(j)} \) containing \( g \) columns of the form \( x^{(j)}, x^{(j+1)}, \ldots, x^j \). The output arrays represent a matrix \( Y \) compiled from the vectors \( y^{(j)}, j = \frac{1}{s}, \ldots, s \). The major problems of primary data processing include data validation, compensation of omissions (if any) and array dimension reduction without loss of information. The last problem substantially depends on a problem under consideration [17], [18].

b) Parameterized models block (PM): ML procedure aims at constructing a suitable approximation \( \hat{\Omega} \) of the functional \( \Omega \) using \( s \) observations \( j = \frac{1}{s} \). To this effect, it is necessary to adopt a certain model

\[
\hat{y}^{(j)} = \hat{\Omega}(X^{(j)}, a)
\]

with unknown parameters \( a \). The choice of an appropriate model is the result of knowledge and ingenuity of a researcher. The parameter vector \( a \) contains real-valued entries only. For a fixed parameter vector \( \hat{a} \), PM block generates the response (output of the model) \( \hat{y}^{(j)}(\hat{a}) \); it can be compared with the output data \( y^{(j)} \), trying to adjust the parameters so that \( \hat{y}^{(j)} \) and \( y^{(j)} \) become close to each other [19]–[21].

c) Algorithms block ML-A: How should such a comparison be performed? What is a good measurement method for the comparison results? Actually, these questions induce very many ML algorithms. And the answers heavily depend on the hypotheses about data. Modern machine learning employs two fundamental hypotheses as follows [22].

The first hypothesis considers the data and parameters as deterministic sets of real or integer values. In this case, comparison is performed using the vector-type loss function

\[
e^{(j)}(a | X^{(j)}, y^{(j)}) = y^{(j)} - \hat{y}^{(j)}(X^{(j)}, a), \quad j = \frac{1}{s}, \ldots, s
\]

and the closeness of outputs and the object’s output data is characterized by its norm known as the empirical risk, i.e.,

\[
Q_{X,Y}(a) = \frac{1}{s} \sum_{j=1}^{s} \| e^{(j)}(a | X^{(j)}, y^{(j)}) \|.
\]

Empirical risk minimization methods define the first family of deterministic ML algorithms [23], [24] (see Fig. 1):

\[
\hat{a}(X, y) = \arg \min Q_{X,Y}(a).
\]

The other hypothesis treats the data \( Z = (X, Y) \) and the parameters \( a \) as random objects having a joint probability density function \( P(Z, a) \). Here \( Z \) denotes the universe of data and \( a \) is a random vector. The observed data arrays are interpreted as samples \( Z^{(j)} \) from the universe \( Z \). Note that each of them is random with the PDF \( P(Z^{(j)}, a) \).

The joint probability density function

\[
L(Z^{(1)}, \ldots, Z^{(s)}; a) = \prod_{j=1}^{s} P(Z^{(j)}, a)
\]
is called the likelihood function for the sample \( \{Z^{(1)}, \ldots, Z^{(s)}\} \) \cite{25}, \cite{26}. Under a fixed sample, the likelihood function depends on the parameter \( a \). A quite natural estimate of this parameter is a vector maximizing the likelihood function. For convenience, the likelihood function is often replaced by its logarithm \( \hat{L} = \log L \). Therefore, the likelihood maximization methods form the family of probabilistic ML algorithms:
\[
\hat{a}(Z^{(1)}, \ldots, Z^{(s)}) = \arg\max_a L(Z^{(1)}, \ldots, Z^{(s)}, a). \tag{6}
\]

The function \( P(Z, a) \) fully characterizes the random objects (the data and parameters) irrespective of the experiments (trials) used to obtain this characteristic and the characteristics of the parameters and data before trial and after trial. T. Bayes suggested the concept of conditional probability, distinguishing between the prior and posterior probabilistic characteristics of the data and parameters. This approach yielded the notions of the posterior conditional probability density \( p(Z | a) \) of the data under fixed parameters, the prior conditional probability density \( w(a | Z) \) of the parameters under fixed data and the prior probability density \( g(Z) \) of the data. They have a relationship defined by the Bayes formula
\[
p(Z | a) = \frac{w(a | Z)g(Z)}{\int_Z w(a | Z)g(Z)dz}. \tag{7}
\]

Posterior probability density maximization methods form the second family of probabilistic ML algorithms: \( \hat{a}(Z) = \max_a p(Z | a) \). \tag{8}

Both families of ML-A, (6, 8), are based on the joint probability density functions of the data and parameters. Definition of these functions involve hypotheses regarding the probabilistic properties of the random objects. According to the above expressions, within both approaches ML-A yield the numerical estimates of the parameters associated with the data, i.e., samples from universes.

III. GENERAL PROPERTIES OF ML PROCEDURES

The above analysis of the structure and statements of ML problems gives the following conclusions on their general properties.

1) Block D operates the data having unknown origin or random nature. There are no grounds to assign some “native” characteristics to the first group of data. Due to their specifics, the repeatability of the results obtained by a complete machine learning cycle is not guaranteed for other data arrays. Perhaps, the level of such guarantee can be increased using larger learning arrays; but it is always possible to construct (or encounter) a data array for which the learning model would generate an inadequate solution of the initial problem.

The situation seems more encouraging for the data possessing random nature. At the same time, “random nature” makes up merely a hypothesis, and its quantitative description is transformed into a probability density function or a distribution function. As the data arrays have a finite length, only some approximations of these functions can be considered. But how changes in the set of data arrays would affect the approximations is actually unknown. Therefore, the absence of the above guarantee for the first group of data is “roofed” by the probabilities for the second group of data.

2) Classification, clustering and pattern recognition problems proceed from an assumption that the data in block D have exact attributes. If these data represent numerical values, then they are assumed to be the result of exact measurements of a real process. However, in applications such an assumption seems rather questionable. For instance, in the text classification problems, the input data — the weights of documents — may incorporate errors as the volumes and other indicators of collections vary unpredictably. The classes of warehouses possibly intersect, complicating the task of a “learner” who assigns a current document to an appropriate class. This aspect causes errors in the learning sequence. The behavioral modeling problems of dynamic objects operate data measured by imperfect devices; their procedural and random errors contribute to the performance of ML algorithms.

3) ML procedures employ models with nonrandom parameters having unknown values. Here learning aims at estimating the values of these parameters. As a matter of fact, the parameterized models comprise two classes, namely, the parameterized relationships between the input and output data and the parameterized joint probabilistic characteristics (probability density functions and probability distributions) of the input and output data. But the parameters are nonrandom in both classes of models, and the term “probabilistic models” applies to the data models only (not to their parameters).

