

A SYNTHESIS OF PSEUDO-BOOLEAN EMPIRICAL MODELS
BY PRECEDENTIAL INFORMATION

V.I. Donskoy, Crimean Federal University, Simferopol, Russian Federation,
vidonskoy@mail.ru

The problem of decision-making based on partial, precedential information is the most important when the creation of artificial intelligence systems. According to the results of observations over the behaviour of external objects or systems it is necessary to synthesize or, more precisely, extract from the data a mathematical model of optimization of the object on the basis of accumulated empirical information in the form of a finite set of triples: "a state vector, the value of the quality of functioning of the object, a binary indicator of the admissibility of this state". The aim of the work is to create and substantiate mathematical methods and algorithms that allow to synthesize models of scalar pseudo-Boolean optimization with a constraint in the form of disjunctive normal form (DNF) using this precedential information. The peculiarity of pseudo-Boolean optimization models with separable objective functions and DNF constraint, which has a bounded constant length, is their polynomial solvability. However, the complexity of bringing the problem to the form with a DNF constraint in general case is exponential. When extracting the model from the data the DNF constraint is synthesized approximately but with polynomial complexity and the number of conjunctions in the extracted DNF does not exceed the number of examples in the initial precedential information. In the paper is shown how to use binary decision trees to construct a disjunctive constraint, proposed the methods to identify the properties of monotonicity and linearity of the partially defined objective functions, and developed algorithms for solving problems pseudo-Boolean scalar optimization in the presence of incomplete, precedential initial information. The scope of application of the obtained results includes intelligent control systems, intelligent agents. Although the control models derived from the data are approximate, their application can be more successful than the use of less realistic, inconsistent with the objects models which are chosen on the base of subjective considerations.

Keywords: pseudo-Boolean optimization; disjunctive constraint; machine learning; decision trees.

Introduction

Discrete pseudo-Boolean models of conditional scalar optimization generally have the form

$$\max(\min) f(\tilde{x}) \text{ under condition } \tilde{x} \in \Omega \subset B^n, \quad (1)$$

where $B^n = \{0, 1\}^n$ is the set of vertices of a unit n -dimensional cube, Ω is the set of admissible solutions defined by constraints of different types, $f : B^n \rightarrow \mathbb{R}$ is a pseudo-Boolean objective function, $\tilde{x} = (x_1, \dots, x_i, \dots, x_n) \in B^n$ is an arbitrary Boolean vector or *set of variables* that accept values 0 or 1 [1]. In the most common particular case, the model (1) contains restrictions in the form of linear inequalities and a linear objective

function, such as

$$\begin{cases} \max \sum_{i=1}^n c_i x_i; \\ a_{j1}x_1 + \dots + a_{ji}x_i + \dots + a_{jn}x_n \leq b_j; \quad j = \overline{1, m}; \\ x_i \in \{0, 1\}; \quad c_i, a_{ji} \in \mathbb{R}; \quad i = \overline{1, n}. \end{cases} \quad (2)$$

Model (2) is widely used to solve problems of production planning, equipment loading, investment, and in many other applications. When large dimension of the model (2) (when n and m are large) can arise the situation of *the complexity or even the impossibility of obtaining the full source numeric information in the form of objective function coefficients c_i , matrix constraints $[a_{ji}]_{m \times n}$, and values b_j* . Moreover, it may be even that there is no information on how the model constraints that define the set of admissible solutions Ω are "arranged". The statement of problems solved in this article can be explained using the concept of "black box". The system or controlled object under study is considered as a black box exposed to \tilde{x} and generating some responses in the form of numbers $f(\tilde{x})$ and possibly values of predicates $[\tilde{x} \in \Omega]$. It is required to construct as accurate as possible a model $\hat{\mathcal{M}} = \langle \hat{f}, \hat{\Omega} \rangle$ approximating its true but unknown model \mathcal{M} , knowingly belonging to the class (1) by the given set of impacts and responses. Then find the optimum point $\tilde{x}^* = \text{agmax}(\text{min}) \hat{f}(\tilde{x}) / \tilde{x} \in \hat{\Omega}$, and if some of the vector variables \tilde{x} are controllable, suggest an optimal control or planning for the model $\hat{\mathcal{M}}$.

In broad terms, problems considered in this paper are included in the scope of problems *acquisition of optimization models from data and making decisions by incomplete information*. Such problems were the first investigated by V.I. Mazurov [2].

Source partial information is represented by the training sample $\{(\tilde{x}_j, f(\tilde{x}_j), \gamma_j)\}_{j=1}^l$ which is a set of triples: points with known value of the function f and the predicate $\gamma_j = [\tilde{x} \in \Omega]$ in these points. This source information is believed to reflect the properties of a regular object or system and to be error-free (correct).

1. Basic Definitions and Statements Required

Definition 1. *The set of variables (the point) $\tilde{x} = (x_1, \dots, x_i, \dots, x_n)$ precedes the set $\tilde{z} = (z_1, \dots, z_i, \dots, z_n)$ (denoted as $\tilde{x} \preceq \tilde{z}$) if for all $i = \overline{1, n}$ the inequality $x_i \leq z_i$ is satisfied, and strictly precedes ($\tilde{x} \prec \tilde{z}$) if $\forall i (x_i \leq z_i)$ and at the same time $\exists i (x_i < z_i)$.*

Definition 2. *Pseudo-Boolean (Boolean) function f is called monotone if for any pair of points $\tilde{x}, \tilde{z} \in \text{Dom}(f)$ such that $\tilde{x} \preceq \tilde{z}$ the following inequality is satisfied $f(\tilde{x}) \leq f(\tilde{z})$.*

