



Серия «Математика»  
2025. Т. 51. С. 116–129

Онлайн-доступ к журналу:  
<http://mathizv.isu.ru>

---

---

ИЗВЕСТИЯ

Иркутского  
государственного  
университета

---

---

Research article

УДК 004.89

MSC 68T35, 68T27

DOI <https://doi.org/10.26516/1997-7670.2025.51.116>

## Object Ontologies as *a Priori* Models for Logical-Probabilistic Machine Learning

Denis N. Gavrilin<sup>1</sup>, Andrei V. Mantsivoda<sup>1</sup>✉

<sup>1</sup> Irkutsk State University, Irkutsk, Russian Federation

✉ [andrei@baikal.ru](mailto:andrei@baikal.ru)

**Abstract.** Logical-probabilistic machine learning (LPML) is an AI method able to explicitly work with *a priori* knowledge represented in data models. This feature significantly complements traditional deep learning knowledge acquiring. *Object ontologies* are a promising example of such *a priori* models. They are an expanded logical analog of object oriented programming models. While forming the core of the bSystem platform, object ontologies allow solving the applied problems of high complexity, in particular, in the field of management. The combination of LPML and object ontologies is capable of solving the forecasting problems, the tasks of automated control, problem detection, decision making, and business process synthesis. The proximity of object ontologies to the LPML formalism due to the same semantic modeling background makes it possible to integrate them within a single hybrid formal system, which is presented in this paper. In the paper we introduce the approach to integration of these two formalisms and provide some algorithmic basis for the implementation of the resulting hybrid formalism on the bSystem platform.

**Keywords:** object ontology, logical-probabilistic inference, bSystem platform

**For citation:** Gavrilin D. N., Mantsivoda A. V. Object Ontologies as *a Priori* Models for Logical-Probabilistic Machine Learning. *The Bulletin of Irkutsk State University. Series Mathematics*, 2025, vol. 51, pp. 116–129.

<https://doi.org/10.26516/1997-7670.2025.51.116>

Научная статья

Объектные онтологии как априорные модели логико-вероятностного вывода

Д. Н. Гаврилин<sup>1</sup>, А. В. Манцивода<sup>1</sup>✉

<sup>1</sup> Иркутский государственный университет, Иркутск, Российская Федерация  
✉ andrei@baikal.ru

**Аннотация.** Логико-вероятностное машинное обучение (ЛВМО) – метод искусственного интеллекта, который способен работать не только со знаниями, полученными через глубокое обучение, но и с априорными знаниями, явно представленными в виде моделей данных. Перспективный пример таких моделей – объектные онтологии, которые являются расширенным логическим аналогом объектно-ориентированных моделей в программировании. Реализованные в рамках платформы bSystem объектные онтологии позволяют решать прикладные задачи высокой сложности, например в области управления. Комбинация возможностей ЛВМО и объектных онтологий позволяет решать задачи прогнозирования, автоматического контроля, выявления проблем, поддержки принятия решений и синтеза бизнес-процессов, направленных на достижение целей. Близость формализмов ЛВМО и объектных онтологий, основанных на семантическом моделировании, позволяет интегрировать их в рамках единой гибридной формальной системы, которая представлена в данной работе. В ней описывается механизм интеграции этих двух систем и закладываются алгоритмические основы реализации получившегося гибридного формализма в рамках платформы bSystem.

**Ключевые слова:** объектная онтология, логико-вероятностный вывод, платформа bSystem

**Ссылка для цитирования:** Gavrilin D. N., Mantsivoda A. V. Object Ontologies as *a Priori* Models for Logical-Probabilistic Machine Learning // Известия Иркутского государственного университета. Серия Математика. 2025. Т. 51. С. 116–129.  
<https://doi.org/10.26516/1997-7670.2025.51.116>

## 1. Introduction

The paper is focused on developing a hybrid formalism that combines object ontologies and logical probabilistic machine learning (LPML). This work is fulfilled in line with a strategic direction related to the use of semantic methods to application (apps) development. We implement these techniques on bSystem, a platform intended for web application development, mainly in business administration and management.

Earlier we have developed the technology of *object ontologies* (also called document models) [6], which are simple logical systems having the expressive power close to that of object-oriented programming models, but based on a logical formalism. An object ontology can be viewed as a basic model that specifies the facts, terms, objects and relationships operating in a subject domain. Reliance on logic makes such a description transparent and semantically manageable.

An alternative – programmer’s – interpretation of object ontologies in bSystem allows them to serve as data storages for applications [5]. Ontologies can evolve over time through transactional mechanisms [6].

A visualization method of model development [5] allowed us to implement within bSystem a new low-code technology (*declarative low-code*)

based on automated code synthesis from the ontology. It increases developer productivity by 10–15 times.