4) All ML procedures are based on minimization of functions that reflect empirical risk. These functions depend on the available data and parameters of the model. An important property of ML algorithms is their orientation towards the unconditional minimization methods of a multivariable function.

IV. RML PROCEDURE

Define RML procedure, relying on the structure of ML procedure described in section 2, see Fig. 1. Such an approach makes it possible to compare both procedures, fixing their distinctions.

a) Data block (D): There are two data arrays, the first being treated as the input one with a matrix \( X = [x^{(1)}, \ldots, x^{(s)}] \) and the second as the output one with a matrix \( Y = [y^{(1)}, \ldots, y^{(j)}] \). Here the vectors belong to appropriate spaces, \( x^{(j)} \in R^m \), \( y^{(j)} \in R^m \), and \( s \) denotes the number of measurements (observations) for the input and output. By assumption, there exists a connection between them, with a practical interpretation predetermined by problem specifics. In the text classification problems, input data form the learning matrix of term weights in a document, while output data are the matrix of the “learner” answers on document belonging to a corresponding class. In the industrial process modeling problems, input data form the numerical matrix of raw material characteristics, while output data are the numerical matrix of industrial product characteristics. Suppose that the input
array contains exact data, whereas the output one contains data with interval-type errors, i.e., known ranges. The output data errors will be characterized by a random noise matrix \( \Xi = [\xi^{(1)}, \ldots, \xi^{(s)}] \). Here the random vectors \( \xi^{(j)}, s = 1, s \) are independent with the independent components:

\[
\xi^{(j)} \in \mathbb{E}^\alpha \left[ \xi^{(j)} \right], \quad k = 1, m, j = 1, s; \nonumber \]

\[
\xi^{(j)} \in \Xi_j = \bigcup_{k=1}^m \mathbb{E}^{\alpha} \xi^{(j)}; \quad \Xi = \bigcup_{j=1}^s \Xi_j = [\Xi^-, \Xi^+]. \tag{9} \]

The probabilistic properties of the noises are characterized by the probability density functions (PDFs) of their components \( q_k^{(j)}(\xi^{(j)}) \). We believe that the PDFs represent continuously differentiable functions. Due to independence,

\[
Q(\Xi) = \prod_{j=1}^s Q_j(\xi^{(j)}), \quad Q_j(\xi^{(j)}) = \prod_{i=1}^m q_i^{(j)}(\xi^{(j)}). \tag{10} \]

Note that block D in RML procedure is similar to its counterpart in ML procedure, in the modification making no hypotheses about the nature of data.

By defining in some sense appropriate PDFs of the noises, it is possible to generate the random vectors \( \xi^{(j)} \) with these PDFs, thereby constructing an ensemble \( \mathcal{E}(j | Q_j(\xi^{(j)})) \) of the random vectors \( \xi^{(j)} \).

b) Parameterized models block (PM): The parameters \( a \in \mathbb{R}^d \) of PM are random with interval-type independent components:

\[
a_i \in \mathcal{A}_i = [a_i^-, a_i^+], \quad i = 1, r; \nonumber \]

\[
a \in \mathcal{A} = [a^-, a^+]. \tag{11} \]

The probabilistic properties of the parameters are characterized by the probability density functions (PDFs) \( p_i(a_i) \). By assumption, these functions possess continuous derivatives. Due to independence, the joint PDF takes the form

\[
P(a) = \prod_{i=1}^r p_i(a_i). \tag{12} \]

Generally, the input and output data have a dynamic relationship. In other words, the output data observed at moment \( j \) depend on the input data observed on some historical interval \( j - g, j - g + 1, \ldots, j \).

In mathematical terms, this relationship is described by a nonrandom vector functional \( \hat{\Omega}(\hat{X}^{(j)} | a, P(a)) \) with random parameters \( a \). For each observation \( j \), the input array \( (a \text{ matrix } \hat{X}^{(j)}) \) consists of \( g \) column vectors \( x(j - g), x(j - g + 1), \ldots, x(j) \). The model with the above-mentioned properties will be called the random\(\text{ized parameterized model (RPM).} \]

Consequently, the model output at observation \( j \) represents an ensemble \( \hat{Y}(j | P(a)) \) of the random vectors \( \hat{y}(j | P(a)) \) relating to the input data and random parameters through the vector functional \( \hat{\Omega}(\hat{X}^{(j)} | a, P(a)) \), i.e.,

\[
\hat{Y}(j | P(a)) = \hat{\Omega}(\hat{X}^{(j)} | a, P(a)), \quad j = 1, s. \tag{13} \]

The errors in the output data are modeled by an ensemble \( \mathcal{E}(j | Q_j(\xi^{(j)})) \) of the random vectors \( \xi^{(j)} \) with the PDF \( Q_j(\xi^{(j)}) \), which is added to the ensemble of the RPM output:

\[
\hat{\mathcal{V}}(j | P(a), Q_j(\xi^{(j)})) = \hat{\mathcal{Y}}(j | P(a)) + \mathcal{E}(j | Q_j(\xi^{(j)})), \quad j = 1, s. \tag{14} \]

Disposing of the PDF \( P(a), Q_j(\xi^{(j)}) \), it is possible to generate the ensemble for \( j = 1, s \), e.g., to perform randomized forecasting or to predict different numerical characteristics using well-known methods of mathematical statistics.

Consider the ensemble \( \mathcal{V}(j | P(a), Q_j(\xi^{(j)})) \); actually, it consists of an array of the random vectors

\[
\mathcal{V}(j | P(a), Q_j(\xi^{(j)})) = \hat{\mathcal{Y}}(j | P(a)) + \xi(j | Q_j(\xi^{(j)})), \quad j = 1, s. \tag{15} \]

Define a vector \( M_k^{(j)}(j | P(a), Q_j(\xi^{(j)})) \) using the k-th moments of the components of the vector \( \mathcal{V}(j | P(a), Q_j(\xi^{(j)})) \):

\[
M_k^{(j)}(j | P(a), Q_j(\xi^{(j)})) = \mathcal{M}_P(\hat{\mathcal{Y}}^{(j)} | P(a)) + \mathcal{M}_Q(\xi^{(j)} | Q_j(\xi^{(j)})), \quad i = 1, m, j = 1, s. \tag{16} \]