Definition 3. *A pseudo-Boolean function is called a linear one if it has the form $f(x_1, \dots, x_i, \dots, x_n) = c_0 + c_1x_1 + \dots + c_ix_i + \dots + c_nx_n$ where c_i are real numbers.*

A *literal* is an expression x^σ where x is Boolean variable, and σ is a Boolean constant such that $x^\sigma = x$ when $\sigma = 1$ and $x^\sigma = \bar{x}$ when $\sigma = 0$. To match the values of the Boolean variable x equivalent to its real values, formulas are introduced $\bar{x} = 1 - x$ и $x = \bar{\bar{x}} = 1 - (1 - x)$. For variables that accept values only 0 or 1 the identities $x_i \cdot x_i \cdot \dots \cdot x_i \equiv x_i$ and $x_i \cdot \bar{x}_i \equiv 0$ are performed. Therefore pseudo-Boolean polynomials may not include members that contain a degree variable, and may include only the products of different variables without inversions.

Theorem 1. *Any pseudo-Boolean function can be represented as a polynomial.*

Proof. Let arbitrary pseudo-Boolean function of n variables $f : B^n \rightarrow \mathbb{R}$ takes at points $\tilde{\sigma}_0, \dots, \tilde{\sigma}_j, \dots, \tilde{\sigma}_{2^n-1}$ values $y_0, y_1, \dots, y_{2^n-1}$, $\tilde{\sigma}_j = (\sigma_j^1, \dots, \sigma_j^n)$. If one multiply the characteristic function of each of these points by the corresponding value of the function f in them and then add the resulting members, then an expression $f(x_1, x_2, \dots, x_n) = \sum_{j=0}^{2^n-1} y_j \cdot x_1^{\sigma_j^1} \cdot x_2^{\sigma_j^2} \cdot \dots \cdot x_n^{\sigma_j^n}$ where $x_i^{\sigma_j^i} = 1 \Leftrightarrow x_i = \sigma_j^i$ will be obtained. Replacing $x_i^0 = 1 - x_i$, multiplication and bringing like terms gives the polynomial. \square

In the optimization model (1) constraints, generally speaking, can be presented in different forms – algebraic, graph, logical, even in the form of a specific algorithm. The objective function as well can be represented by various ways. The variety of possible representations of the model (1) makes it reasonable to introduce the concept of *the form of representation*. For example (2) is a model with constraints in the form of non-strict inequalities and with a linear objective function.

Definition 4. *The model of scalar pseudo-Boolean optimization in the form*

$$\max (\min) f(x_1, \dots, x_n) \text{ provided } \bigvee_{j=1}^m x_{j1}^{\sigma_{j1}} \wedge \dots \wedge x_{jk_j}^{\sigma_{jk_j}} = 1 \quad (3)$$

is called the first form with disjunctive (DNF) constraint and the model

$$\max (\min) f(x_1, \dots, x_n) \text{ provided } \bigvee_{q=1}^{\mu} x_{q1}^{\sigma_{q1}} \wedge \dots \wedge x_{qk_q}^{\sigma_{qk_q}} = 0 \quad (4)$$

is the second form with DNF constraint.

Theorem 2. *Any model of pseudo-Boolean conditional optimization can be represented both in the first and in the second forms with DNF constraint.*

Proof. The constraint $\tilde{x} \in \Omega$ in general form contained in the model (1) can be represented in the following equivalent form. We introduce Boolean function $\varphi_{\Omega}(\tilde{x}) = 1 \Leftrightarrow \tilde{x} \in \Omega$. Then the constraint takes the equivalent form $\varphi_{\Omega}(\tilde{x}) = 1$. Any Boolean function can be represented in a disjunctive normal form, so when representing the function φ_{Ω} in the DNF the first form can be obtained. Defining otherwise Boolean function: $\Psi_{\Omega}(\tilde{x}) = 0 \Leftrightarrow \tilde{x} \in \Omega$ one can get the second form. \square

Lemma 1. *If in the linear pseudo-Boolean optimization model (1) all coefficients of the constraints are positive and*

$$a_{j1}x_1 + \dots + a_{ji}x_i + \dots + a_{jn}x_n \leq b_j \Leftrightarrow \psi_j(\tilde{x}) = 0$$

then ψ_j is monotone Boolean function and $\Psi_{\Omega} = \bigvee_{j=1}^m \psi_j$ is monotone Boolean function.

Proof. Let $\tilde{\alpha} \leq \tilde{\beta}$, i.e. $\forall i (\alpha_i \leq \beta_i)$. Then

$$S_{\tilde{\alpha}} = \sum_{i=1}^n a_{ji}\alpha_i - b_j \leq \sum_{i=1}^n a_{ji}\beta_i - b_j = S_{\tilde{\beta}}.$$

By the condition of the Lemma $S_{\tilde{\alpha}} \leq 0 \Leftrightarrow \psi_j(\tilde{\alpha}) = 0$ и $S_{\tilde{\beta}} \leq 0 \Leftrightarrow \psi_j(\tilde{\beta}) = 0$. Because of the inequality $S_{\tilde{\alpha}} \leq S_{\tilde{\beta}}$ the following mathematical expressions are valid: $\psi_j(\tilde{\beta}) = 0 \Rightarrow \psi_j(\tilde{\alpha}) = 0$ и $\psi_j(\tilde{\alpha}) = 1 \Rightarrow \psi_j(\tilde{\beta}) = 1$ whence it follows the inequality $\psi_j(\tilde{\alpha}) \leq \psi_j(\tilde{\beta})$, i.e. ψ_j is monotone Boolean function. The class of monotone Boolean functions is closed and contains disjunction, so the function $\Psi_{\Omega} = \bigvee_{j=1}^m \psi_j$ is also monotone. \square

Theorem 3. *If the problem of conditional linear pseudo-Boolean optimization (2) with positive coefficients in constraints-inequalities is presented in the second form with DNF constraint*

$$\Psi_{\Omega}(\tilde{x}) = \bigvee_{q=1}^{\mu} x_{q1}^{\sigma_{q1}} \wedge \dots \wedge x_{qk_q}^{\sigma_{qk_q}} = 0,$$

then Ψ_{Ω} is monotone function.