In its turn, *LPML* [8] is a machine learning method that can bring semantic (qualitative) analytics to developed applications. *Semantic analytics* complements the operational layer and quantitative business intelligence provided by object ontologies. This three-layer structure of applications accompanied by a single data model, within which the layers interact with each other, allow such applications to be characterized as ‘*intelligent*’, and capable not only of supporting operations and business processes, but also of self-monitoring, semantic process analysis, adaptation, forecasting, and decision support.

Unlike neural networks, the result of deep learning in *LPML* is a set of logical probabilistic production rules postulating the laws of the subject domain. Thus, the process of deep learning in *LPML* can be understood as revealing implicit knowledge about the subject domain, and the system of rules itself can be considered as an automatically generated *expert system* intended for helping in forecasts, decision-making, situation monitoring, etc.

This feature explains some impressive capabilities of *LPML*:

- *LPML* can accumulate knowledge not only through deep learning, but also discover it as direct *a priori* knowledge supplied by a formal model.
- *LPML* is able to explain its machine learning results (e.g., through interpreting logical rules into human texts).
- The accumulated rules can be automatically analyzed within meta-knowledge reasoning. Such activity corresponds to human holistic analysis of the overall situation.

The integration of object ontologies and *LPML* within a unified formalism is feasible because both *LPML* and object ontologies are based on the same semantic modeling methodology [3]. It also provides a single data space, in which the operational, analytical and AI layers of applications interact.

The aim of this paper is to develop a formal system that integrates object ontologies and *LPML* in the way that allows us to apply this formalism for ‘intelligent’ apps development, that is, such apps that combine operational and intelligent features. Our plan for the development of such a formalism is as follows:

- We nominate the object ontology as the *a priori* knowledge model, on which *LPML* deep learning is based.
- We enrich the language of object ontologies with primitives that allow introducing logical probabilistic rules.
- We introduce a knowledge discovery method as an *LPML* deep learning algorithm over ontology data.

- Finally, we implement the logical and probabilistic superstructure over object ontologies on the bSystem platform and provide an environment for developing intelligent apps.

In future, we plan to develop a ‘meta-level’, which analyzes the resulting production rules and builds up working strategies basing on this analysis. In particular, if a task statement is formulated as the achievement of a specific goal described by a formula  $F$ , then the solution of the problem can be interpreted as a chain of transactions performed in the ontology that turns  $F$  into a production rule with a very high level of probability (e.g.,  $> 0.999$ ). Probably, the development of this meta-level could pave the way towards the creation of a variation of artificial general intelligence (AGI) based on the integration of object ontologies and LPML.

## 2. A Simplified Model of Knowledge Representation

A logical-probabilistic approach to machine learning that provides discovering probabilistic laws on arbitrary data has been developed by E.E. Vityaev [8]. In [1] it is shown that this approach can be useful in a variety of applications.

In [8], the input data are represented in the form of a relational table whose rows correspond to objects and columns to object properties. The algorithm described in that article can be applied to data of various types.

In this paper we use a simplified data model based on Boolean data types. That is, let  $B$  be a table whose rows represent some objects and columns represent properties, and these properties can have only two states  $\mathbb{1}$  and  $\mathbb{0}$ . We can represent the table as matrix (2.1)

$$B = \begin{pmatrix} B_{11} & B_{12} & \dots & B_{1n} \\ B_{21} & B_{22} & \dots & B_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ B_{m1} & B_{m2} & \dots & B_{mn} \end{pmatrix} \quad (2.1)$$

Let  $B = \{B_1, B_2, \dots, B_m\}$ , where  $B_i$  is  $i$ -th row of the table, and  $B_i = \{B_{i1}, B_{i2}, \dots, B_{in}\}$ .  $B_{ij}$  is the value in the cell located at the intersection of the  $i$ -th row and  $j$ -th column. We also introduce a predicate  $P_j^\varepsilon$  such that:

$$P_j^\varepsilon(B_i) = \begin{cases} B_{ij}, & \text{if } \varepsilon = 0 \\ \neg B_{ij}, & \text{if } \varepsilon = 1 \end{cases}$$

Let us consider the algorithm for discovering laws in this data model. Let  $U(Th) = \{A_1, A_2, \dots, A_{2n}\}$  be the set of all literals of the form  $A_i = P_k^\varepsilon$ , where  $k$  is the column index,  $\varepsilon \in \{0, 1\}$ . We also introduce a target predicate  $A_0 = P_s^\varepsilon$ , in which  $s$  is the index of the target column.

