BUILDING OPTIMIZATION MODELS FROM DATA FOR THE INTELLIGENT CONTROL SYSTEMS [1]

Donskoy V. I.
V.I. Vernadsky Crimean Federal University
Department of Mathematics and Computer Science Theory
donskoy@crimea.edu

ABSTRACT

In this paper it is shown, how can be synthesized models of the optimal control on the basis of samples or precedents. The proposed BOMD approach is based on empirical induction and directed to obtaining regularities in the form of empirical optimization models which are synthesized in analytical form. We follow the Kolmogorov idea about regularity as non-randomness. This allows us to estimate the probability of non-random model selection from the set of admissible models which are consistent to the sample or to given initial data. The proposed methods and algorithms can be applied to solve wide range of tasks of intelligent control, in particular, in Robotics.

KEY WORDS
Intelligent Control, Optimization Models, Synthesis from Data.

1. Introduction

Intelligent Control methods are classified as a rule in the following categories: Model-based methods, Knowledge-based methods, Fuzzy logic methods, Neural network methods, Hybrid methods, and other methods [4, 10, and 11]. Unlike to other approaches, Building Optimization Models from Data (BOMD) is directed to building or synthesis of optimization models by data and knowledge. These data and knowledge can be presented in various forms, mainly as experimental observations or samples and expert knowledge presented in the form of predicates and logical productions (rules). BOMD paradigm involves the construction of the optimization models consisting of objective functions and constraints in the explicit form. Because of this it is possible design the intelligent control systems which have the explanation ability – answer questions “Why” and “How” when they produce control actions.

The main distinctive property of BOMD compared to traditional mathematical modeling is in that available data and knowledge are incomplete. This incompleteness leads to ambiguity of possible solutions – synthesized models. The problems arising in the synthesis of optimization models due to the incompleteness of the initial information can be overcome by constantly learning and correcting of these models. In this sense we can talk about learning models.

As in traditional mathematical modeling, the choice of model largely depends on variables type: logical, integer, real or mixed. The use of predicates which maps states of external environment in binary variables allows one to work with mixed variables. Binarization can be considered as a special case of application of such predicates.

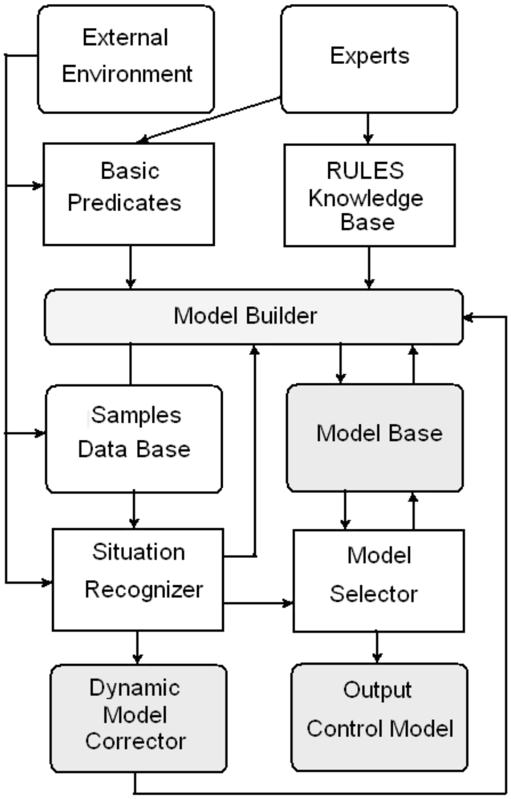


Figure 1. A system based the BOMD paradigm

The structure of the system that implements the BOMD principles is presented in Figure 1. Its main part is the subsystem of model building. Due to the nonstationarity of the external environment it is important provide the possibility of dynamic adjustment or correction of intelligent control models. Therefore, the Dynamic Model Corrector subsystem should contain a procedure for the estimation of obsolescence and forgetting data.

This paper mainly devoted to the ways of constructing objective functions and constraints from

[1] *This paper was prepared and sent to the IASTED International conference on Robotics and Control Systems (RCS- 2016) which unfortunately was cancelled*

experimental data which can be used for intelligent control systems developing.

If the optimization model has been synthesized, the control process is realised as shown on the Figure 2. The main element of the controller is the optimization model which is synthesized from data.

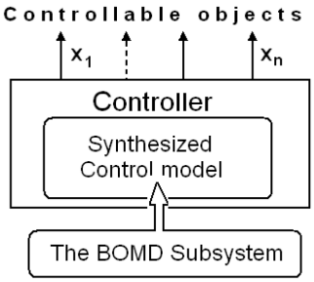


Figure 2. BOMD-Model based controller

2. Building Linear Optimization Model with Real Variables from Data

2.1 Introduction and preliminaries

Consider the real space $\mathbb{R}^n$ and the training sample set $T_l = \{X_k, y_k, \alpha_k\}_{k=1}^{l}$, where $X_k = (x_1^k, \dots, x_n^k) \in \mathbb{R}^n$ – vectors or points; y_k is a value of unknown function $f: \mathbb{R}^n \to \mathbb{R}$; $\alpha_k = 0$ if vector X_k is admissible solution of the optimization problem which needs to be build, otherwise Boolean value $\alpha_k = 1$. As additional initial information we suppose: *it is a priori known that the optimization model which is synthesized is linear*:

$$f(X) = \langle W, X \rangle = w_1 x_1 + \cdots + w_i x_i + \cdots + w_n x_n + w_0 \,,$$

where coefficients w_i and w_0 are unknown. A solution X is admissible iff

$$AX \le B,\ A = [a_{ji}]_{m \times n},\ X = \begin{bmatrix} x_1 \\ \dots \\ x_i \\ \dots \\ x_n \end{bmatrix},\ B = \begin{bmatrix} b_1 \\ \dots \\ b_j \\ \dots \\ b_m \end{bmatrix}.$$

Matrix A and vector B are unknown. So, we have only initial incomplete information for all X_k from the sample set T_l in the form

$$f(X_k) = y_k;$$
$$(\alpha_k = 0) \ \Rightarrow\ AX_k \le B;$$
$$(\alpha_k = 1) \ \Rightarrow\ AX_k > B;$$

We suppose without loss of generality that some of variables $x_1, \dots, x_n$ describe the environment and other of these variables describes the control system, including its control parameters. We also suppose that the initial information contained in the sample set T_l is exact, i.e. the formulation of the problem is deterministic. Therefore, if solving the problem will reduce to the contradiction indicating the nonlinearity, it is necessary either to revise the deliberate assumption of linearity of the model or to find errors in the initial information T_l. Further, we show how such contradictions are detected.

2.2 Building of the linear Constraints

We will use the classical Rosenblatt-Novikoff Linear Error-Correction Procedure (RNLCP):

$$\Lambda_0 \quad = \ (0, \dots, 0);$$

$$\Lambda_{t+1} = \begin{cases} \Lambda_t \,, & if\ \ (1)\ is\ true; \\ \Lambda_t + cX_t \,, & if\ \ (2)\ is\ true; \\ \Lambda_t - cX_t, & if\ \ (3)\ is\ true; \end{cases}$$

$$((\langle \Lambda_t, X_t \rangle > 0) \wedge (\mathrm{X_t} \in \mathrm{W_1}) \vee ((\langle \Lambda_t, X_t \rangle \le 0) \wedge (\mathrm{X_t} \in \mathrm{W_0}); \quad (1)$$
$$((\langle \Lambda_t, X_t \rangle \le 0) \wedge (X_t \in W_1); \quad (2)$$
$$((\langle \Lambda_t, X_t \rangle > 0) \wedge (X_t \in W_0). \quad (3)$$

The coefficient c is chosen from the interval $0 < c \le 1$; $W_0, W_1 \in \mathbb{R}^n$.

It's well known if $conv(W_0) \cap conv(W_1) = \emptyset$, where $conv(W_0)$ and $conv(W_1)$ are convex hulls of the sets $\mathrm{W_0}$ and $\mathrm{W_1}$, then exists such unit vector Λ^* and the real positive number ρ that $(\Lambda^*, X_t) < -\rho$ for any $X_t \in W_0$ and $(\Lambda^*, X_t) > \rho$ for any $X_t \in W_1$. Under this conditions, the RNLCP finds the hyperplane $(\Lambda, X) = 0$ separating the sets W_0 and W_1 after $k \le D^2/\rho^2$ steps, where

$$D = \sup_{X \in W_0 \cup W_1} \|X\|.$$

We suppose that $X_t = (x_1^t, \dots, x_{1n}^t, 1)$ is the vector extended by adding the additional component equal to one; so the equation of the hyperplane contains the constant term.

Denote T_0 the set of vectors X_k from the sample set T_l such that $\alpha_k = 0$ and T_1 – the set of vectors X_k such that $\alpha_k = 1$. If the assumption of linearity of the model is true, than the region of admissible solutions is a convex set. Then each point from the set T_1 can be separated by some hyperplane from all points from the set T_0. The idea on which building of linear constraints is based consists of finding the set of hyperplanes which separate together each point $X_k \in T_1$ from all points of the set T_0.

1. Initialize the set for memorizing hyperplanes which will be built: $\mathfrak{L} \coloneqq \emptyset$.
   $i \coloneqq 1$; $T_i \coloneqq T_1$.
2. Choose the point $X \in T_i$ which is closest to the set of points T_0.
3. Using the RNLCP, build the hyperplane $\mathcal{L}_i$ which separates the point X from all points of the set T_0.
4. Memorize this hyperplane: $\mathfrak{L} \coloneqq \mathfrak{L} \cup \{\mathcal{L}_i\}$.
5. Denote S_i the set the points from T_1 which are separated from all points of the set T_0 by the hyperplane $\mathcal{L}_i$ just as the point X.
   Reduce the set $T_i \coloneqq T_i \backslash S_i$.
6. If $T_i \ne \emptyset$ then goto 2.
7. End.
   The set of separating hyperplanes $\mathfrak{L} = \{\mathcal{L}_1, \dots, \mathcal{L}_i, \dots, \mathcal{L}_q\}$ is constructed.