Proof. Using Theorem 2 and Lemma 1 it is easy to get a proof of the theorem. \square

Theorem 4. *The problem of scalar pseudo-Boolean conditional optimization of a linear function with n variables and DNF constraint*

$$\max (\min) \sum_{i=1}^n c_i x_i / \bigvee_{j=1}^m x_{j1}^{\sigma_{j1}} \wedge \dots \wedge x_{jk_j}^{\sigma_{jk_j}} = 1 \quad (5)$$

which contains m conjunctions is solvable with a time complexity $O(mn)$.

Proof. DNF-constraint in (5) contains m conjunctions $x_{j1}^{\sigma_{j1}} \wedge \dots \wedge x_{jk_j}^{\sigma_{jk_j}}$ each of which is drawn per unit at its corresponding interval of rank k_j . Boolean variables with numbers $j1, \dots, jk_j$ are fixed for this interval by values $\sigma_{j1}, \dots, \sigma_{jk_j}$, and the other variables are free. If the linear function is maximized any free variable x_i should be taken equal to one if $c_i > 0$, equal to zero, if $c_i < 0$, and arbitrary when $c_i = 0$. Therefore, the linear pass at all m conjunctions with simultaneous search of the maximum of the linear function at all intervals will require $O(mn)$ of calculation steps. \square

Corollary 1. *If the problem of conditional optimization of a linear pseudo-Boolean function is reduced to the form with DNF constraint for polynomial number of steps, then it is polynomial solvable.*

Most often, the problems of conditional optimization of pseudo-Boolean functions including those presented by the model (2) are NP-hard. The exceptions are, in particular, models with separable objective function the set of constraints of which corresponds to the structure of the matroid. Therefore, taking into account corollary 1, the construction of the constraint DNF is in itself a complex problem. However, *within the framework of the problems considered in this article, the approximation of the domain of admissible solutions by logical machine learning algorithms leads to the construction of DNFs having a length not exceeding the number of training sample examples.*

2. DNF Constraint Synthesis as a Problem of Machine Learning: Approach Based on the Use of Decision Trees

If the initial partial information contains the predicate values $\gamma_j = [\tilde{x}_j \in \Omega]$ at the training sample points then mathematical precedent-based machine learning methods

can be used to construct an approximation $\hat{\varphi}_\Omega(\tilde{x})$ or $\hat{\Psi}_\Omega(\tilde{x})$ of $\hat{\Omega}$ in the form of DNF. The most suitable for this purpose are learning algorithms based on the construction of binary decision trees (BDT) which are algorithmic operators that display training sequences of precedents in a family of Boolean functions having a tree-like structural representation [3]. It is necessary for such algorithmic operators to possess the ability for empirical generalization and guarantee *learnability* in the sense of arbitrarily accurate approximation of the constructed empirical DNF $\hat{\mathcal{D}}$ to the true but unknown DNF \mathcal{D} as the number of precedents in the training sample grows.

When one builds BDT classifying the points $\tilde{x} \in B^n$ (considered in this article the case – onto two classes: satisfying the constraints and do not satisfying them), the initial information is a training sample $\{(\tilde{x}_j, \gamma_j)\}_{j=1}^l$, where $\gamma_j = 1 \Leftrightarrow \tilde{x}_j \in \Omega$ and $\gamma_j = 0 \Leftrightarrow \tilde{x}_j \notin \Omega$. The procedure of BDT synthesis consists in sequential execution of the *same-type branching steps* or, in a set-theoretic interpretation, splitting intervals of B^n , starting from the set of B^n as n -dimensional interval, when the root vertex of the tree is built, containing one of the variables and two outgoing edges that correspond to the unit and zero values of the selected variable. Such a partition is called *splitting or branching*. Branching continues until the condition of stopping is met which is the presence in each interval of the resulting partition (or, let's say, in each leaf of the tree) of points of only one and the same class [4].

The following simple example is intended to illustrate the proposed approach to the construction of a constraint in the form of DNF by decision tree machine learning.

Example 1. Assume that the learning information was generated by the following \mathcal{M} model with a single constraint in the form of an inequality:

$$\max f(x_1, x_2, x_3, x_4) = 2x_1 + x_2 + 4x_3 + 2x_4; \quad 3x_1 + x_2 + 2x_3 + x_4 \leq 3. \quad (6)$$

Let the target function in (6) be known exactly, and the area of admissible solutions actually defined by inequality (6) is unknown, but is partially represented by eight precedent points (Table 1).