We call a *probabilistic law* [8] a rule  $A_{k_1} \wedge A_{k_2} \wedge \dots \wedge A_{k_n} \rightarrow A_{k_0}$ , whose conditional probability is defined and is strictly greater than the conditional probability of any of the subrules, that is, satisfying conditions (2.2) and (2.3):

$$p(A_{k_0}|A_{k_1} \wedge \dots \wedge A_{k_n}) > 0 \quad (2.2)$$

$$p(A_{k_0}|A_{k_1} \wedge \dots \wedge A_{k_n}) > p(A_{k_0}|A_{t_1} \wedge \dots \wedge A_{t_m}) \\ \forall t_1, \dots, t_m \subset \{k_1, \dots, k_n\} \quad (2.3)$$

For discovering such laws we can apply the algorithm from [2] including forecast generation and decision making.

Note that more advanced models can be converted into the simplified data model introduced above. For instance, let us consider the initial relational data model from [2]. Suppose, the original data is represented as a relational table  $D$ , the rows of which correspond to objects and the columns correspond to object attributes, i.e.  $D = \{D_1, \dots, D_n\}$ , where the row  $D_i$  represents the  $i$ -th object,  $D_i = \{D_{i1}, \dots, D_{im}\}$ . Here  $D_{ij}$  is the value of the  $j$ -th property of the  $i$ -th object.

Let  $P = \{P_1, \dots, P_n\}$  be a set of predicates. By sequentially applying the predicates to all data rows of table  $D$ , we form a new table  $B$ , where for each row  $D_i$ ,  $B$  contains the row  $B_i = \{B_{i1}, \dots, B_{in}\}$  such that  $B_{ij}$  is equal to the value of  $P_j(D_i)$ . In this case  $B$  contains truth values and has the form (2.1).

### 3. Probabilistic Object Ontologies

#### 3.1. OBJECT ONTOLOGY MODEL

An object ontology is a simple logical model. In its expressiveness, object ontologies are a logical analogue of object models in programming. Object ontologies (also referred as document models) were considered in a number of papers [4–6]. Processes are defined over ontologies [7], which can change their state and content in time. In this paper we consider the ontology at an instant moment, so suppose that its content is immutable.

Let  $\Omega$  be a set of elements. We call a *sequence* (over  $\Omega$ ) the expression

$$(e_1, \dots, e_m), e_i \in \Omega$$

We denote by  $()$  the *empty sequence* (which contains no elements). To determine the number of elements in a sequence we define the notion of *cardinality* and introduce the following cardinalities:

- $()$  is the empty sequence
- $?$  is a sequence containing zero or one element
- $!$  is a sequence containing strictly one element
- $+$  is a sequence containing at least one element

– \* is a sequence containing an arbitrary number of elements.

The *set of names* is a countable collection of constants  $N = \{n_1, n_2, \dots\}$ . This set consists of two disjoint subsets: the set of class names  $N_F$  and the set of object field names  $N_D$ .

A *field type* is a tuple

$$\mathbf{d} = \langle d, \mathbf{c} \rangle$$

where  $d \in N_D$  is the field name and  $\mathbf{c}$  is its cardinality. The *field* is represented as a tuple

$$\langle d, v \rangle$$

where  $d \in N_D$  is the field name, and  $v$  is its value as a sequence.

An *object class* is a tuple

$$\mathbf{f} = \langle f, \mathbf{d}_1, \dots, \mathbf{d}_n \rangle$$

where  $f \in N_F$  is an object class, and  $\mathbf{d}_1, \dots, \mathbf{d}_n$  is a finite set of field types.

An *object* is defined as a tuple

$$\mathbf{o} = \langle f, id, \mathfrak{d}_1, \dots, \mathfrak{d}_n \rangle$$

where  $f \in N_F$  is the class name of the object,  $id$  is its unique identifier, and  $\mathfrak{d}_1, \dots, \mathfrak{d}_n$  is a finite set of object fields such that their names are identical to the names from the field types of class  $\mathbf{f}$ .

**Definition 1.** An object ontology is a pair containing a domain (a set of valid field values) and a finite set of object classes  $D = \langle \Omega, \mathbf{f}_1, \dots, \mathbf{f}_m \rangle$ .

A *state of an object ontology* is a finite set of objects  $\{\mathbf{o}_1, \dots, \mathbf{o}_n\}$ , where each  $\mathbf{o}_i$  is an object of some class  $\mathbf{f}_j \in \{\mathbf{f}_1, \dots, \mathbf{f}_m\}$ , and its fields have values from domain  $\Omega$ .

### 3.2. THE LANGUAGE OF LOGICAL STATEMENTS OVER ONTOLOGY

In order to most effectively use the logical-probabilistic ML over object ontologies, it is necessary to convert data from ontologies into a data model compatible with the logical-probabilistic method. For this, we introduce a special query language, which will help us to formulate logical statements over ontologies.