Similarly, define a vector \( m_k^{(j)}(j | P(a), Q_j(\xi^{(j)})) \) using the k-means of the components of the vector \( \mathcal{V}(j | P(a), Q_j(\xi^{(j)})) \):

\[
m_k^{(j)}(j | P(a), Q_j(\xi^{(j)})) = \mathcal{M}_P(\hat{\mathcal{Y}}^{(j)} | P(a)) + \mathcal{M}_Q(\xi^{(j)} | Q_j(\xi^{(j)})) \right)^{1/k}, \quad i = 1, m, j = 1, s. \tag{17} \]

The k-mean vector \( m_k^{(j)}(j | P(a), Q_j(\xi^{(j)})) \) will be applied as a numerical characteristic of the ensemble \( \mathcal{V}(j | P(a), Q_j(\xi^{(j)})) \).

c) Block of RML algorithms: This block in RML procedure has fundamental differences from ML-A. For learning quality assessment, here we introduce the information entropy functional [28] defined on the PDF \( P(a) \) of the random parameters of RPM and on the vector PDF \( Q(\xi) = \{Q_1(\xi^{(1)}), \ldots, Q_s(\xi^{(s)})\} \) of the noises as follows:

\[
\mathcal{H}[\mathcal{P}(a), Q(\xi)] = - \int_A P(a) \ln \frac{P(a)}{P^0(a)} \, da - \sum_{j=1}^s \int_{\Xi_j} Q_j(\xi^{(j)}) \ln \frac{Q_j(\xi^{(j)})}{Q^j(\xi^{(j)})} \, d\xi^{(j)}, \tag{18} \]

where \( P^0(a), Q^j(\xi^{(j)}) \) denote the prior PDFs of the parameters and noises, respectively.

Such an approach proceeds from at least two considerations. The first is associated with likelihood generalization, i.e., transition from the likelihood function to the likelihood functional and, subsequently, to the informational entropy functional. This transformation will be discussed in the forthcoming section.

\[1\text{In particular, option transactions employ the mean values of financial tools having power dependence on random parameters [27].} \]
The second consideration concerns the methodological interpretations of the notion of informational entropy as a measure of uncertainty. Its maximization yields best solutions under maximum uncertainty. This logical chain was first declared in [29] (see also [30]–[34]). Informational entropy characterizes uncertainty connected with the random parameters of RPM and, moreover, with the measurement noises; the last property guarantees best estimates under maximum uncertain noises (in entropy units). Therefore, the PDF estimates obtained by informational entropy maximization can be interpreted as robust ones\(^2\).

Recall that the PDFs \(P_1, Q_1(\xi)\) are continuously differentiable and normal, i.e.,

\[
\int_P P(\mathbf{a}) d\mathbf{a} = 1, \quad \int_{Q_j} Q_j(\xi(j)) d\xi(j) = 1, \quad j = 1, \ldots, s. \tag{19}
\]

Introduce the notion of empirical balances stating that the \(k\)-means of the observable RPM output are balanced with the output data — the vectors \(y(j), \quad j = 1, \ldots, s\). The empirical balance conditions are described by the following system of equalities:

\[
\mathbf{m}^{(k)}(j | P_1, Q_1(\xi(j))) = y(j), \quad j = 1, \ldots, s. \tag{20}
\]

Here the vector \(\mathbf{m}^{(k)}\) contains the components (17). For the existing data array, equalities (20) define the set \(B_s\) of admissible PDFs:

\[
B_s = \mathcal{P} \cap \bigcap Q \cap \mathcal{D}_s, \tag{21}
\]

where \(\mathcal{P}\) is the set of continuously differentiable normal functions (the left equality in (19)); \(Q = \bigcap_{j=1}^s Q_j\), where \(Q_j\) indicates the set of continuously differentiable normal functions (the right equality in (19)); and finally, \(\mathcal{D}_s\) means the set of continuously differentiable functions \(P, Q_1, \ldots, Q_s\), satisfying the empirical balances (20) for the existing output data array.

And so, the \textbf{RML algorithm} is written in the form

\[
\mathcal{H}[P_1, Q_1] \Rightarrow \text{max}, \tag{22}
\]

subject to the condition

\[
(P_1, Q_1) \in B_s. \tag{23}
\]

This is a functional entropy-linear programming problem defined on the set of continuously differentiable functions satisfying the normality conditions (19) and the empirical balances (20, 23).

Direct comparison of the probabilistic ML algorithms (6, 8) and the \textbf{RML} algorithms (22, 23) shows that they are based on the hypothesis about the random origin (random nature) of the data and model parameters. However, the “structures” developed within each of them possess fundamental differences and procedural analogies.

In ML procedure, the random nature of the data and parameters implies the \textit{existence} of a corresponding PDF \(P(Z, \mathbf{a})\). According to the Bayes concept [30], the function \(P(Z, \mathbf{a})\) exists in two modifications, namely, the prior PDF \(P_{ap}(Z, \mathbf{a})\) and the posterior PDF \(P_{ap}(Z | \mathbf{a})\), which are separated by a trial (experiment) for knowledge accumulation. Here \(Z = (X, Y)\) denotes the arrays of input and output data.

Assume that the data before and after trial remain the same. Then the above-mentioned PDFs are functions of the random parameters \(\mathbf{a}\), i.e., \(P_{ap}(\mathbf{a}) = P_{ap}(Z, \mathbf{a})\) and \(P_{ap}(\mathbf{a} | Z) = P_{ap}(Z | \mathbf{a})\).

For fixed data \(Z\), consider the logarithmic likelihood ratio [25], [26] in the form

\[
L_Z(\mathbf{a}) = \ln \frac{P_{ap}(\mathbf{a})}{P_{ap}(Z | \mathbf{a})}, \tag{24}
\]

and define its expectation as

\[
L_Z[\mathbb{Z}_p(\mathbf{a})] = \int_A P_{ap}(\mathbf{a}) \ln \frac{P_{ap}(\mathbf{a})}{P_{ap}(Z | \mathbf{a})} d\mathbf{a}. \tag{25}
\]

The expectation \(L_Z(\mathbb{Z}_p(\mathbf{a}))\) is a functional of the PDF \(P_{ap}(\mathbf{a})\) under a fixed data array. Clearly, taken with the minus sign, it represents the informational entropy functional [28], [36], [37] for a fixed data array \(Z\):

\[
\mathcal{H}[P_{ap}(\mathbf{a})] = - \int_A P_{ap}(\mathbf{a}) \ln \frac{P_{ap}(\mathbf{a})}{P_{ap}(Z | \mathbf{a})} d\mathbf{a}. \tag{26}
\]

Actually, RML procedure performs maximization of this functional subject to the empirical balance conditions over the PDF belonging to the class of continuously differentiable functions.