Table 1

Precedents for learning

x_1	x_2	x_3	x_4	γ
0	0	0	1	1
0	0	1	1	1
0	1	0	1	1
0	1	1	1	0
1	0	0	0	1
1	0	1	0	0
1	0	1	1	0
1	1	1	0	0

Each point has a value of γ equal to one if the point satisfies the inequality (6) and zero – otherwise. One of the possible BDT built on the learning sample contained in the Table 1 is presented on Figure. This BDT is equivalent to the following DNF:

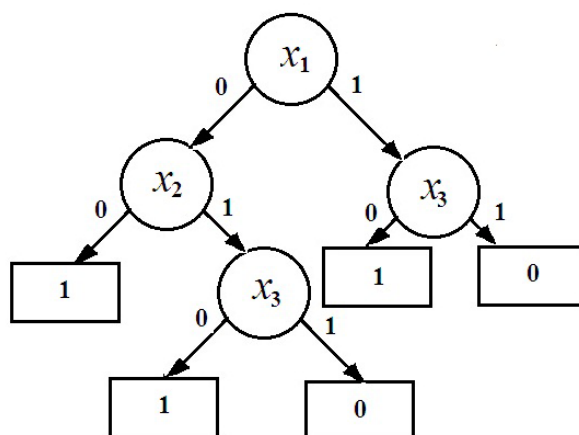
$$\bar{x}_1\bar{x}_2 \vee \bar{x}_1x_2\bar{x}_3 \vee x_1\bar{x}_3 \equiv \bar{x}_1\bar{x}_2 \vee \bar{x}_3. \quad (7)$$

We obtain approximation $\hat{\varphi}(\tilde{x}) = \bar{x}_1\bar{x}_2 \vee \bar{x}_3$ and, accordingly, constraint $\hat{\varphi}_\Omega(\tilde{x}) = 1$ in the first form. Since $\hat{\Psi}_\Omega(\bar{x}) = 0 \Leftrightarrow \hat{\varphi}(\tilde{x}) = 1$, we obtain the approximating function $\hat{\Psi}_\Omega(\bar{x}) = \bar{x}_1\bar{x}_2 \vee \bar{x}_3 = x_1x_3 \vee x_2x_3$. DNF (7) defines two possibilities for choosing a solution: either $x_1 = x_2 = 0$, and the remaining variables can be assigned by any (since the coefficients of the target function are positive, then by units); or $x_3 = 0$, and the remaining variables can be assigned by units. As a result, $f(0, 0, 1, 1) = 6$, $f(1, 1, 0, 1) = 5$, and the decision $\max f(\tilde{x}) = 6$ at point $\tilde{x} = (0, 0, 1, 1)$. Approximations $\hat{\varphi}_\Omega(\tilde{x})$ and $\hat{\Psi}_\Omega(\tilde{x})$ are different from the true functions

$$\varphi_\Omega(\hat{x}) = \underbrace{\bar{x}_1\bar{x}_2}_6 \vee \underbrace{\bar{x}_1\bar{x}_3}_3 \vee \underbrace{\bar{x}_1\bar{x}_4}_5 \vee \underbrace{\bar{x}_2\bar{x}_3\bar{x}_4}_2 = 1 \Leftrightarrow 3x_1 + x_2 + 2x_3 + x_4 \leq 3 \text{ and}$$

$$\Psi_\Omega(\tilde{x}) = x_1x_2 \vee x_1x_3 \vee x_1x_4 \vee x_2x_3x_4 = 0 \Leftrightarrow 3x_1 + x_2 + 2x_3 + x_4 \leq 3$$

(under conjunctions of DNF of function $\varphi_\Omega(\hat{x})$ the maximal values of the object function $2x_1 + x_2 + 4x_3 + 2x_4$ in the intervals corresponding to these conjunctions are signed).



Binary Decision Tree

In a particular case considered in the example 1 the constraint $\hat{\varphi}_\Omega(\hat{x}) = 1$ built on the learning sample information really allowed to find the exact solution of the problem (6). But in the general case, it is no guarantee of obtaining an exact solution of this problem according to partial learning information. \square

Denote $\hat{\Omega} = \{\tilde{x} : \hat{\varphi}_\Omega(\hat{x}) = 1\}$. When constructing a BDT to obtain an approximation of the constraints region $\hat{\Omega}$ at each branching step t the set B^n is split into shredded intervals $N_1, \dots, N_s, \dots, N_t$.

Definition 5. Interval $N_s \subset B^n$ is called correct if $N_s \subset \Omega$ or $N_s \subset B^n \setminus \Omega$. The union of all correct intervals obtained after execution the step t of BDT constructing is called the correct approximation region Ω_t^{corr} , and the set $B^n \setminus \Omega_t^{corr}$ is called incorrect region.

Theorem 5.

1^o $\Omega_1^{corr} \subseteq \dots \subseteq \Omega_s^{corr} \subseteq \dots \subseteq \Omega_t^{corr}$ for all t satisfying the double inequality $2 \leq t \leq T$ where T is the final step of the BDT synthesis;

2^o In some step $t^* \leq (2^n - 1)$ the equality $\Omega_{t^*}^{corr} = B^n$ can be performed provided that the length of the training sample $l \geq t^* + 1$.

Proof. 1) Let's suppose that after the branching step $t-1$, $t \geq 2$, the correct approximation region Ω_{t-1}^{corr} is formed (this region may be empty at the first branching steps). By definition, it consists only of correct intervals. In step t the correct intervals are not partitioned, so a certain interval \mathfrak{N} from the region $B^n \setminus \Omega_{t-1}^{corr}$ will be selected for partitioning. This interval will be split into two intervals \mathfrak{N}_0 и \mathfrak{N}_1 : $\mathfrak{N}_0 \cup \mathfrak{N}_1 = \mathfrak{N}$. If at least one of the intervals $\mathfrak{N}_0, \mathfrak{N}_1$ will be correct the area of correctness will expand otherwise it will remain unchanged, so $\Omega_{t-1}^{corr} \subseteq \Omega_t^{corr}$.