A *retrieval operator* returns the value of a field named  $d$  in an object denoted by  $\mathbf{x}$ :

$$get(\mathbf{x}, d).$$

Let  $\mathbb{M} = \langle \Omega, N_F, N_D; get, exists, =, <, > \rangle$  be a *model* augmented with *exists*, equality and inequality relations, and  $\mathbf{x}$  a variable. *exists*( $\mathbf{x}, d$ ) is true if the value of the field named  $d$  is not empty in  $\mathbf{x}$ . A special query language is defined as follows:

- $x$  is a term
- $get(x, d)$  is a term
- $e$ , where  $e \in \Omega$  is also a term
- the constants (identifiers of objects)  $id_i$  are terms
- if  $t_1, t_2$  are terms, then expressions of the form
  - $t_1 = t_2$ ,  $t_1$  equals  $t_2$
  - $t_1 < t_2$ ,  $t_1$  is less than  $t_2$
  - $t_1 > t_2$ ,  $t_1$  is greater than  $t_2$
  - $exists(x, d)$
 are *atomic formulas*.
- an atomic formula is a *formula*.
- if  $F_1, F_2$  are formulas then the expressions
  - $(F_1)$
  - $F_1 \wedge F_2$
  - $F_1 \vee F_2$
  - $\neg F_1$
 are also *formulas*.

If  $t$  is a term then the *substitution*  $F|_x^t$  is a formula equal to the formula  $F$ , in which all occurrences of the variable  $x$  are replaced by the term  $t$ .

### 3.3. THE INTERPRETATION OF PROBABILISTIC RULES

Let us define the *interpretation function*  $E_M$  as follows:

$$\begin{aligned}
 E_M(id) &\equiv \mathbf{o}, \text{ where } \mathbf{o} \text{ is the object with identifier } id \\
 E_M(get(id, d)) &\equiv \{E(id_i) \mid M \models get(id, d) = v \text{ and } id_i \in v\} \\
 E_M(t_1 = t_2) &\equiv M \models E(t_1) = E(t_2) \\
 E_M(t_1 < t_2) &\equiv M \models E(t_1) < E(t_2) \\
 E_M(t_1 > t_2) &\equiv M \models E(t_1) > E(t_2) \\
 E_M(exists(id, d)) &\equiv \begin{cases} \mathbf{1}, & M \models \exists v : (get(id, d) = v \wedge v \neq ()) \\ \mathbf{0}, & \text{otherwise} \end{cases} \\
 E_M((F_1)) &\equiv E(F_1) \\
 E_M(\neg F_1) &\equiv \begin{cases} \mathbf{1}, & E(F_1) \equiv \mathbf{0}, \\ \mathbf{0}, & E(F_1) \equiv \mathbf{1}, \end{cases} \\
 E_M(F_1 \wedge F_2) &\equiv \begin{cases} \mathbf{1}, & \text{if } E(F_1) \equiv \mathbf{1} \text{ and } E(F_2) \equiv \mathbf{1}, \\ \mathbf{0}, & \text{otherwise,} \end{cases} \\
 E_M(F_1 \vee F_2) &\equiv \begin{cases} \mathbf{1}, & \text{if } E(F_1) \equiv \mathbf{1} \text{ or } E(F_2) \equiv \mathbf{1}, \\ \mathbf{0}, & \text{otherwise.} \end{cases}
 \end{aligned}$$

Let  $\{F_0, F_1, \dots, F_n\}$  be formulas with only one variable  $x$ . Then a *rule*  $R$  is an implication of the form:

$$R : F_1 \wedge \dots \wedge F_n \rightarrow F_0 \quad (3.1)$$

By  $N(F)$  we denote the set of such objects  $\mathbf{o}_i$  that

$$N(F) = \{\mathbf{o}_i \mid \mathbb{M} \models E(F|_x^{id_i}) \equiv \mathbb{1} \text{ and } E_{\mathbb{M}}(id_i) = \mathbf{o}_i\},$$

Then the *conditional probability*  $p(R)$  of the rule  $R$  is the ratio of the number of objects on which both the premise and the conclusion are true to the number of objects on which the premise is true:

$$p(R) = p(F_0 | F_1 \wedge \dots \wedge F_n) = \frac{N(F_1 \wedge \dots \wedge F_n \wedge F_0)}{N(F_1 \wedge \dots \wedge F_n)}, \quad (3.2)$$

where  $N(F_1 \wedge \dots \wedge F_n) > 0$ .