The next step is to select the minimum number hyperplanes from the set $\mathfrak{L}$ which are sufficient for linear separation of the sets T_0 and T_1.

Denote $\beta_{pj} = \beta_{pj}(X) = 1$, if the hyperplane $\mathcal{L}_j$ separates the point $X_p \in T_1$ from all points of the set T_0, otherwise $\beta_{pj} = 0$. The condition of separation of the point X_p at least by one hyperplane takes the formal expression

$$\beta_{p1}\mathcal{L}_1 \vee \ldots \vee \ \beta_{pj}\mathcal{L}_j \vee \ldots \vee \ \beta_{pq}\mathcal{L}_q \,. \qquad (4)$$

Symbol $\mathcal{L}_j$ in the expression (4) should be understood as a formal variable that denotes the hyperplane $\mathcal{L}_j$. The hyperplane $\mathcal{L}_j$ is included in the set of hyperplanes separating the point X_p iff $\beta_{pj} = 0$.

Let $T_1 = \{X_1, \ldots, X_p, \ldots, X_\mu\}$. Consider the expression

$$\bigwedge_{p=\overline{1,\mu}} \left(\beta_{p1}\mathcal{L}_1 \vee \ldots \vee \ \beta_{pj}\mathcal{L}_j \vee \ldots \vee \ \beta_{pq}\mathcal{L}_q \right). \qquad (5)$$

By a logical multiplication and use of the absorption law to the expression (5) we can obtain all possible sets of separating hyperplanes and then choose the shortest of them and denote it $\widehat{\mathfrak{L}} = \{\widehat{\mathcal{L}_1}, \ldots, \widehat{\mathcal{L}_\mu}\}$. Below we'll explain why we should choose the shortest separating set.

Example 1. Let $n = 2$ and 19 point presented in Tables 1 and Table 2 are given.

Table 1

Points corresponding to the admissible solutions (T_0)

T_0	1	2	3	4	5	6	7	8	9	10	11	12
x_1	2	2	3	4	4	5	6	6	9	9	11	12
x_2	10	9	5	9	8	2	9	6	8	5	4	2

Table 2

Points which must be separated
from the admissible points (T_1)

T_1	13	14	15	16	17	18	19
x_1	1	3	5	8	12	12	17
x_2	12	14	11	14	10	6	5

The set of constructed hyperplanes $\mathfrak{L} = \{\mathcal{L}_1, \ldots, \mathcal{L}_7\}$ is presented in Table 3. This set is redundant. In the next step we'll find the irreducible set of separating hyperplanes.

Table 3

Set $\mathfrak{L}$ of hyperplanes built by using the RNLCP

Separated point index	*Hyperplane*	*All separated points of T_1*
13	$\mathcal{L}_1$: $15x_1 - 3x_2 + 2 = 0$	13
14	$\mathcal{L}_2$: $6x_1 - 3x_2 + 19 = 0$	13,14
15	$\mathcal{L}_3$: $-5x_2 + 52 = 0$	13,14,15,16
16	$\mathcal{L}_4$: $-8x_1 - 4x_2 + 115 = 0$	16,17,18,19
17	$\mathcal{L}_5$: $-19x_1 - 2x_2 + 244 = 0$	17,19
18	$\mathcal{L}_6$: $-36x_1 - 2x_2 + 440 = 0$	17,18,19
19	$\mathcal{L}_7$: $-18x_1 - 26x_2 + 170 = 0$	19

The Boolean expression that allows finding all possible irreducible collections of separating hyperplanes can be represented in the form (see Table 4)

$$(\mathcal{L}_1 \vee \mathcal{L}_2 \vee \mathcal{L}_3) \wedge (\mathcal{L}_2 \vee \mathcal{L}_3) \wedge \mathcal{L}_3 \wedge (\mathcal{L}_3 \vee \mathcal{L}_4) \wedge (\mathcal{L}_4 \vee \mathcal{L}_5 \vee \mathcal{L}_6) \wedge$$
$$\wedge (\mathcal{L}_4 \vee \mathcal{L}_6) \wedge (\mathcal{L}_4 \vee \mathcal{L}_5 \vee \mathcal{L}_6 \vee \mathcal{L}_7) = \mathcal{L}_3\mathcal{L}_4 \vee \mathcal{L}_3\mathcal{L}_6.$$