2) In the first step the set B^n is split into two intervals. Then, at each subsequent step, some selected interval of already obtained partition is split into two new intervals and sooner or later the equality $\Omega_t^{corr} = B^n$ can be fulfilled. Indeed, in the step t of the partitioning $t+1$ intervals will be obtained, and in the worst-case such process will continue until a step $t = 2^n - 1$ when the set B^n will be split onto 2^n intervals of rank n so that each of them will contain exactly one point from the correct learning sample. Therefore, the equality $\Omega_{2^n-1}^{corr} = B^n$ will certainly be achieved. □

Theorem 5 states the monotonous refinement of the region of correct approximation of the partially given constraint with the growth of the length of the training sample right down until an accurate result. But herewith the correctness of all examples in the training sample is obligatory condition. The latter condition is natural in the approach to extracting optimization models from the data when regular processes and systems are investigated. In this case, we are not talking about any probabilistic distributions, but a possibility exists *to assess the acceptability of the synthesized DNF on the base of Kolmogorov approach to the evaluation of the regularity as non-randomness.*

A.N. Kolmogorov emphasized the need to distinguish between actual *randomness as the absence of regularity* and a stochastic randomness as the subject of the theory of probability [5, p. 42]. When empirical extraction of regularity based on the Kolmogorov approach, it is possible to estimate the non-randomness of the found regularity, in the considered case – of empirical DNF approximating the constraints of the optimization problem.

Definition 6. *A result of machine learning is called an exact tuning on a training sample when the obtained empirical decision rule accurately calculates the approximable value at each sampling point (for each precedent).*

Theorem 6. [6] *Let empirical regularity is extracted from the family \mathcal{F} and an appearance in the sample $\{(\tilde{x}_j, \gamma_j)\}_{j=1}^l$ of any example from the general population $\underbrace{B^{n+1} \times \dots \times B^{n+1}}_l$ is equiprobably. Then the probability $P(\mathcal{F}, l)$ of a random exact tuning on a training sample of length l satisfies the inequality $P(\mathcal{F}, l) < 2^{-l+\mathcal{K}(\mathcal{F})}$ where $\mathcal{K}(\mathcal{F})$ is Kolmogorov algorithmic complexity of the family \mathcal{F} .*

Corollary 2. *When the conditions of the theorem 6 are met there is the estimation $P(\mathcal{F}, l) < 2^{-l+pVCD(\mathcal{F})}$ where $pVCD(\mathcal{F})$ is upper bound on the Kolmogorov complexity of the family \mathcal{F} which is obtained by pVCD method [6].*

If binary decision trees are used to extract such regularity as DNF constraint then the family \mathcal{F} is a class $BDT_{n,\mu}$ of trees with at most μ leaves and from n Boolean variables. It is known the estimation [7]

$$pVCD(BDT_{n,\mu}) = (\mu - 1)(\lceil \log_2 n \rceil + \lceil \log_2(\mu + 3) \rceil). \quad (8)$$

We require the condition $P(\mathcal{F}, l) \leq \varepsilon$ which will lead to an equation to determine the required length of the training sample to ensure that the probability of random extraction of the DNF constraint will not exceed ε . From the equation $\varepsilon = 2^{-l+(\mu-1)(\lceil \log_2 n \rceil + \lceil \log_2(\mu+3) \rceil)}$ we get $l = \log_2 \frac{1}{\varepsilon} + (\mu - 1)(\lceil \log_2 n \rceil + \lceil \log_2(\mu + 3) \rceil)$. The results of calculating the required sample length $l = l(n, \mu, \varepsilon)$ are shown in the Table 2.

Table 2

The required length of the training sample when $\frac{1}{\varepsilon} = 128, \varepsilon \approx 0,0078$

n	20	20	20	50	50	100
μ	20	50	100	30	100	100
$l(n, \mu, \varepsilon)$	197	546	1195	355	1294	1393

3. Approximation of Partially Defined Pseudo-Boolean Functions

Assume that objective function f is monotone, we need to find its maximum, and monotone approximation \hat{f} is obtained. Let the empirical DNF constraint $\bigvee_{j=1}^m K_j = 1$ is constructed. Then in any interval of the set B^n which corresponds to the conjunction $K_j = x_{j_1}^{\sigma_{j_1}} \wedge \dots \wedge x_{j_i}^{\sigma_{j_i}} \wedge \dots \wedge x_{j_{k_j}}^{\sigma_{j_{k_j}}}$ the extreme value $\hat{f}(\tilde{x}_{K_j}^*)$ is immediately found in the point $\tilde{x}_{K_j}^*$ which is defined as follows. Variables with numbers j_i such that $\sigma_{j_i} = 1$ are getting unit values, $\sigma_{j_i} = 0$ – zero values, and the rest variables, free from entering the conjunction K_j , according to the monotonicity of the objective function are determined by units. Next is $\hat{x}^* = \operatorname{argmax}_j \hat{f}(\tilde{x}_{K_j}^*) = \operatorname{argmax}_j \hat{f}(\tilde{x}) / x_{j_1}^{\sigma_{j_1}} \wedge \dots \wedge x_{j_i}^{\sigma_{j_i}} \wedge \dots \wedge x_{j_{k_j}}^{\sigma_{j_{k_j}}} = 1$. It is easy to make sure that the time complexity of the described algorithm for finding the extremum of a monotone pseudo-Boolean function with DNF constraint is estimated as $O(mn)$.

As shown above, information on the monotony of the function partially defined by sample greatly simplifies the solution of the problem. To check the monotony condition on a given training sample it is quite simply by the following Algorithm 1.

Algorithm 1. Checking the consistency property with monotonicity of a function on a training sample.

Input: the correct sample $\{(\tilde{x}_j, f(\tilde{x}_j))\}_{j=1}^l$.