### 3.4. PROBABILISTIC LAWS OF OBJECT ONTOLOGIES

By analogy with the LPML model, we introduce a notion of a *probabilistic law* over object ontologies. The logical-probabilistic approach looks quite promising for object ontologies. Real-life data that we work with can have errors and uncertainties, so strict logical laws can not be used effectively under such conditions. On the other hand, probabilistic laws are more flexible and sustainable in the context of noise, blurred data and info gaps.

But we should take into account the following aspects. On the one hand, we are interested only in those discovered rules, which have sufficiently high conditional probability. A rule with conditional probability 0.1 (works for at most each 10-th object) is not interesting, because only shows that the dependence of the conclusion on the premises is very weak at best.

On the other hand, a rule should not be overloaded with premises in order to be general enough. We denote  $p_{min}$  the minimum value of the conditional probability, which makes a rule a probabilistic law.

**Definition 2** (Probabilistic Law). A probabilistic law is a pair  $\langle R, p_{min} \rangle$ , for which the following conditions hold:

$$p(R) \geq p_{min} \quad (3.3)$$

$$p(F_0 | F_1 \wedge \dots \wedge F_n) > p(F_0 | F_{k_1} \wedge \dots \wedge F_{k_m}), \quad \forall k_1, \dots, k_m \subset \{1, \dots, n\} \quad (3.4)$$

Condition (3.3) represents the requirement imposed on the conditional probability. Condition (3.4) ensures that there are no insignificant premises in the rule. If a rule have the same conditional probability with and without a premise  $F_i$  then this  $F_i$  does not bring additional knowledge to our model. It only makes the premises stricter and less general, and thus narrows the scope of applicability.



**Definition 3** (Probabilistic Object Ontology). A probabilistic object ontology is a triplet  $\langle \Omega, \mathbf{f}_1, \dots, \mathbf{f}_m, L_1, \dots, L_k \rangle$ , where  $\langle \Omega; \mathbf{f}_1, \dots, \mathbf{f}_m \rangle$  is an ontology, and  $L_1, \dots, L_k$  are a collection of probabilistic laws defined on this ontology.

The definition of a state of a probabilistic object ontology is similar to that of an ordinary object ontology, that is, a finite set of objects  $\{\mathbf{o}_1, \dots, \mathbf{o}_m\}$ , where each  $\mathbf{o}_i$  is an object of some ontology class, and the fields have values from domain  $\Omega$ .

#### 4. Integration of a Logical-probabilistic Inference Algorithm into an Object Ontology Model

Let

$$\mathfrak{F} = \{F_1, \dots, F_n\}$$

be some set of formulas containing variable  $\varkappa$  and

$$\mathfrak{D} = \{\mathfrak{o}_1, \dots, \mathfrak{o}_m\}$$

some set of objects. Having interpreted each formula by substituting  $\varkappa$  with identifier  $id_i$  for each object from  $\mathfrak{D}$ , we form the *state matrix* of the object ontology:

$$\mathfrak{B} = \begin{pmatrix} B_{11} & B_{12} & \dots & B_{1n} \\ B_{21} & B_{22} & \dots & B_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ B_{m1} & B_{m2} & \dots & B_{mn} \end{pmatrix}$$

where  $B_{ij} = E_{\mathfrak{M}}(F_j|_{\varkappa}^{id_i})$  and  $E_{\mathfrak{M}}(id_i) = \mathfrak{o}_i$ .

The matrix  $\mathfrak{B}$  is formed by interpreting the formulas from  $\mathfrak{F}$ , and thus all its values are truth values reflecting the qualities of objects from  $\mathfrak{D}$  in model  $\mathfrak{M}$ .  $\mathfrak{B}$  has form (2.1) and, thus, its data can be used for discovering logical-probabilistic laws using the algorithm mentioned in section 2.

The selection of formulas from  $\mathfrak{F}$  allows us to focus our interest on a specific segment of knowledge we want to discover in the ontology. We can consider each  $F_i \in \mathfrak{F}$  as the definition of a concept, the behavior of which we want to understand better. We also can use these formulas to discover generalized knowledge, say, by grouping real values into ranges and considering the values belonging to the same range as indistinguishable. The selection of such  $F_i$ s is one of the ways to configure the logical-probabilistic machine learning solver and focus it on solving a specific task. The other way to configure the search is the selection of the set of objects  $\mathfrak{D}$ . In real-life problems the exhaustive search over all ontology is frequently too hard algorithmically, so we need to select those subsets of objects that, as we think, correctly represent the general concepts, behavior of which we want to investigate.

**Definition 4.** Let  $\mathfrak{F} = \{F_1, \dots, F_n\}$  and  $\mathfrak{D} = \{\mathfrak{o}_1, \dots, \mathfrak{o}_m\}$ . An ontology segment based on  $\mathfrak{F}$  and  $\mathfrak{D}$  is such a structure  $\langle \mathfrak{D}, \mathfrak{f}_1, \dots, \mathfrak{f}_m \rangle$  that can interpret formulas from  $\mathfrak{F}$ .