From the set of dead-end separators is more preferable $\mathcal{L}_3\mathcal{L}_4$ because hyperplane $\mathcal{L}_4$ separates four points, and the hyperplane $\mathcal{L}_6$ separates only three points. So, we chose the couple hyperplanes $\mathcal{L}_3\mathcal{L}_4$. Two separating hyperplanes can be made more precise. The hyperplane $\mathcal{L}_3$ separates the set $T_1^3 = \{X_{13}, X_{14}, X_{15}, X_{16}\}$ from the set T_0. By using the RNLCP we find more precise linear separator (Figure 3)

$$\mathcal{L}_3^* = 2\mathrm{x}_1 - 17\mathrm{x}_2 + 167 = 0.$$

Similarly we find

$$\mathcal{L}_4^* = -12x_1 - 6x_2 + 158 = 0.$$

Table 4

The logical table of linear separation of sets T_0 and T_1

Separated points	13	14	15	16	17	18	19
Hyperplanes							
$\mathcal{L}_1$	1						
$\mathcal{L}_2$	1	1					
$\mathcal{L}_3$	1	1	1	1			
$\mathcal{L}_4$				1	1	1	1
$\mathcal{L}_5$					1		1
$\mathcal{L}_6$					1	1	1
$\mathcal{L}_7$							1

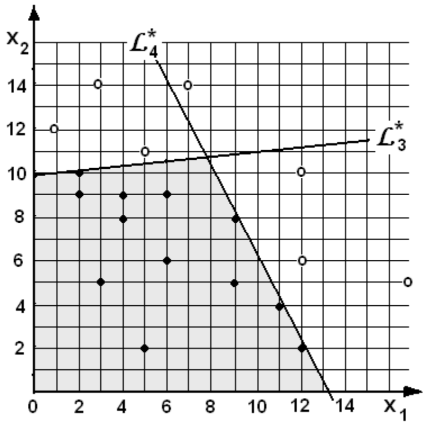


Figure 3. The equations of the hyperplanes $\mathcal{L}_3^*$ and $\mathcal{L}_4^*$ separating the sets T_0 (black points) and T_1 (white points)

As a result, the region of admissible solutions has the form

$$\begin{cases} 2\mathrm{x}_1 - 17\mathrm{x}_2 + 167 \le 0; \\ -12x_1 - 6x_2 + 158 \le 0; \\ x_1, x_2 \ge 0. \end{cases}$$

2.3 Building of the Linear Objective Function

Next we want to find approximation to the linear unknown objective function which the best fits to the given real data $\{X_k, y_k\}_{k=1}^{l}$, where $X_k = (x_1^k, \ldots, x_n^k) \in$

$\mathbb{R}^n$ and y_j is a value of unknown function $f\colon \mathbb{R}^n \to \mathbb{R}$. So, we assume the linear objective function can be presented in the form

$f(x_1, \dots, x_n) = w_1 x_1 + \dots + w_n x_n + w_0 = \langle W, X \rangle + w_0$, which need to be maximized.

For the finding W and w_0 the Method of Least Squares of can be used as well as the SVM Linear Regression which leads to the optimization problem

$$\begin{cases} \frac{1}{2}\langle W, W \rangle + C \sum_{k=1}^{l} (\xi_k^+ + \xi_k^-) \to \min_{W, w_0, \xi^+, \xi^-}; \\ y_k - \varepsilon - \xi_k^- \le \langle W, X_k \rangle + \mathrm{w}_0 \le y_k + \varepsilon + \xi_k^+; \\ \xi_k^+ \ge 0;\ \xi_k^- \ge 0;\quad k = 1, \dots, l, \end{cases}$$

where ξ_k^+ and ξ_k^- are additional slack variables;

$$\xi^+ = (\xi_1^+, \dots, \xi_l^+);\quad \xi^- = (\xi_1^-, \dots, \xi_l^-);$$

C is the regularization parameter; ε is the parameter set by expert.

The SVM method is more preferable since it aims at minimizing the norm $\|W\| = \sqrt{\langle W, W \rangle}$ of the vector W but it is more time consuming than the Method of Least Squares. Minimizing the norm $\|W\|$ leads to a reduction of the complexity of the synthesized model.

We propose another way of constructing linear objective function that allows checking the consistency of source data to assumed linear hypothesis.

We assume that in the sample there are no identical points and without loss of generality we assume that $y_1 > \dots > y_k > \dots > y_l$ (if not, one can renumber these data).

It is easy to prove that an unknown function built from the data T_l in reality can be linear if and only if for any $k = 2,3, \dots, l-1$ the set of points $\{X_1, \dots, X_k\}$ can be separated from the set of points $\{X_{k+1}, \dots, X_l\}$ by some hyperplane $\langle \widehat{W}, X \rangle + d_k = 0$.

Really, let an unknown function has the form $f(x_1, \dots, x_n) = \langle W, X \rangle + w_0$ and

$$y_1(X_1) > \dots > y_k(X_k) > \dots > y_l(X_l).$$

Then we have the inequalities

$$\langle W, X_1 \rangle > \dots > \langle W, X_k \rangle > \langle W, X_{k+1} \rangle > \dots > \dots > \langle W, X_l \rangle. \quad (6)$$

Putting $d_k = (y_{k+1} - \langle W, X_{k+1} \rangle - (y_k - \langle W, X_k \rangle)/2$ we obtain the required hyperplane $\langle W, X \rangle + d_k = 0$ for any $k = 2,3, \dots, l-1$.