Output: $M = 1$ if the sample does not contradict the monotony condition otherwise $M = 0$.

```

1:  $M := 1$ ;
2: for  $j := 1$  to  $l - 1$  do
3:   for  $s := j + 1$  to  $l$  do
4:     if  $(\tilde{x}_j < \tilde{x}_s) \wedge (f(\tilde{x}_j) > f(\tilde{x}_s)) \vee (\tilde{x}_s < \tilde{x}_j) \wedge (f(\tilde{x}_s) > f(\tilde{x}_j))$ 
       then  $M := 0$ ; stop end then.

```


Let us denote further $(C, \tilde{x}) = \sum_{i=1}^n c_i x_i$.

Definition 7. The function f which is partially defined by the training sample $\{(\tilde{x}_j, f(\tilde{x}_j))\}_{j=1}^l$ allows linear approximation if there is such a vector $C \in \mathbb{R}^n$ that for any pair of points $(\tilde{x}_p, \tilde{x}_q)$ from this sample such that $f(\tilde{x}_p) < f(\tilde{x}_q)$ the inequality $(C, \tilde{x}_p) < (C, \tilde{x}_q)$ is satisfied.

Theorem 7. The function f which is partially defined by the training sample $\{(\tilde{x}_j, f(\tilde{x}_j))\}_{j=1}^l$ where $f(\tilde{x}_j) \neq f(\tilde{x}_m)$, $1 \leq j < m \leq l$, admits a linear approximation if and only if in sorted by values of objective function $f(\tilde{x}_j)$ sequence of sample points $\tilde{x}_{j_1}, \dots, \tilde{x}_{j_p}, \dots, \tilde{x}_{j_l}$ such that $f(\tilde{x}_{j_1}) < \dots < f(\tilde{x}_{j_p}) < \dots < f(\tilde{x}_{j_l})$ for all $p = 1, 2, \dots, (l - 1)$ the points $\tilde{x}_{j_1}, \dots, \tilde{x}_{j_p}$ can be separated from the points $\tilde{x}_{j_{p+1}}, \dots, \tilde{x}_{j_l}$ by the hyperplane $(C^*, \tilde{x}) = \lambda_p$, at that $(C^*, \tilde{x}_{j_p}) < (C^*, \tilde{x}_{j_{p+1}})$.

Proof. The necessity. Let the sample allows a linear approximation. Then there exists a vector $C^* \in \mathbb{R}^n$ such that $(C^*, \tilde{x}_{j_1}) < \dots < (C^*, \tilde{x}_{j_p}) < (C^*, \tilde{x}_{j_{p+1}}) < \dots < (C^*, \tilde{x}_{j_l})$. It is obvious that the hyperplanes $(C^*, \tilde{x}) = \lambda_p$, $p = \overline{1, (l - 1)}$, where $\lambda_p = \frac{1}{2}((C^*, \tilde{x}_{j_p}) + (C^*, \tilde{x}_{j_{p+1}}))$ satisfy the condition of the theorem.

Sufficiency. Let the separating hyperplanes $(C^*, \tilde{x}) = \lambda_p$ exist. Then the vector C^* satisfies the admissibility requirement of linear approximation by the transitivity of the relation " $<$ ". □

Theorem 8. If for the sorted by values of objective function $f(\tilde{x}_j)$ sequence of the points $\tilde{x}_{j_1}, \dots, \tilde{x}_{j_p}, \dots, \tilde{x}_{j_l}$ such that $f(\tilde{x}_{j_1}) < \dots < f(\tilde{x}_{j_p}) < \dots < f(\tilde{x}_{j_l})$ for all $p = 1, 2, \dots, (l - 1)$ exists a hyperplane $(C_p, \tilde{x}) = \lambda_p$ separating the points $\tilde{x}_{j_1}, \dots, \tilde{x}_{j_p}$ from the points $\tilde{x}_{j_{p+1}}, \dots, \tilde{x}_{j_l}$ and $(C_p, \tilde{x}_{j_p}) < (C_p, \tilde{x}_{j_{p+1}})$ then a vector $C^* \in \mathbb{R}^n$ exists which defines the hyperplanes $(C^*, \tilde{x}) = \beta_p$ separating the points $\tilde{x}_{j_1}, \dots, \tilde{x}_{j_p}$ from the points $\tilde{x}_{j_{p+1}}, \dots, \tilde{x}_{j_l}$, herewith $(C^*, \tilde{x}_{j_p}) < (C^*, \tilde{x}_{j_{p+1}})$.

Proof. The opposite assumption

$$\overline{\exists C^* \in \mathbb{R}^n \quad \forall p (C^*, \tilde{x}_{j_p}) < (C^*, \tilde{x}_{j_{p+1}})} \Leftrightarrow \forall C^* \in \mathbb{R}^n \exists p : (C^*, \tilde{x}_{j_p}) \geq (C^*, \tilde{x}_{j_{p+1}})$$

entails the denial of the condition $\exists C_p \in \mathbb{R}^n (C_p, \tilde{x}_{j_p}) < (C_p, \tilde{x}_{j_{p+1}})$. □

To verify the possibility of linear approximation of the objective function according to the theorems 7 and 8 we need to check the linear separability of points $\tilde{x}_{j_1}, \dots, \tilde{x}_{j_p}, \dots, \tilde{x}_{j_l}$ of the training sample which are ordered by increasing values of the objective function. Namely: j_1 point from all the others, j_1 point and j_2 points from all the others and forth until separability of all points with numbers $j_1, \dots, j_{(l-1)}$ from the point j_l . Overall we need to perform such check $l - 1$ times.