A ML procedure can be applied to an ontology segment if all class symbols occurring in the formulas of  $\mathfrak{F}$  are among  $\{\mathfrak{f}_1, \dots, \mathfrak{f}_m\}$ .

**Lemma.** *An ontology segment is itself an object ontology.*

An ontology segment has the same structure as the ontology from which it was extracted, and since no additional constraints are placed on the ontology, the proof of the lemma follows from definitions 1 and 4.

#### 4.1. INTERPRETATION OF DISCOVERED LAWS

The laws obtained by the machine learning algorithm are built from the formulas of the set  $\mathfrak{F}$  and thus can be interpreted in the context of the initial object ontology. Suppose that as a result we have obtained some probabilistic law

$$A_{k_1} \wedge \dots \wedge A_{k_n} \rightarrow A_{k_0}$$

Since by definition each element  $A_i$  of the premise  $A_{k_1} \wedge \dots \wedge A_{k_n}$  represents some formula or its negation within the ontology  $\mathbb{M}$

$$A_r = P_j^\varepsilon(B_i) = F_j^\varepsilon = \begin{cases} F_j, & \varepsilon = 0, \\ \neg F_j, & \varepsilon = 1 \end{cases}$$

then within the object ontology the probabilistic law can be represented by a rule in the form (3.1)

$$F_{j_1}^{\varepsilon_1} \wedge \dots \wedge F_{j_n}^{\varepsilon_n} \rightarrow F_{j_0}^{\varepsilon_0}.$$

Further, each row  $B_i$  of matrix  $B$  can be matched with some object  $\mathfrak{o}_i$ . Then

$$A_r(B_i) \Leftrightarrow E_{\mathbb{M}}(F_j^\varepsilon |_{\mathfrak{x}}^{id_i}) \equiv \mathbb{1}$$

Hence the conditional probability of the rule in terms of LPML is equal to the same probability within the object ontology:

$$p(A_{k_0} | A_{k_1} \wedge \dots \wedge A_{k_n}) = p(F_{j_0}^{\varepsilon_0} | F_{j_1}^{\varepsilon_1} \wedge \dots \wedge F_{j_n}^{\varepsilon_n})$$

Now if we select only rules with the conditional probability greater than some threshold  $p_{min}$  defined for the object ontology then their counterparts in the ontology are also the probabilistic laws since condition (3.3) evidently holds and (3.4) follows from (2.3) in the definition of probabilistic laws.

Thus, if we need to estimate the behavior of the object ontology in the terms of formulas  $\mathfrak{F}$  then using the technique above we can convert them into the form, for which the machine learning technique from [2] is

applicable. Having found the probabilistic laws we can convert them back to their counterparts working in the context of the object ontology. These counterparts also appear to be probabilistic laws.

#### 4.2. AN EXAMPLE

Let us consider a simple example about people. We have three properties (fields) to characterise them: **hairColor**, **age** and **occupation**. The basic formulas  $\{F_1, \dots, F_{10}\}$  select people with a specific hair color ( $F_1, \dots, F_3$ ), the age range ( $F_4, \dots, F_7$ ), and the occupation ( $F_8, \dots, F_{10}$ ):

$$\begin{aligned} F_1 &\equiv \text{get}(x, \text{hairColor}) = \text{"brunette"} \\ F_2 &\equiv \text{get}(x, \text{hairColor}) = \text{"blond"} \\ F_3 &\equiv \text{get}(x, \text{hairColor}) = \text{"green"} \\ F_4 &\equiv \text{get}(x, \text{age}) < 16 \\ F_5 &\equiv \text{get}(x, \text{age}) \geq 16 \wedge \text{get}(x, \text{age}) < 25 \\ F_6 &\equiv \text{get}(x, \text{age}) \geq 25 \wedge \text{get}(x, \text{age}) < 50 \\ F_7 &\equiv \text{get}(x, \text{age}) \geq 50 \\ F_8 &\equiv \text{get}(x, \text{occupation}) = \text{"musician"} \\ F_9 &\equiv \text{get}(x, \text{occupation}) = \text{"scientist"} \\ F_{10} &\equiv \text{get}(x, \text{occupation}) = \text{"student"} \end{aligned}$$

Suppose that we have a specific ontology that describes facts about specific people, and for these people the following probabilistic laws have been generated as a result of machine learning in the ontology:

	$L_i$	$p(H_i)$
$L_1$	$F_4 \rightarrow F_{10}$	0.98
$L_2$	$F_3 \wedge F_5 \rightarrow \neg F_9$	0.95
$L_3$	$F_7 \rightarrow \neg F_3$	0.999

Using linguistic patterns we can convert these laws into a text in the natural language, and this text can explain us what is going on in our model:

- $L_1$ : if a person is under 16 then this person is a student, with probability 98% of being a scholar,
- $L_2$ : if a person has green hair and is between 16 and 25 years old then he/she is not doing science with high probability (95%),
- $L_3$ : if a person is over 50, it is almost certain (with probability 99.9%) that this person does not have green hair.