V. \textbf{RANDOMIZED DYNAMICAL MODEL IN RML Procedure}

An important component of the \textbf{RML} algorithm (22) that is connected with the empirical balances comes to formation of the set \(B_s\) (18 - 21). The structure of this set depends on the \(k\)-mean empirical balances and the chosen RPM.

Here we will use the description of the set \(B_s\) induced by the system of 1-mean empirical balances. As mentioned earlier, the choice of RPM is informal and depends substantially on the problem specifics. At the same time, it is possible to state some formal considerations regarding the functional \(\hat{\mathcal{H}}(\hat{X}_{\hat{\mathcal{Q}}} | \hat{\mathcal{X}}, P_{ap}(\mathbf{a}))\) (13) that restrict the search domain of an appropriate RPM.

Whenever the assumption about \textit{continuity} of the functional \(\hat{\mathcal{H}}\) defined on the set of continuously differentiable functions is true, one can adopt its general form ( [38], [39]) as a series of power functionals. The book [40] extended this result to the continuous functionals defined on the set of piecewise constant functions. Of course, in applications such a representation contains a finite number \(R\) of terms.

Let the RPM be described by the following functional \textit{monomial} of degree \(R\):

\[
\hat{X}_{\hat{\mathcal{Q}}}^{(j)} = \sum_{h=1}^R \sum_{(k_1, \ldots, k_h) = 0}^0 A^{(k_1, \ldots, k_h)} x^{(j-k_1, \ldots, j-k_h)}. \tag{27}
\]

In this expression, the components of the vector \(x^{(j-k_1, \ldots, j-k_h)}\) are the lexicographically ordered \(h\)-products for the components of the vectors \(x^{(j-k_1)}, \ldots, x^{(j-k_h)}\).
Denote by \( n_h \) the dimension of this vector. The matrices \( A^{(k_1,\ldots,k_h)} \) have the form
\[
A^{(k_1,\ldots,k_h)} = \begin{pmatrix}
a_{1,1}^{(k_1,\ldots,k_h)} & \cdots & a_{1,n_h}^{(k_1,\ldots,k_h)} \\
\vdots & \ddots & \vdots \\
a_{m,1}^{(k_1,\ldots,k_h)} & \cdots & a_{m,n_h}^{(k_1,\ldots,k_h)}
\end{pmatrix},
\]
(28)
These matrices contain random independent interval-type elements of the form
\[
a_{i,s}^{(k_1,\ldots,k_h)} \in \mathcal{A}_{i,s}^{(k_1,\ldots,k_h)} = \left\{ (-) a_{i,s}^{(k_1,\ldots,k_h)}, (+) a_{i,s}^{(k_1,\ldots,k_h)} \right\};
\]
(29)
Designate by
\[
\mathcal{A}_h = \{ A^{(k_1,\ldots,k_h)} \mid (k_1,\ldots,k_h) = 0 \}
\]
the set of the matrices (28). The elements of this set possess values from the interval
\[
\mathcal{A}_h = \bigcup_{(k_1,\ldots,k_h)=0} \mathcal{A}^{(k_1,\ldots,k_h)}.
\]
(30)
On the above intervals, there exists the PDF
\[
P_h(N_h) = \prod_{(k_1,\ldots,k_h)=0} P_h(A^{(k_1,\ldots,k_h)}),
\]
(31)
\[
P_{i,s}^{(k_1,\ldots,k_h)}(A^{(k_1,\ldots,k_h)}) = \prod_{i=1}^{m} \prod_{s_h=1}^{n_h} P_{i,s_h}^{(k_1,\ldots,k_h)}(a_{i,s}^{(k_1,\ldots,k_h)}),
\]
(32)
The last expressions follow from independence of the matrices (28) and their elements.

According to (15) and (27), the ensemble of the observable RPM output consists of the random vectors
\[
v^{(j)} = \sum_{h=1}^{R} v^{(j)}_h(N_h) + \xi^{(j)}, \quad j = 1,\ldots,n.
\]
(33)
where
\[
v^{(j)}_h(N_h) = \sum_{(k_1,\ldots,k_h)=0} A^{(k_1,\ldots,k_h)} x^{j-k_1,\ldots,k_h-j}
\]
(34)
Recall that the noise vectors \( \xi^{(j)} j = 1,\ldots,n \) are random independent with the intervals (9) and the PDF \( Q_j(\xi^{(j)}) \), \( j = 1,\ldots,n \).

**Definition.** For the RPM (27), the RML algorithm (22, 23) has the form
\[
\mathcal{H}(P(A), Q(\xi)) = -\sum_{h=1}^{R} \int_{\mathcal{A}_h} P_h(N_h) \ln \frac{P_h(N_h)}{P_h(N_{h_0})} dN_h - \\
- \sum_{j=1}^{s} \int_{\Xi_j} Q_j(\xi^{(j)}) \ln \frac{Q_j(\xi^{(j)})}{Q_{j0}(\xi^{(j)})} d\xi^{(j)} \Rightarrow \max (35)
\]
subject to
- the normalization conditions of the PDFs
\[
\int_{\mathcal{A}_h} P_h(N_h) dN_h = 1, \quad h = 1,\ldots,R;
\]
(36)
- and the 1-mean empirical balances
\[
\sum_{h=1}^{R} \int_{\mathcal{A}_h} P_h(N_h) v^{(j)}_h(N_h) dN_h + \sum_{j=1}^{s} \int_{\Xi_j} Q_j(\xi^{(j)}) \xi^{(j)} d\xi^{(j)} = y^{(j)},
\]
(37)
In the stated definition, \( P(A) = \{ P_1(A_1),\ldots,P_R(A_R) \} \) and \( Q(\xi) = \{ Q_1(\xi),\ldots,Q_s(\xi) \} \), where the PDFs entering these sets are continuously differentiable. The functions \( P_{j0}(\xi) \) and \( Q_{j0}(\xi) \), \( h = 1,\ldots,R; j = 1,\ldots,s \) are the prior PDFs. If no information is available about the prior PDFs, they are 1.

VI. OPTIMALITY CONDITIONS OF RML ALGORITHM.

**Structure of entropy-optimal PDFs**

The RML algorithm (35–37) is formulated in terms of a functional, the Lyapunov-type optimization problems [41] which belongs to the Lyapunov-type optimization problems [42–44] (all components are described by integral functionals). For such problems, optimality conditions can be obtained using the Lagrange functional and multipliers. Since the RML algorithm involves continuously differentiable PDFs, the variation of the Lagrange functional can be defined via the Gateaux derivatives [45].