Remark 1. If the training sample $\{(\tilde{x}_j, f(\tilde{x}_j))\}_{j=1}^l$ contains a subset of the points with the same value of the objective function f then such a subset must always be included in one of the groups to be separated. It is easy to verify that with this addition the theorem 7 remains true, and the number of separability checks required will be one less than the number of subsets of training sample points with the same values of the objective function.

To check sequentially the separability of two groups of points with simultaneous construction of separating hyperplanes (in the direction of increasing the objective function

as in 8) it is advisable to use the iterative procedure of Rosenblatt-Novikov linear correction [8]. If two finite sets of points $G1$ and $G2$ are linearly separable then this procedure provides constructing a separating hyperplane in a finite number of correction steps $k \leq \lceil D^2/\rho^2 \rceil$ where $D = \sup_{\tilde{x} \in G1 \cup G2} \|\tilde{x}\|$, and ρ is half the distance between convex shells of the sets $G1$ and $G2$ [8].

Let us denote $X_j = (x_1^j, \dots, x_n^j, 1) \in \mathbb{B}^{n+1}$ – extended vector representing the description of the point \tilde{x}_j with adding the $(n + 1)$ -th coordinate to which assigned a value 1; $C = (c_1, \dots, c_n, c_{n+1}) \in \mathbb{R}^{n+1}$ is an extended vector which defines the separating hyperplane $c_1x_1 + \dots + c_nx_n + c_0 = 0$. The linear correction procedure starting with an arbitrarily given initial vector C "adjusts" its coefficients by formulas implemented in lines 13 and 14 of the Algorithm 2 below. The parameter of this algorithm is the maximum number of exceeding of cyclic views of the sample when learning what means that the algorithm failed to establish the separability of sets $G1$ and $G2$ formed in accordance with the theorem 8.

Algorithm 2. Test of admissibility of linear approximation of the function f .

Input: A sample $\{(\tilde{x}_j, f(\tilde{x}_j))\}_{j=1}^l$ sorted by non-decreasing values $f(\tilde{x}_j)$;

Max – the maximum number of cyclic views of the sample when learning.

Output: $L = 1$ if the sample does not contradict the monotony condition otherwise $L = 0$.

```

1:  $L := 1$ ;                                \ \ A flag of result.
2:  $C = (0, \dots, 0, 0)$ ;  $\alpha := 1$ ;      \ \ Initialization.
3:  $q := 1$ ;                                \ \ The position of the split of the sample.
4: while  $f(\tilde{x}_q) = f(\tilde{x}_{q+1})$  do  $q := q + 1$ ; \ \ If the function values are equal.
5: if  $q = l$  then stop;                  \ \ If all groups are separated.
6:  $G1 := \{X_1, \dots, X_p, \dots, X_q\}$ ;   \ \ The first group to be separated.
7:  $G2 := \{X_{q+1}, \dots, X_p, \dots, X_l\}$ ; \ \ The second group to be separated.
8:  $Count := 0$ ; \ \ A count of number of scan cycles of training sample when learning.
9:  $t := 1$ ; \ \ A count of number of corrections.
10: if  $Count > Max$  then  $L := 0$ ; stop end then \ \ Corrections number  $> Max$ .
11:  $LS := 1$ ; \ \ Assume that corrections was not, and next separator is constructed.
12: for  $p := 1$  to  $l$  do \ \ Cycle by all points of the sample.
13:   if  $(C, X_p) \geq 0 \wedge X_p \in G1$  then  $C := C - \frac{\alpha}{t} \cdot X_p$ ;  $t := t + 1$ ;  $LS := 0$  end then;
14:   if  $(C, X_p) \leq 0 \wedge X_p \in G2$  then  $C := C + \frac{\alpha}{t} \cdot X_p$ ;  $t := t + 1$ ;  $LS := 0$  end then;
       end for  $p$ ;
15: if  $LS = 0$  then  $Count := Count + 1$ ; goto 10 end then; \ \ Increase cycles number.
16: if  $q < (l - 1)$  then  $q := q + 1$ ; goto 4 end then. \ \ To the following separation.

```

If it turns out that the data does not allow linear approximation, it is possible to check the possibility of quadratic approximation. Assuming that $f(x) = \sum_{i=1}^{n-1} \sum_{j=i+1}^n c_{ij}x_i x_j + \sum_{j=1}^n c_{0j}x_j + c_{00}$ and by change of variables $x_1x_2 = y_1, x_1x_3 = y_2, \dots, x_1 = y_{n(n-1)/2+1}, x_2 = y_{n(n-1)/2+2}, \dots, x_n = y_{n(n+1)/2}$ it is easy to go to the capability check and then – actually to the approximation of the linear target function of $n(n + 1)/2$ variables [9]. The number of variables becomes very large but iterative algorithms allow to cope with this task.

For nonlinear approximation in the general case regression trees [4] and the trained neural networks are applicable.