## 4.3. PREDICTION AND DECISION MAKING

Now consider how the prediction algorithm described in [2] can be applied to probabilistic object ontologies. A *prediction* is a formula used to predict the value of a given object property. A prediction is valid if it forecasts the property value with some high probability.

Suppose that we have discovered in an ontology some set of probabilistic laws  $\mathbb{L} = \{L_1, \dots, L_n\}$ ,  $\circ$  is an ontology object, denoted by constant  $id$ , and we want to predict the outcome of an  $\circ$ 's property with  $k$  possible alternatives that expressed by formulas  $\{F_1, \dots, F_k\}$ , respectively. Let us partition the original set of laws into disjoint subsets  $\{\mathbb{L}^{F_1}, \dots, \mathbb{L}^{F_k}\}$  selecting only those laws whose premises are true on object  $\circ$  and whose conclusions are equal to  $F_i$ , that is,

$$\mathbb{L}^{F_i} = \{L \mid L \in \mathbb{L} \wedge Cn(L) = F_i \wedge E_{\mathfrak{M}}(Pr(L)|_{\mathfrak{x}}^{id}) \equiv \mathbb{1}\},$$

where  $Pr(L)$  is the premise and  $Cn(L)$  the conclusion of the implication  $L$ . For each  $\mathbb{L}^{F_i}$  we determine the probability that the conclusion  $F_i$  is true as the maximum conditional probability among all laws included in  $\mathbb{L}^{F_i}$ :  $p(\mathbb{L}^{F_i}) = \max\{p(L) : L \in \mathbb{L}^{F_i}\}$ .

Now, two options are available. In case if the property allows several outcomes simultaneously (several  $F_i$ s hold on the same object) then all  $F_i$  such that  $p(L^{F_i}) > \delta$ , where  $\delta \in [0, 1]$  is some minimum threshold, can form a prediction for the object  $\circ$ .

If the property can have at most one outcome (so at most one  $F_i$  can be true on the same object simultaneously) then it is necessary to make an assessment of the consistency of the obtained result. For this, we define the threshold of acceptable consistency  $\delta > 0$  and then look for such a formula  $F \in \{F_1, \dots, F_n\}$ , the probability of which is maximum:

$$p(L^F) = \max\{p(L^{F_i}) \mid i \in \{1..n\}\}$$

Now, the obtained forecast result is consistent only if the minimum difference between  $p(L^F)$  and other  $p(L^{F_i})$  is not less than the given consistency threshold  $\delta$ , i.e.

$$\delta \leq \min\{p(L^F) - p(L^{F_i}) \mid i \in \{1..n\}, L^F \neq L^{F_i}\}$$

So, a prediction can be considered as valid only if we have a clear champion among all possible outcomes, otherwise the prediction fails. And the reliability of a successful forecast depends on its probability.

In case of a mixed situation when some formulas from the initial set should be pairwise contradictory and others can be true simultaneously, it is necessary to partition this set into pairwise disjoint subsets in which either all formulas are pairwise contradictory or can hold simultaneously, and perform the forecasting procedure for each subset separately.

Both situations, when the formulas are pairwise contradictory and when they are not are quite practical. For instance, if we have continuous data with possibly infinite number of values (like speed, weight etc.), it is convenient to partition them into a finite collection of subsets, so each object can belong to no more than one subset (like we split child age into five stages: newborn, infant, toddler, preschool, school-age). And we should ensure that any object belongs to only one subset. On the other hand, the properties can overlap like hobbies: I like music, tennis and hiking. This means that several formulas describing hobbies can hold for the same person.