Conversely, if $\langle W, X \rangle + d_k = 0$ for any $k = 2,3, \dots, l-1$ are the separating hyperplanes then the expression (6) is a compatible system of inequalities, so one can find its linear solution in the form $\langle \widehat{W}, \mathrm{X} \rangle + \widehat{w_0}$ by using the RNLCP.

The desired vector is obtained as a result of executing the following calculations:

$$W_0 = (0, \dots, 0);$$

$$W_t = \begin{cases} W_{t-1}, & if\ \langle W_{t-1}, X_j - X_q \rangle > 0; \\ W_{t-1} + (X_j - X_q), & if\ \langle W_{t-1}, X_j - X_q \rangle \le 0, \end{cases}$$

where $1 \le j < q \le l$ (recall that $y_j > y_q$); the computation steps t are repeated cyclically for all $T = l\,(l-1)/2$ pairs (j, q) until $W_t = W_{t-1}$ is executed T times. Denote the found vector $\widehat{W}$.

The value of w_0 which is not of special importance for the optimization model does can be chosen equal to

$$\widehat{w_0} = \tfrac{1}{l}\left(\sum_k y_k - \sum_k \langle \widehat{W}, X_k \rangle\right).$$

If the search procedure of the vector $\widehat{W}$ gets caught in an endless loop (or the number of steps exceeds the allowable value), the hypothesis of linearity of the objective function is not confirmed.

As the result of building of objective function and constraints we've obtained the model

$$\begin{cases} \langle \widehat{W}, \mathrm{X} \rangle + \widehat{w_0} \to \max; \\ \widehat{\mathcal{L}_1}(X) \le 0; \\ \dots\dots\dots\dots.. \\ \widehat{\mathcal{L}_\mu}(X) \le 0. \end{cases}$$

2.4 Justification of the Models Built from Data based on the Kolmogorov Complexity Theory

Heuristic approach to the building of shortest optimization models from data based on the principle of Occam's razor [3] can be justified on the base of the theory of Kolmogorov complexity [6]. This principle can be interpreted as a choice among competing hypotheses when the hypothesis with the fewest assumptions should be selected. In particular, Solomonoff's theory of inductive inference [9] is a mathematically formalised Occam's razor, namely: the shortest computable models have more weight compared to other. Under the shortest optimization models we understand such ones which have the shortest description in the form of specially-designed string of characters.

The model in the form

$$f(X) = w_1 x_1 + \dots + w_n x_n + w_0 = \langle W, X \rangle + w_0 \to max;$$

$$[a_{ji}]_{m \times n} \begin{bmatrix} x_1 \\ \dots \\ x_1 \\ \dots \\ x_n \end{bmatrix} \le \begin{bmatrix} b_1 \\ \dots \\ b_j \\ \dots \\ b_m \end{bmatrix};$$

with the parameters n, m, W, w_0, A, B is the individual representative of the a class $\mathbb{L}$ models of linear programming. The synthesized model as a rule does not coincide with the true unknown model, but approximates it.

The synthesized model is called a *consistent* (*consistent hypothesis*) if the substitution points from the training sample set $T_l = \{X_k, y_k, \alpha_k\}_{k=1}^{l}$ in the objective function and the linear inequality of this model satisfy the following conditions:

$$\langle \widehat{W}, X_k \rangle + \widehat{w_0} = y_k;$$

$$\widehat{\mathcal{L}_j}(X_k) \le 0\ if\ \alpha_k = 0;$$

$$\widehat{\mathcal{L}_j}(X_k) > 0\ if\ \alpha_k = 1;$$

$$k = 1, \dots, l.$$

Let's denote a consistent synthesized model $\mathcal{M}_T$.

It should be noted that in the wide family of models $\mathbb{L}$ there is a narrower subclass of consistent models $\mathbb{L}_\mathrm{T}$. A choice of the consistent model from the subclass $\mathbb{L}_\mathrm{T} \subset \mathbb{L}$ is complicated and incorrect by Hadamard task that requires serious justification.

Taking in account that all proposed in this paper methods and algorithms designed for computer realization, we need to narrow the considered family of the consistent models to finite models, restricting the representation of rational numbers by values belonging to the interval, which is defined with the bitness of the computer used.

We will call the Kolmogorov complexity of the consistent synthesized model $\mathcal{M}_{\mathrm{T}}$ for a given a Turing machine U

$$KC_U(\mathcal{M}_T) = min\{|p|: U(p) = s_{\widehat{W}} * s_{\widehat{w_0}} * s_{\mathfrak{L}}\},$$

where $|p|$ is the length of the binary string p such that a Turing machine U given an input string p will fully restore this synthesized model $\mathcal{M}_{\mathrm{T}}$ in the form of string description $\mathrm{s}_{\widehat{\mathrm{W}}} * \mathrm{s}_{\widehat{\mathrm{w}_0}} * \mathrm{s}_{\mathfrak{L}}$ (concatenation) of all components $\widehat{W}, \widehat{w_0}, \mathfrak{L}$ of the model $\mathcal{M}_{\mathrm{T}}$.

We will call a Turing machine U_s the generator of the string s if exists such binary string p that $U_s(p) = s$ and denote $\mathcal{U}_{\mathrm{s}}$ all possible generators for any string s.