Introduce the following notation:
- vector \( w(P(A)) \) with the components
\[
w_h(P_h(N_h)) = 1 - \int_{\mathcal{A}_h} P_h(N_h) dN_h, \quad h = 1,\ldots,R;
\]
(38)
- vector \( q(Q(\xi)) \) with the components
\[
q_j(Q_j(\xi^{(j)})) = 1 - \int_{\Xi_j} Q_j(\xi^{(j)}) d\xi^{(j)}, \quad j = 1,\ldots,n;
\]
(39)
- vectors \( z^{(j)}(P(A), Q(\xi)) \), \( j = 1,\ldots,n \) with the components
\[
z^{(j)}_i(P(A), Q(\xi)) = y^{(j)}_i - \sum_{h=1}^{R} \int_{\mathcal{A}_h} P_h(N_h) v^{(j)}_{h,i}(N_h) dN_h + \int_{\Xi_j} Q_j(\xi^{(j)}) \xi^{(j)}_i d\xi^{(j)}, \quad i = 1,\ldots,m;
\]
(40)
- vectors of the Lagrange multipliers
\[
\bar{\nu} = \{ \nu_1,\ldots,\nu_s \}, \quad \bar{\theta} = \{ \theta_{1j},\ldots,\theta_{sj} \}, \quad j = 1,\ldots,s.
\]
(41)
Using this system of symbols, define the Lagrange functional
\[
\mathcal{L}[P(A), Q(\xi)] = \mathcal{H}[P(A), Q(\xi)] + \langle \bar{\nu}, w(P(A)) \rangle + \langle \bar{\theta}, q(Q(\xi)) \rangle + \sum_{j=1}^{s} (\bar{\theta}^{(j)}, z^{(j)}(P(A), Q(\xi))).
\]
(42)
Here \( (\cdot, \cdot) \) is the scalar product of vectors \( \cdot \) and \( \cdot \). Recall that \( P(\mathcal{A}) = \{ P_1(\mathcal{A}), \ldots, P_R(\mathcal{A}) \} \) and \( Q(\xi) = \{ Q_1(\xi)(\xi), \ldots, Q_s(\xi)(\xi) \} \), all PDFs in these sets belong to \( \mathcal{C}^1 \), the class of continuously differentiable functions.

In \( \mathcal{C}^1 \), consider a set of arbitrary functions
\[
\{ t_1(\mathcal{A}), \ldots, t_R(\mathcal{A}) \} \quad \{ u_1(\xi)(\xi), \ldots, u_s(\xi)(\xi) \}. \tag{43}
\]
Express the PDF from (42) in the form
\[
P_h(\mathcal{A}) = P_h^0(\mathcal{A}) + \alpha h t_1(\mathcal{A}) + \beta u_1(\xi)(\xi), \quad h = 1/R, \tag{44}
\]
where \( P_h^0(\mathcal{A}), Q_h^0(\xi)(\xi) \) are the PDF that solve the problem (35 - 37), \( \alpha = (\alpha_1, \ldots, \alpha_R) \) and \( \beta = (\beta_1, \ldots, \beta_s) \) are vectors with scalar components.

Substituting (44) into (42) gives the parameterized representation of the Lagrange functional:
\[
\mathcal{L}[P(\mathcal{A}), Q(\xi)] = \mathcal{L}[\alpha, \beta]. \tag{45}
\]
The stationarity conditions of the functional (45) in terms of the Gateaux derivatives have the form
\[
\frac{\partial \mathcal{L}[\alpha, \beta]}{\partial \alpha} \bigg|_{\alpha = 0} = 0, \quad h = 1/R, \tag{46}
\]
A chain of simple (yet, arid) transformations yields the following integral stationarity conditions:
\[
\int_{\mathcal{A}} t_h(\mathcal{A}) \left( \ln \frac{P_h^0(\mathcal{A})}{P_h(\mathcal{A})} + 1 + \mu_h + \sum_{j=1}^s (\theta^{(j)}), v^{(j)h}(\mathcal{A}) \right) d\mathcal{A} = 0, \quad h = 1/R, \tag{47}
\]
\[
\int_{\mathcal{B}} u_j(\xi)(\xi) \left( \ln \frac{Q_j^0(\xi)(\xi)}{Q_j(\xi)(\xi)} + 1 + \nu_j + \sum_{j=1}^s (\theta^{(j)}, \xi^{(j)}) \right) d\xi = 0, \quad j = 1/s. \tag{48}
\]
For these integral equalities to hold for any functions \( (t_1(\mathcal{A}), \ldots, t_R(\mathcal{A})) \) & \( (u_1(\xi)(\xi), \ldots, u_s(\xi)(\xi)) \) \( \in \mathcal{C}^1 \), it is necessary that the corresponding integrands vanish; by-turn, the latter form the stationarity conditions for the Lagrange functional (42):
\[
\ln \frac{P_h^0(\mathcal{A})}{P_h(\mathcal{A})} + 1 + \mu_h + \sum_{j=1}^s (\theta^{(j)}), v^{(j)h}(\mathcal{A}) = 0, \quad (49)
\]
\[
\ln \frac{Q_j^0(\xi)(\xi)}{Q_j(\xi)(\xi)} + 1 + \nu_j + \sum_{j=1}^s (\theta^{(j)}, \xi^{(j)}) = 0, \quad (50)
\]
For these integral equalities to hold for any functions \( (t_1(\mathcal{A}), \ldots, t_R(\mathcal{A})) \) & \( (u_1(\xi)(\xi), \ldots, u_s(\xi)(\xi)) \) \( \in \mathcal{C}^1 \), it is necessary that the corresponding integrands vanish; by-turn, the latter form the stationarity conditions for the Lagrange functional (42):
\[
\ln \frac{P_h^0(\mathcal{A})}{P_h(\mathcal{A})} + 1 + \mu_h + \sum_{j=1}^s (\theta^{(j)}), v^{(j)h}(\mathcal{A}) = 0, \quad (49)
\]
\[
\ln \frac{Q_j^0(\xi)(\xi)}{Q_j(\xi)(\xi)} + 1 + \nu_j + \sum_{j=1}^s (\theta^{(j)}, \xi^{(j)}) = 0, \quad (50)
\]
In fact, these equations result from the empirical balances (37) by substituting the PDFs from (49).