## 5. Conclusion

In this paper we primarily consider only the static properties of probabilistic object ontologies, that is, their states at instant moments and do not take into account their capability to develop in time due to transactions and processes [7]. But we think that the approach to machine learning considered above has significant potential of acting in changing contexts. For instance, periodical re-learning can be established that can reveal conceptual changes in the ontology, and we can discover those changes via the probabilistic law meta-analysis. In particular, this allows us to implement the task approach [9] by targeting the goal using logical-probabilistic criteria of the task solution and achieving this solution by finding the chain of transactions that enable the ontology to comply with these criteria with sufficiently high probability. That is, given a set of logical-probabilistic solution criteria  $L_1, \dots, L_k$  for a task  $T$  and a set of reals  $p_1, \dots, p_k$ ,  $0 \leq p_i \leq 1$ , then a sequence of ontology transactions  $t_1, \dots, t_m$  is a solution of the task  $T$  if the consecutive application of these transactions to the ontology will change it in such a way that

$$p(L_i) \geq p_i \text{ for } \forall i \in \{1, \dots, k\}.$$

The probabilistic laws also can serve for making some qualitative assessment of the ontology data at each instance of time. Some laws also can appear and disappear during the ontology lifetime indicating some external influences on the ontology. So we plan to investigate this and other topics in more detail together with implementing on the platform bSystem the logical-probabilistic machine learning approach described above.

## References

1. Demin A.V., Ponomaryov D.K. Machine Learning with Probabilistic Law Discovery: a Concise Introduction. *The Bulletin of Irkutsk State University. Series Mathematics*, 2023, vol. 43, pp. 91–109. <https://doi.org/10.26516/1997-7670.2023.43.91>

2. Demin A.V., Vityaev E.E. Relyatsionnyy podkhod k izvlecheniyu znaniy i ego primeneniya [Relational Approach to Knowledge Discovery and its Applications]. *Materialy Vserossiyskoy konferentsii s mezhdunarodnym uchastiem "Znaniya – Ontologii – Teorii"* [Proc. ZONT Conference], Novosibirsk, 2013, vol. 1, pp. 122–130. (in Russian)
3. Ershov Yu.L., Goncharov S.S., Sviridenko D.I. Semantic Foundations of Programming. *Fundamentals of Computation Theory: Proc. Intern. Conf. FCT 87*, Lect. Notes Comp. Sci. Kazan, 1987, vol. 278, pp. 116–122. [https://doi.org/10.1007/3-540-18740-5\\_28](https://doi.org/10.1007/3-540-18740-5_28)
4. Gavrilin D.N., Kustova I.A., Mantsivoda A.V. Object Modelas as Microservices: a query language. *The Bulletin of Irkutsk State University. Series Mathematics*, 2022, vol. 42, pp. 121–137. <https://doi.org/10.26516/1997-7670.2022.42.121> (in Russian)
5. Gavrilina D.Je. and Mantsivoda A.V. Low-code and Object Spreadsheets. *The Bulletin of Irkutsk State University. Series Mathematics*, 2022, vol. 40, pp. 93–103. <https://doi.org/10.26516/1997-7670.2022.40.93> (in Russian)
6. Mantsivoda A.V., Ponomaryov D.K. Towards Semantic Document Modelling of Business Processes. *The Bulletin of Irkutsk state university. Series Mathematics*, 2019, vol. 29, pp. 52–67. <https://doi.org/10.26516/1997-670.2019.29.52>
7. Mantsivoda A.V., Ponomaryov D.K. On Termination of Transactions over Semantic Document Models. *The Bulletin of Irkutsk state university. Series Mathematics*, 2020, vol. 31, pp. 111–131. <https://doi.org/10.26516/1997-7670.2020.31.111>
8. Vityaev E.E. *Logiko-verojatnostnye metody izvlecheniya znaniy iz dannyh i kompjuternoe poznanie* [Logical-probabilistic methods of knowledge extraction from data and computer cognition]. Dr. sci. diss. Novosibirsk, 2006, 170 p. (in Russian)
9. Vityaev E.E., Goncharov S.S., Gumirov V.S., Mantsivoda A.V., Nechesov A.V., Sviridenko D.I. Task Approach: On the Way to Trusting Artificial Intelligence. *World Congress Theory, Algebraic Biology, Artificial Intelligence: Mathematical Foundations and Applications*, 2023, pp. 179–243. <https://doi.org/10.18699/sblai2023-41>

## Об авторах

**Гаврилин Денис Николаевич**,  
аспирант, Иркутский  
государственный университет,  
Иркутск, 664003, Российская  
Федерация

**Манцивода Андрей Валерьевич**,  
д-р физ.-мат. наук, проф.,  
Иркутский государственный  
университет, Иркутск, 664003,  
Российская Федерация,

## About the authors

**Denis N. Gavrilin**, Postgraduate,  
Irkutsk State University, Irkutsk,  
664003, Russian Federation

**Andrei V. Mantsivoda**, Dr. Sci.  
(Phys.-Math.), Prof., Irkutsk State  
University, Irkutsk, 664003, Russian  
Federation

*Поступила в редакцию / Received 15.10.2024*

*Поступила после рецензирования / Revised 25.11.2024*

*Принята к публикации / Accepted 27.11.2024*