Denote exact Kolmogorov complexity of the model $\mathcal{M}_T$

$$KC(\mathcal{M}_T) = \min_{U \in \mathcal{U}_s} KC_U(\mathcal{M}_T)\,.$$

Because of $KC(\mathcal{M}_T)$ is non-computable, we can only obtain some upper bound h such that $KC(\mathcal{M}_T) < h$.

Theorem 1 [5]. Let a drawing of any training sample from the set of all possible samples is equally probable and the consistent model $\mathcal{M}_T$ which is built from the training sample T_l has an estimation of exact Kolmogorov complexity $KC(\mathcal{M}_T) < h$. Then this model is non-random (regular) with probability not less $1-\varepsilon$ if

$$l_0 \geq (h + \log(1/\varepsilon))/log(\lceil\hat{\rho}/\delta\rceil - 2),$$

where $\hat{\rho} = \min_{X \in T_1} \rho(X, conv(T_0));$ $\rho(X, conv(T_0)$ is Euclidean distance from point X to the convex hull of the set T_0; $\varepsilon > 0$; δ is the module of the smallest rational number which can be presented in the format used by the computer; $l_0 = |T_0| = |\{X_k : \alpha_k = 0\}|$.

For example, $l_0(\varepsilon = 0.01, \hat{\rho} = 1, \delta = 10^{-4}, h = 3000) \approx 230$ and then $l = l_0 + l_1$ is approximately equal to 460.

The upper bound h of the exact Kolmogorov complexity can be found by summarizing the lengths of all model $\mathcal{M}_{\mathrm{T}}$ components which should be represented in the form of binary string. This implies the need for most compressed synthesized models and the usefulness of the SVM approach which provides minimization of norms of the vectors both functions, and constraints.

3. Building Optimization Model with Boolean Variables from Data

When the variables are Boolean, linear and nonlinear cases not need be necessarily divided if for the synthesis of the objective function and constraints binary trees used which have the property of functional completeness. Decision trees are widely used for a problem solving on the base of empirical generalization. This issue is devoted to the extensive scientific literature. [8].

If there is a priori assumption about the linearity of the objective function, in some cases it is useful to apply the pseudo-Boolean regression [2].

Now we consider the Boolean space $\mathbb{B}^n$ and the training sample set $T_l = \{X_k, y_k, \alpha_k\}_{k=1}^{l}$, where $X_k = (x_1^k, \dots, x_n^k) \in \mathbb{B}^n$; y_k is a value of unknown pseudo-Boolean function $f: \mathbb{B}^n \to \mathbb{R}$; $\alpha_k = 0$ if a vector X_k is admissible solution of the optimization problem which needs to be build, otherwise Boolean value $\alpha_k = 1$.

We'll use regression tree to build approximation $\hat{f}$ of the unknown objective function and classification tree to build the approximation of the admissible solution region $\widehat{\Omega}$. A consistent model $\mathcal{M}_{\mathrm{T}}$ built from data T_l must satisfy the following conditions: if $\alpha_k = 0$ then $X_k \in \widehat{\Omega}$; if $\alpha_k = 1$ then $X_k \notin \widehat{\Omega}$; $\hat{f}\,(X_k) = y_k$ for all $k = 1, \dots, l$.

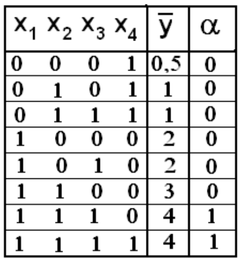
We omit detailed descriptions of well-known procedures for the synthesis of regression and classification trees [7]. Recall that binary tree defines specific set of orthogonal conjunctions such that any conjunction corresponds to the function value (for the regression tree) or the class number (for the classification tree). So, when the regression tree $\mathcal{T}_{\hat{f}}$ and the classification tree $\mathcal{T}_{\widehat{\Omega}}$ will be built, we'll have two sets of conjunction $\mathcal{K}_{\hat{f}}$ and $\mathcal{K}_{\widehat{\Omega}}$. Any conjunction $K \in \mathcal{K}_{\widehat{\Omega}}$ is drawn into 1 $(K(X) = 1)$ only if $X \in \widehat{\Omega}$ (when X is admissible solution), else if $K(X) = 0$ then $X \notin \widehat{\Omega}$.

There exists a method which allows combine both tree $\mathcal{T}_{\hat{f}}$ and $\mathcal{T}_{\widehat{\Omega}}$ to obtain pseudo-Boolean optimization model. Let's show this method with the following

Example 2. The training sample $(n = 4, l = 8)$ is presented in Table 5.