If there exists no prior information, i.e., \( P_h^0(\mathcal{A}) = \text{const}, h = 1/R \) and \( Q_j^0(\xi)(\xi) = \text{const}, j = 1/s \), then the equalities defining the entropy-optimal PDF (49, 50) acquire the simplified form
\[
P_h^*(\mathcal{A}) = \frac{\exp \left[ - \sum_{j=1}^s (\theta^{(j)}), v^{(j)h}(\mathcal{A}) \right]}{P_h(\theta)}, \tag{51}
\]
\[
Q_j^*(\xi)(\xi) = \frac{\exp \left[ - \sum_{j=1}^s (\theta^{(j)}, \xi^{(j)}) \right]}{Q_j(\theta)}, \tag{52}
\]
where
\[
P_h^*(\mathcal{A}) = P_h(\mathcal{A}) \exp \left[ - \sum_{j=1}^s (\theta^{(j)}, v^{(j)h}(\mathcal{A})) \right], \tag{53}
\]
\[
Q_j^*(\xi)(\xi) = Q_j(\xi)(\xi) \exp \left[ - \sum_{j=1}^s (\theta^{(j)}, \xi^{(j)}) \right], \tag{54}
\]
The entropic components of the RPM parameters and measurement noises are defined by
\[
\tilde{P}_h(\theta) = \int_{\mathcal{A}} \exp \left[ - \sum_{j=1}^s (\theta^{(j)}, v^{(j)h}(\mathcal{A})) \right] d\mathcal{A}, \tag{55}
\]
\[
\tilde{Q}_j(\theta) = \int_{\mathcal{B}} \exp \left[ - \sum_{j=1}^s (\theta^{(j)}, \xi^{(j)}) \right] d\xi, \tag{56}
\]
In this case, equations (51) do not incorporate the prior PDFs, too:

\[
\begin{align*}
\sum_{h=1}^{R} P^{-1}_h(\theta) \int_{A_h} & \exp \left[ -s \sum_{j=1}^{n} (\theta(y_{i,j}^{(j)}, v_{h,v_{j,h}}^{(j),h})) \right] d\theta_h + \\
+ & \sum_{j=1}^{n} \tilde{Q}_j^{-1}(\theta) \int_{\Xi_j} \exp \left[ -s \sum_{j=1}^{n} (\theta(y_{i,j}^{(j)}, \xi_{j}^{(j)})) \right] d\xi = y^{(j)},
\end{align*}
\]

Equations (49, 51) and (52, 53) represent a special class of nonlinear equations with integral components defining the functional dependencies of their left-hand sides on the Lagrange multipliers \(\theta\). Within the framework of the RPM in the form of the functional Volterra monomial (27), the above integral components admit no analytic definition. Here the alternative is to perform numerical calculation, e.g., based on Monte Carlo trials [41].

VII. APPLICATIONS

A. “2-soft” classification of texts

In the sequel, “n-soft” classification implies the allocation of objects among \(n\) classes with probability defined by application of the RML procedure.

Consider a randomized linear classifier for 2 classes. Let there exist two collections of text documents, namely, a learning collection \(E = \{e_1, \ldots, e_m\}\) and a classification (test) collection \(T = \{t_1, \ldots, t_s\}\). The documents in the learning collection are marked by their belonging to class 1 or to class 2. The documents in the test collection have no marking.

The documents in both collections are described by the vectors containing the weights of the words (terms) in a given document. Weight admits many interpretations and quantitative characterizations that are not discussed in the paper. The sets of vectors in corresponding collections are somehow ordered, i.e.,

\[
\mathcal{E} = \{e^{(1)}, \ldots, e^{(m)}\}, \quad \mathcal{T} = \{t^{(1)}, \ldots, t^{(n)}\}. \tag{55}
\]

By assumption, these vectors have a same dimension, that is, \((e^{(j)}, t^{(j)}) \in R^n\), where \(R^n\) denotes the attribute space (in this case, weights).

1) Learning stage: Recall that the documents are marked at this stage. And so, there exists a learning sequence representing a sequence of pairs \(\mathcal{O} = \{(1, 1), (2, 1), \ldots, (m, 1)\}\) of length \(m\), where the first element is document number and the second element gives its class. Transform it into a sequence of numbers from the interval \([0, 1]\), where element position in the sequence stands for document number, while the numbers \([1/2, 1]\) and \([0, 1/2]\) mean document belonging to class 1 and class 2, respectively. Therefore, the “learning” vector \(y\) has values from the interval \([0, 1]\) as its components and dimension coinciding with the number \(m\) of documents in the collection:

\[
y = \{y^{(1)}, \ldots, y^{(m)}\}. \tag{56}
\]

The randomized model (decision rule) is defined by a random vector \(\hat{y}(a)\) that depends on the random parameters \(a\) of a single-layer neural network (see Fig. 2). The independent components of the vector \(\hat{y}(a)\), i.e.,

\[
\hat{g}^{(i)}(a) = \text{sign} \left( \langle e^{(i)}, a \rangle \right), \quad i = 1, m, \tag{57}
\]

are generated by the transformer

\[
\text{sign}(x) = \frac{1}{1 + \exp[-\alpha(x - \Delta)]}, \tag{58}
\]

with fixed parameters \(\alpha, \Delta\). This function has random argument, as the parameters \(a\) of the randomized model are random. The values of \(\text{sign}(x)\) from the interval \([1/2, 1]\) correspond to class 1, while the values from the open interval \([0, 1/2)\) to class 2.

The parameters \(a = \{a_1, \ldots, a_n\}\) in the randomized model (55) are independent interval-type:

\[
a_k \in \mathcal{A}_k = [a_k^-, a_k^+] \quad \text{for} \quad k = 1, n,
\]

\[
\mathcal{A} = \bigcup_{k=1}^{n} \mathcal{A}_k. \tag{59}
\]

There exists the joint probability density function

\[
P(a) = \prod_{k=1}^{n} p_k(a_k) \tag{60}
\]

on these interval sets, where \(p_k(a_k)\) indicate the PDFs for the components of the vector \(a\).