References

1. Antamoshkin A.N., Macich I.S. Search Algorithms for Conditional Pseudo-Boolean Optimization. *Control, Communications and Security Systems*, 2016, no. 1, pp. 103–145. (in Russian)
2. Mazurov V.I.D. Application of Methods of Theory of Pattern Recognition in the Optimal Planning and Management. *Proceeding of I-st all-Union Conference on Optimal Planning and National Economy Management*. Moscow, 1971, p. 49. (in Russian)
3. Rokach L., Maimon O.Z. *Data Mining with Decision Trees: Theory and Applications*. New Jersey, London, Singapore, Beijing, Shanghai, Hong Kong, Taipei, Chennai, World Scientific, 2014. DOI: 10.1142/9097
4. Loh W.-Y. Classification and Regression Trees. *Data Mining and Knowledge Discovery*, 2011, vol. 1, no. 14, pp. 14–23.
5. Kolmogorov A.N. *Algorithm, Information, Complexity*, Moscow, Znanie, 1991. (in Russian)
6. Donskoy V.I. Complexity of Families of Learning Algorithms and Estimation of the Nonrandomness of Extraction of Empirical Regularities. *Cybernetics and Systems Analysis*, 2012, vol. 48, no. 2, pp. 233–241. DOI: 10.1007/s10559-012-9402-2
7. Donskoy V.I. Capacity Estimates of the Main Classes of Empirical Generalizations Derived by the pVCD Method. *Scientific Notes of Taurida National V.I. Vernadsky University*, 2010, vol. 23 (62), no. 2, pp. 56–65. (in Russian)
8. Nilsson N.J. *Learning Machines*. N.Y., McGraw-Hill, 1965.
9. Bonates T.O., Hammer P.L. *Pseudo-Boolean Regression*. Rutcor Research Report RRR 3-2007. New Jersey, Rutgers Center for Operations Research of Rutgers University, 2007.

Received February 25, 2018

УДК 519.854

DOI: 10.14529/mmp180208

СИНТЕЗ ЭМПИРИЧЕСКИХ ПСЕВДОБУЛЕВЫХ МОДЕЛЕЙ ПО ПРЕЦЕДЕНТНОЙ ИНФОРМАЦИИ

В.И. Донской, Крымский федеральный университет, г. Симферополь,
Российская Федерация

Проблема принятия решений по частичной, прецедентной информации является важнейшей при создании систем искусственного интеллекта. По результатам наблюдений над поведением внешних объектов или систем необходимо на основе накопленной информации в виде конечного множества троек: «вектор состояния, значение качества функционирования объекта, бинарный индикатор допустимости этого состояния» синтезировать или, точнее, извлечь из данных математическую модель оптимизации объекта. Целью работы является создание и обоснование математических методов и алгоритмов, позволяющих синтезировать модели скалярной псевдоболевой оптимизации с ограничением в виде дизъюнктивной нормальной формы (ДНФ), используя указанную прецедентную информацию. Особенностью псевдоболевых оптимизационных моделей с сепарабельными целевыми функциями и ДНФ ограничением, имеющим ограниченную константой длину, является их полиномиальная разрешимость. Однако сложность

приведения задачи к форме с ДНФ ограничением в общем случае является экспоненциальной. При извлечении модели из данных ДНФ ограничение синтезируется приближенно, и сложность его аппроксимации оказывается полиномиальной, а число конъюнкций в извлеченной ДНФ не превышает числа примеров в начальной прецедентной информации. В статье показано, как использовать для построения дизъюнктивного ограничения бинарные решающие деревья. Предложены методы выявления свойств монотонности и линейности частично заданной целевой функции и алгоритмы решения задач псевдодобулевой скалярной оптимизации при наличии неполной, прецедентной начальной информации. Область применения полученных результатов – системы интеллектуального управления, интеллектуальные агенты. Несмотря на то, что модели управления, извлеченные из данных, являются приближенными, их применение может быть более успешным, чем использование менее реалистичных, не согласованных с моделируемым объектом и выбранных из субъективных соображений моделей.

Ключевые слова: псевдодобулева оптимизация; дизъюнктивное ограничение; машинное обучение; интеллектуальное управление; решающие деревья.

Литература

1. Антамошкин, А.Н. Поисковые алгоритмы условной псевдодобулевой оптимизации / А.Н. Антамошкин, И.С. Масич // Системы управления, связи и безопасности. – 2016. – Т. 1, № 1. – С. 103–145.
2. Мазуров, В.Д. Применение методов теории распознавания образов в оптимальном планировании и управлении / В.Д. Мазуров // Труды I Всесоюзной конференции по оптимальному планированию и управлению народным хозяйством. – М.: ЦЭМИ, 1971.
3. Rokach, L. Data Mining with Decision Trees: Theory and Applications / L. Rokach, O.Z. Maimon. – New Jersey; London; Singapore; Beijing; Shanghai; Hong Kong; Taipei; Chennai: World Scientific, 2014.
4. Loh, W.-Y. Classification and Regression Trees / W.-Y. Loh // Data Mining and Knowledge Discovery. – 2011. – V. 1, № 14. – P. 14–23.
5. Колмогоров, А.Н. Алгоритм, информация, сложность / А.Н. Колмогоров. – М.: Знание, 1991.
6. Донской, В.И. Сложность семейств алгоритмов обучения и оценивание неслучайности извлечения эмпирических закономерностей / В.И. Донской // Кибернетика и системный анализ. – 2012. – Т. 48, № 2. – С. 86–89.
7. Донской, В.И. Оценки емкости основных классов эмпирического обобщения, полученные рVCD методом / В.И. Донской // Ученые записки Таврического национального университета им. В.И. Вернадского. – 2010. – Т. 23 (62), № 2. – С. 56–65.
8. Нильсон, Н. Обучающиеся машины / Н. Нильсон. – М.: Мир, 1967.
9. Bonates, T.O. Pseudo-Boolean Regression / T.O. Bonates, P.L. Hammer // Rutcor Research Report RRR 3-2007. – New Jersey: Rutgers Center for Operations Research of Rutgers University, 2007.

Владимир Иосифович Донской, доктор физико-математических наук, профессор, кафедра информатики, Крымский федеральный университет (г. Симферополь, Российская Федерация), vidonskoy@mail.ru.

Поступила в редакцию 25 февраля 2018 г.