Table 5

The training sample $(n = 4)$



x_1	x_2	x_3	x_4	$\bar{y}$	α
0	0	0	1	0,5	0
0	1	0	1	1	0
0	1	1	1	1	0
1	0	0	0	2	0
1	0	1	0	2	0
1	1	0	0	3	0
1	1	1	0	4	1
1	1	1	1	4	1

At first we choose for example the conjunction $K_2 = x_1\overline{x_2}$ which allows separate points with a value $\bar{y} = 2$ of the object function from all other points contained in Table 5. Similarly, $K_{0,5} = \overline{x_1}\,\overline{x_2}$, $K_1 = \overline{x_1}\, x_2 x_4$, $K_3 = x_1 x_2 \overline{x_3}$, $K_\emptyset = x_1 x_2 x_3$, where conjunction $K_\emptyset$ corresponds the points for which $\alpha = 1$.

The trees presented in Figure 4 correspond the conjunctions K_2 (A), $K_{0,5}$ (B), K_1(C), K_3 (D), $K_\emptyset$(E). Regression decision tree obtained as a result of synthesis is shown in Figure 4 F. To achieve the maximum value of the objective function according to the synthesized decision tree (F) one should *assign the following values of control variables*: $x_1 = 1;\ x_2 = 1;\ x_3 = 0$.

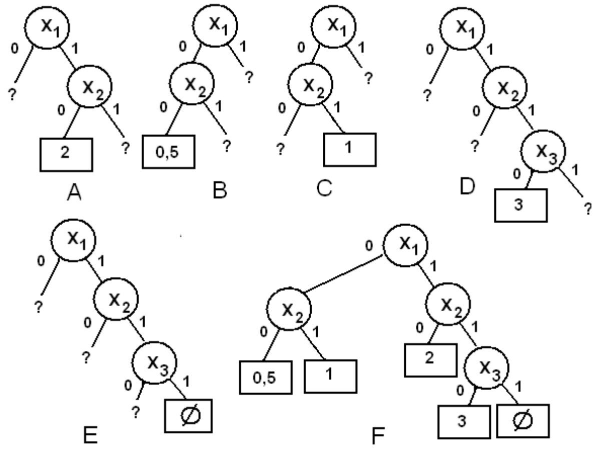


Figure 4. Synthesis of the combined regression tree

Let us mention one important result obtained by L. G. Babat [1] which allows in some cases estimate a number of elements of partitioning the interval of possible values of the linear pseudo-Boolean objective function on the classes of equivalent values.

Consider the Linear pseudo-Boolean function $f(X) = \sum_{i=1}^{n} c_i x_i$ with the coefficients $c_i > 0_\text{i}$. Let's say that the values $f(A)$ and $f(B)$ belong to the same ε-group ($\varepsilon > 0$) if

$$|f(A) - f(B)|/min\{f(A), f(B)\} \leq \ \varepsilon$$

or

$$max\{f(A), f(B)\}/min\{f(A), f(B)\} \leq \ 1 + \varepsilon.$$

Theorem 2 [1]. For any $\varepsilon > 0$ the function f values at the vertices of a unit cube $\mathbb{B}^n$ can be divided into ε-groups so that quantity of those groups will not exceed the number

$$q_n = n(1 + log_2(1 + \varepsilon)).$$

So, in some cases we can divide the interval $\left[\min_k y_k, \max_k y_k\right]$ on q_n pieces and assign to each piece the average of all values y_k from the training sample which hits in this piece. These averages $\overline{y_1}, \dots, \overline{y_{q_n}}$ can be used for approximation of sample values y_k and be considered as the set of labels for the leaves of the synthesized decision tree. By adding one more label $\emptyset$, corresponding to solutions that are not admissible, we can assign labels for all points from the training sample and then synthesize classification tree.

We named this method *the CBFA – Classification Based Function Approximation*.

There is another method that allows to directly synthesizing unknown linear pseudo-Boolean function $f(X) = w_1 x_1 + \dots + w_n x_n + w_0$ by use the training sample T_l. This method is based on the solution of the following problem:

$$w_0^2 + \sum_{i=1}^{n} w_i^2 + \gamma \sum_{k=1}^{l} \varepsilon_k^2 \to min;$$
$$\varepsilon_k + w_0 + \sum_{i=1}^{n} w_i x_i^{(k)} \ \geq y_k\,;$$
$$\varepsilon_k - w_0 - \sum_{i=1}^{n} w_i x_i^{(k)} \ \geq -y_k\,;$$
$$\varepsilon_k \geq 0;\ \ k = 1, \dots l,$$

where a constant γ is a parameter of the algorithm.

If the solution $\widehat{w}_0, \widehat{w}_1, \dots, \widehat{w}_i, \dots, \widehat{w}_n$ of this optimization problem is found then we have the synthesized pseudo-Boolean function

$$f(X) = \widehat{w}_0 + \textstyle\sum_{i=1}^{n} \widehat{w}_i x_i\ .$$

After then we can use the classification tree $\mathcal{T}_{\widehat{\Omega}}$ obtained by learning from the sample T_l. Let the tree $\mathcal{T}_{\widehat{\Omega}}$ corresponds the set of conjunction presented in the expression $\mathcal{K}_{\widehat{\Omega}} = K_1 \vee \dots \vee K_r \vee \dots \vee K_s$ and

$$K_r = x_{i_1^r}^{\sigma_{i_1^r}} \wedge \dots \wedge x_{i_{v_r}^r}^{\sigma_{i_{v_r}^r}}\ ,$$