Since the parameters \(a\) are random with the PDF \(P(a)\), for each document with number \((i)\) we obtain an ensemble \(\hat{y}^{(i)}(a)\) of random values from the interval \([0, 1]\) (57). The mean values of the components are

\[
\mathcal{M}\{\hat{y}^{(i)}(a)\} = \int_{\mathcal{A}} P(a) \text{sign} \left( \langle e^{(i)}, a \rangle \right) da. \tag{61}
\]

Thus, the “2-soft” classification problem in terms of RML is stated as

\[
\mathcal{H}[P(a)] = - \int_{\mathcal{A}} P(a) \ln P(a) da \Rightarrow \max, \tag{62}
\]

subject to the conditions

\[
\int_{\mathcal{A}} P(a) da = 1, \tag{63}
\]

\[
\int_{\mathcal{A}} P(a) \text{sign} \left( \langle e^{(i)}, a \rangle \right) da = y^{(i)}, \quad i = 1, m. \tag{64}
\]
TABLE I

<table>
<thead>
<tr>
<th>i</th>
<th>$e_r^{(1)}$</th>
<th>$e_r^{(2)}$</th>
<th>$e_r^{(3)}$</th>
<th>$e_r^{(4)}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.11</td>
<td>0.75</td>
<td>0.08</td>
<td>0.21</td>
</tr>
<tr>
<td>2</td>
<td>0.91</td>
<td>0.65</td>
<td>0.11</td>
<td>0.81</td>
</tr>
<tr>
<td>3</td>
<td>0.57</td>
<td>0.17</td>
<td>0.31</td>
<td>0.91</td>
</tr>
</tbody>
</table>

Introduce the Lagrange multipliers $\theta = \{\theta_1, \ldots, \theta_m\}$ for the constraints (64). According to (52), the solution of this problem is adapted to the model (57) acquires the form

$$P^*(\mathbf{a}) = \frac{W^*(\mathbf{a})}{P(\theta)},$$

(65)

where

$$W^*(\mathbf{a}) = \exp \left( -\langle \theta, \hat{y}(\mathbf{a}) \rangle \right),$$

(66)

$$P(\theta) = \int_A \exp \left[ -\langle \theta, \hat{y}(\mathbf{a}) \rangle \right] d\mathbf{a},$$

(67)

and the vector $\hat{y}(\mathbf{a})$ consists of the components (55). Lagrange multipliers $\theta$ are defined by the system of balance equations (64) of the form

$$P^{-1}(\theta) \int_A \exp \left[ -\langle \theta, \hat{y}(\mathbf{a}) \rangle \right] \hat{y}^{(i)}(\mathbf{a}) d\mathbf{a} = y^{(i)}, \quad i = 1, \ldots, m.$$  

(68)

Interestingly, the dimension of this system coincides with the number of documents in the learning collection. Therefore, in the sense of computing effort, the randomized solution procedure for the classification problem is more efficient under limited-volume learning collections. Consider two examples demonstrating the main idea of “2-soft” classification of texts without discussion of computational aspects.

a) Example 1 (Learning): The dimension of RML-algorithm is 4, the learning collection consists of three documents each described by four weights, see Table I.

The randomized model (58) has the parameters $\alpha = 1.0$ and $\Delta = 0$. The “learner” responses are $\mathbf{y} = \{0.18; 0.81; 0.43\}$ ($y_i < 0.5$ corresponds to class 2, $y_i \geq 0.5$ to class 1). The Lagrange multipliers for the entropy-optimal PDF (65) are $\theta = \{3.2807; -3.5127; 1.6373\}$. The parameters belong to the ranges $a_i \in [-10, 10]$, $i = 1, \ldots, 4$. For this learning collection, the entropy-optimal function $W^*(\mathbf{a})$ (66) takes the form

$$W^*(\mathbf{a}) = \exp \left( -\sum_{i=1}^{4} \theta_i y_i(\mathbf{a}) \right),$$

(69)

$$y_i(\mathbf{a}) = \left( 1 + \exp\left(-\sum_{k=1}^{4} e(i)_{k, a_k} \right)^{-1} \right).$$

(70)

Figure 3 shows the two-dimensional section of the function (69) under $a_3 = 0.5; a_4 = 0.5$.

b) Example 2 (Learning): The dimension of RML-algorithm is 2, the learning collection consists of three documents each described by two weights (see the first two columns in Table I). The documents from the learning collection are characterized by the two-element vectors. The values of the parameters $\alpha, \Delta$ and the intervals for the random parameters $\mathbf{a}$ are the same as in Example 1. The Lagrange multipliers for the entropy-optimal PDF (65) are $\theta = \{9.6316; -18.5996; 16.7502\}$. For this learning collection, the entropy-optimal function $W^*(\mathbf{a})$ (66) takes the form

$$W^*(\mathbf{a}) = \exp \left( -\sum_{i=1}^{2} \theta_i y_i(\mathbf{a}) \right),$$

(70)

$$y_i(\mathbf{a}) = \left( 1 + \exp\left(-\sum_{k=1}^{2} e(i)_{k, a_k} \right)^{-1} \right).$$

(70)

Figure 4 illustrates the function $W^*(\mathbf{a})$ (70).

2) Classification stage: This stage employs the collection $\mathcal{T} = \{t_1, \ldots, t_s\}$, where each element is characterized by a vector $t^{(j)} \in \mathbb{R}^a$. Consider classification procedure for an arbitrary document $t^{(j)}$. 

Step 1-i. Generate an ensemble $\mathcal{Y}^{(i)}$ of the randomized model output (decision rules) (57) with the function $P^*(\mathbf{a})$.
(65). The ensemble contains $N$ random values from the interval $[0, 1]$.

**Step 2-i.** If a random value from this ensemble exceeds $1/2$, then document $t^{(i)}$ is assigned class 1; otherwise, class 2.

**Step 3-i**. Suppose that $N_1$ values are assigned class 1 and $N_2$ values class 2. Since the number of trials $N$ is sufficiently large, the quantities $p_1^{(i)} = N_1/N$ and $p_2^{(i)} = N_2/N$ yield the empirical probabilities of assigning appropriate classes to document $t^{(i)}$.

By repeating steps 2-i, 3-i for the whole collection $\mathcal{T}$, we obtain the probability distribution of assigning class 1 or 2 to the document.

**a) Example 1 (Classification).** For classification we use a collection of 500 documents represented by an array of the four-dimensional random vectors $t^{(i)}$, $i = 1, 500$ with independent components obeying the uniform distribution on the interval $[0,1]$. For each element of this sample, generate the random parameters of the model (57) according to the PDF (65) ($N = 1000$) using the acceptance-rejection method [36] and calculate its output. Perform steps 2-i, 3-i.

Figure 5a,b demonstrates the empirical probabilities $p_1^{(i)}, p_2^{(i)}$ of assigning class 1 and 2 to document $t_i$. For different documents their assigning probabilities vary from 15 to 85 percent.

**b) Example 2 (Classification).** All parameters are same as in Example 2 (Learning). Figure 6a,b shows the empirical probabilities $p_1^{(i)}, p_2^{(i)}$ of assigning class 1 and 2 to document $t_i$. In this case, for different documents their assigning probabilities vary from approx. 20 to 80 percent.