where, as usually, $x^\delta = x$ if $\delta = 1$ and $x^\delta = \bar{x}$ if $\delta = 0$. Let's put in correspondence to the each conjunction K_r its weight $V_r = \widehat{w}_{i_1^r}\sigma_{i_1^r} + \dots + \widehat{w}_{i_{v_r}^r}\sigma_{i_{v_r}^r}$ and denote $V_\theta = max\{\ V_1, \dots,\ V_r, \dots,\ V_s\}$. Then the optimal values of control variables are $x_{i_1^\theta} = \sigma_{i_1^\theta}, \dots, x_{i_{v_\theta}^\theta} = \sigma_{i_{v_\theta}^\theta}$.

4. Observations on the Construction of Nonlinear Optimization Models from Data

The above Classification Based Function Approximation method (CBFA) in the case when variables are Boolean provides ability of building both linear and nonlinear models. But in many cases the building of nonlinear regression and nonlinear constraints requires additional information about the type of nonlinearity. If such additional information available, on its basis it is possible to reduce the synthesis of nonlinear model to linear synthesis.

For example, a multiplicative function of n real variables of the form

$$f(X) = \prod_{i=1}^{n} x_i^{a_i}$$

with unknown vector of coefficients $a_1, \dots, a_i, \dots, a_n$ by taking the logarithm is reduced to the linear function

$$a_1\, ln\, x_1 + \dots + a_i\, ln\, x_i + \dots + a_n\, ln\, x_n$$

with respect to the variables $y_i = ln\, x_i$.

In the paper [2] it is proposed an approach to construction of the nonlinear pseudo-Boolean regression based on data transformation. For this goal a map $\mathcal{F}: \{0,1\}^n \to \{0,1\}^m$ must be built, where initial space is $\{0,1\}^n = \mathbb{R}^n$ and resulting space is $\{0,1\}^m$. Components of this new space consist of monomials involving the components of $\{0,1\}^n$. For example, some monomial $x_{i_1}\bar{x}_{i_2}x_{i_3}$ may correspond to some variable y_j.

By this way, the nonlinear regression can be reduced to the synthesis of the linear regression

$$\hat{f}(Y) = \hat{v}_0 + \sum_{j=1}^{m} \hat{v}_j y_j \ .$$

5. Conclusion

In this paper we have shown, how can be synthesized models of the optimal control on the basis of samples or precedents. The proposed BOMD approach is based on empirical induction which is directed to obtaining regularities in the form of empirical optimization models which are synthesized in analytical form.

We follow the Kolmogorov idea about regularity as non-randomness. This allows us to estimate the probability of non-random model selection from the set of admissible models which are consistent to the sample or to given initial data.

As it is shown in the paper, the construction of optimization models of control from data can be implemented by different ways. The choice of method of model synthesis is determined by the accuracy requirements and resource constraints.

We do not claim to be exhaustive in the presentation of results in the field of research. In particular, beyond the scope of this article are left the questions of dynamic adaptation and reconfiguration of synthesized models, associated with the problem of "forgetting" of obsolete data. These issues are expected to devote our future research.

References

[1] L. G. Babat, Linear functions on *n*-dimensional unit cube, In: *Research on discrete optimization*: Moscow: Nauka, 1976, 156-169 (in Russian).

[2] Bonates, T., P. Hammer, *Pseudo-Boolean Regression* (RUTCOR Research Report, RRR 3-2007, Rutgers University: 2007).

[3] P. Domingos, The Role of Occam’s Razor in Knowledge Discovery, *Data Mining and Knowledge Discovery*, *3*(4), 1999, 409–425.

[4] V. I. Donskoy, Intelligent Control (A Survey), *Tavriceskij Vestnik Informatiki i Matematiki*, 2, 2014, 14-35.

[5] V. I. Donskoy, Synthesis of the Consistent Linear Optimization Models by Precedent Information: an Approach Based on Kolmogorov Complexity, *Tavriceskij Vestnik Informatiki i Matematiki*, 1, 2012, 13-23.

[6] M. Li, P. M. B. Vitanyi, Inductive Reasoning and Kolmogorov Complexity, *Journal of Computer and System Sciences*, *44*, 1992, 343-384.

[7] Y.-W. Loh, *Fifty Years of Classification and Regression Trees, International Statistical Review*, *82*(3), 2014, 329–348.

[8] S. Lomax, S. Vadera, A survey of cost-sensitive decision tree induction algorithms, *ACM Computing Surveys*, *45*(2), 2013, 16:1-16:35.

[9] R. J. Solomonoff, Complexity-based induction systems: comparisons and convergence theorems, *IEEE Trans. IT*, *24*, 1978, 422-432.

[10] S.G. Tzafestas, Overview of Intelligent Controls, In: *Methods and Applications of Intelligent Control*, Dordrecht/Boston: Kluwer, 1997, 3-23.

[11] W. Yu (ed.), *Recent Advances in Intelligent Control* (London: Springer, 2009).