**B. RML algorithms for dynamic regression problems**

The general form of the randomized dynamic model (RDM) for dynamic regression problems is given by (13), and it is possible to apply the general RML procedure stated in section 3. Consider this procedure in detail to restore the characteristics of the randomized World population dynamics model using it as a forecasting tool [46].

Let the World population evolve according to the discrete-form exponential randomized model of population dynamics with measurement errors

$$v[ih] = E_i(b,m|E_0) + \xi[ih], \quad i \in [0, I],$$

(71)

with the function

$$E_i(r, u_r | E_0) = E_0 \exp[r + u_r i]h, \quad i \in [0, I].$$

(72)

Here $h = 5$ [years] is the measurement interval, $E_0$ specifies the World population at the beginning of the observation interval, $I$ indicates the length of the measurement interval (an integer number), $r$ means reproduction rate (difference between fertility and mortality flows), $u_r$ is the velocity of its changing.
The measurement errors are modeled by a random vector 
\( \xi = \{\xi[0], \ldots, \xi[7]\} \) with independent interval-type components and a PDF \( Q(\xi) \) defined on the set

\[ \Xi = \bigcup_{j=0}^{I} \Xi_{j}, \quad \Xi_{j} = [\xi_{j}, \xi_{j}^{+}], \quad (73) \]

where \( I + 1 \) gives the number of World population measurements on a corresponding interval. By assumption, the intervals are same for all measurements: \([-0.5; 0.5] \) billion people. And so,

\[ Q(\xi) = \prod_{j=0}^{I} q_{j}(\xi[jh]). \quad (74) \]

A common approach in World population forecasting is to consider model parameters like fertility and mortality rates as constant on definite intervals, analyzing different scenario corresponding to the period from 1960 to 1995 with step size.

1) Learning stage: Find the entropy-optimal PDFs of the model parameters and noises for the retrospective data corresponding to the period from 1960 to 1995 with step size.

The RML algorithm (14-16) yields the following entropy-optimal PDFs:

- model parameters

\[ P^{*}(r, u_{r}) = \frac{1}{R^{*}(\theta)} \prod_{i=0}^{7} p_{i}^{*}(r, u_{r}|\theta_{i}), \]

\[ p_{i}^{*}(r, u_{r}|\theta_{i}) = \exp(-\theta_{i}E_{i}(r, u_{r}|E_{\textrm{real}}^{\textrm{est}}[0])); \quad (76) \]

- noise

\[ Q^{*}(\xi) = \frac{1}{Q^{*}(\theta)} \prod_{j=0}^{7} q_{j}^{*}(\xi[jh]|\theta_{j}), \]

\[ q_{j}^{*}(\xi[jh]|\theta_{j}) = \exp\left(-\theta_{j}\xi[jh]\right). \quad (77) \]

Assume that reproduction rate and velocity of its changing are random, taking values from the intervals \( \mathcal{I}_{r} \) and \( \mathcal{I}_{u_{r}} \) with a joint PDF \( P(r, u_{r}) \), where \( (r, u_{r}) \in \mathcal{I} = \mathcal{I}_{r} \cup \mathcal{I}_{u_{r}} \). In the sequel, \( \mathcal{I}_{r} = [-0.025; 0.075]; \mathcal{I}_{u_{r}} = [-0.002; 0.001] \). Selection of such intervals for model parameters assumes the possibility for both positive and negative trends of World population growth.

2) Testing stage: The model has been tested using the data from Table IV.

### Table II
WORLD POPULATION IN billion people

|------|------|------|------|------|------|------|------|

### Table III
LAGRANGE MULTIPLIERS

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Value</td>
<td>0.0000</td>
<td>0.3833</td>
<td>0.5984</td>
<td>0.5839</td>
<td>0.3802</td>
<td>0.4679</td>
<td>0.1812</td>
</tr>
</tbody>
</table>

Fig. 7. Joint PDF of \( r, u_{r} \) in interval \( \mathcal{I}_{r} \cup \mathcal{I}_{u_{r}} \).

Fig. 8. Ensemble of PDFs of noise \( \xi_{i}, i \in [0, 7] \).

where

\[ R(\theta|E_{\textrm{real}}^{\textrm{est}}[0]) = \int_{\mathcal{I}} \prod_{i=0}^{7} \exp\left(-\theta_{i}E_{i}(r, u_{r}|E_{\textrm{real}}^{\textrm{est}}[0])\right) dr du_{r} \quad (78) \]

and

\[ Q(\theta) = \prod_{j=0}^{7} \int_{\xi_{j}^{-}}^{\xi_{j}^{+}} \exp(-\theta_{j}\xi[jh])d\xi[jh] = \prod_{j=0}^{7} \frac{1}{\theta_{j}} \left(\exp(-\theta_{j}\xi_{j}^{+}) - \exp(-\theta_{j}\xi_{j}^{-})\right) \quad (79) \]

The Lagrange multipliers \( \theta \) are defined by solving the system of balance equations, see (17). Table III shows the Lagrange multipliers calculated for the above intervals.

The curves of the PDF \( P^{*}(r, u_{r}) \) of the model parameters and the PDFs \( q^{*}(\xi[ih]), i \in [0, 7] \) of the noise components are shown by Fig. 7–8.

2) Testing stage: The model has been tested using the data from Table IV.
The World population is evaluated by formula (71, 72), where \( r, u_r \) represent the random parameters with the PDF \( P^*(r, u_r) \) (76); \( \xi[ih] \) are the random noises with the PDFs \( q_i(\xi[ih]), i \in [0, 4] \) (77).

To generate an ensemble of the random variables (the model parameters \( r, u_r \) and the noises \( \xi[ih], i \in [0, 4] \), we have adopted the generalized two-dimensional modification of the Ulam-von Neumann method (acceptance-rejection method [36]). The size of the generated sample is \( k = 10^5 \). Figure 9 illustrates the resulting ensemble with the following notation: (1) are the ensemble-average trajectories of population dynamics; (2) is the real population dynamics; (3) is the population dynamics on the testing interval (1995–2015) according to the UN forecast made at 1985. The relative mean-square deviation between the real trajectory and the ensemble-average one is 0.3%. The relative mean-square deviation for the UN forecast is 0.8%.

### VIII. Conclusion

This paper has proposed a new machine learning concept called randomized machine learning, which proceeds from the following components:

1. Real data containing measurement errors in the form of additive sequences of random independent vectors;
2. Input-output models with random parameters described by entropy-optimal probability density functions that are constructed for available data arrays;

The RML procedure has been applied to text classification and dynamic regression problems. Optimal randomization of models in these problems yields complete probabilistic characteristics for the results of classification and forecasting.

### References


