LOGICAL AND PRECEDENT-RELATED REPRESENTATION OF INFORMATION IN PATTERN RECOGNITION

O.V.Shut
BSU
Minsk, BELARUS
e-mail: olgashut@tut.by

Abstract

The algebra of objects used in problems of pattern recognition is considered in the paper. The connection between operations on objects and Boolean operations is determined. The completeness of the set of operations on objects is shown. Algorithms for the conversion of ways of representating information are provided and analysed.

1 Introduction

The main concepts of pattern recognition are essential for development of modern information systems. Corresponding problems are objects of interdisciplinary research in information theory, statistics, physics, chemistry, linguistics, psychology, biology, physiology and medicine [2].

There are two main methods for representating prior knowledge in supervised pattern recognition:

- logical: representation as a rule;
- precedent-related: representation as a set of objects.

One of the actual problems is development of an algebraic construction for conversion from rules to the corresponding set of objects and vice versa.

2 Basic Concepts

The following way of describing objects [3] will be used in this paper. Let an object have n signs which possess a finite number of values. These signs will be called *nominal* [1]. Let S_i denote a set of signs from which the ith sign is chosen and let D_i denote a set of values of this sign. Let an *object* be defined as the following mapping:

$$p : S_1 \times S_2 \times ... \times S_n \to D_1 \times D_2 \times ... \times D_n$$

An object p which has signs $s_1, s_2, ..., s_n$ with correspondent values $d_1, d_2, ..., d_n$ will be written as follows:

$$p(s_1, s_2, ..., s_n) = (d_1, d_2, ..., d_n)$$

Two objects will be considered as *equal*, if sets of their signs are equal, and so do corresponding values of their signs.

Let a *collection of objects* be a set of objects where all objects have the same signs. Collections P and Q will be considered as *equal* if for every object p in P one can find such an object q in Q that $p = q$ and vice versa.

Operations of adding, multiplying and neglecting objects were introduced and some of their properties proved in [3].

Equivalence of operations on objects with nominal signs and Boolean operations is explored in this paper. Completeness of the set of operations on objects is shown. Algorithms for conversion from logical to precedent-related and from precedent-related to logical ways of representating information are provided and analysed.

3 Algebra of Objects and Boolean Algebra

The correspondence between operations in the algebra of objects and Boolean operations will be explored. Let all objects be numbered. For every object p consider its following code:

$$C(p) = (0...010...0)$$

Here 1 occupies the position which number is equal to the number of p.

For every collection f objects consider its following code:

$$C(P) = \bigvee_{p \in P} C(p)$$

The correspondence between collections (objects) and their codes is bijective.

Operations on collections (objects) and Boolean operations will be considered as *equal*, if for every two collections P and Q the following equations are fulfilled:

$$C(P) \wedge C(Q) = C(P \wedge Q) \tag{1}$$

$$C(P) \vee C(Q) = C(P \vee Q) \tag{2}$$

$$\overline{C(P)} = C(\overline{P}) \tag{3}$$

Theorem 1. *Operations $\neg, \wedge, \vee$ in the algebra of objects with nominal signs and Boolean operations $\neg, \wedge, \vee$ are equal.*

Completeness of the set of operations on collections (objects) will be explored now. It was shown [3] that every collection can be represented as a sum of its objects and that every object can be represented as a product of its signs. Thus theorem 2 follows.

Theorem 2. *The set of operations $\{\neg, \wedge, \vee\}$ is complete.*

Corollary 1. *The sets of operations $\{\neg, \wedge\}$ and $\{\neg, \vee\}$ are complete.*

4 Algorithms for conversion of ways of representating information

One of the main ways of representating prior knowledge in supervised pattern recognition is using a rule for determining the class of the considered object. This rule often takes on the form of a function or several functions of algebra of logic, where each of these functions describes belonging of the object to a specific class.

Let all objects have equal number of signs and all signs have k values from the set $D = \{0, 1, ..., k-1\}$. All other cases can be easily reduced to this case.

Let the rule φ describe belonging of an object p to a class Y and be written as follows:

$$\varphi(d_1, d_2, ..., d_n) = \begin{cases} k-1, & p \in Y \\ 0, & p \notin Y \end{cases}$$

For conversion from the logical representation to the precedent-related one the following algorithms were developed:

- an algorithm based on algebra of objects (algorithm A_1);
- an algorithm based on algebra of logic (algorithm A_2).

General scheme of the algorithm A_1:

1. An object is introduced for each variable in φ.
2. Operations on these objects which correspond to operations of k-valued logic in φ are performed.
3. In case if not all signs are used in φ the collection obtained at stage 1 is complemented with missing signs.

General scheme of the algorithm A_2:

1. The rule φ is transformed to full DNF.
2. An object is introduced for each variable in full DNF obtained at stage 1.
3. Operations on these objects which correspond to operations of k-valued logic in full DNF are performed.

An algorithm called algorithm B was developed for conversion from the precedent-related representation to the logical one.

General scheme of the algorithm B:

1. A variable is introduced for each sign of the given collection.
2. For each object conjunction of variables corresponding to its signs is performed.

3. Disjunction of expressions obtained at stage 2 is considered as the required rule

Let $P = A_i(\varphi)$ denote that the collection P is an output of the algorithm A_i applied to the rule φ. Let also $\varphi = B(P)$ denote that the rule φ is an output of the algorithm B applied to the collection P.

Theorem 3. *For every rule φ and every object $p(s_1, s_2, ..., s_n) = (d_1, d_2, ..., d_n)$ two following statements are right:*

$$\varphi(d_1, d_2, ..., d_n) = k - 1 \Leftrightarrow (B \circ A_i(\varphi)) = k - 1 \quad (4)$$

$$p \in P \Leftrightarrow p \in (A_i \circ B(P)) \quad (5)$$

Thus the conversions made by algorithms A_i and B are mutually inverse.

Complexity of developed algorithms will be explored now. Consider collections P and Q containing r_1 and r_2 objects and having n_1 and n_2 signs correspondently.

Theorem 4. *The algorithm A_1 builds the collection $P \vee Q$ using N_1 operations. N_1 is defined as follows:*

$$N_1 = (k^{n_1} + k^{n_2})(r_1 + r_2) - r_1 r_2$$

Theorem 5. *The algorithm A_2 builds the collection $P \vee Q$ using N_2 operations, N_2 is defined as follows:*

$$N_2 = k^{n-n_1} r_1 + k^{n-n_2} r_2$$

Here n is the number of signs of the collection $P \vee Q$.

Theorem 6. *The complexity of algorithm B is $O(nr)$, r being the number of objects in the collection and n being the number of signs.*

Given φ in the form of DNF, the algorithm A_2 works faster than A_1. However, in general, if φ is an arbitrary function of k-valued logic, the algorithm A_1 is preferable because in that case complexity of stage 1 of the algorithm A_2 increases greatly.

Estimations of complexity of described algorithms were confirmed by computer experiments.

References

[1] *Classification (a problem).* http://dic.academic.ru/dic.nsf/ruwiki/969082.

[2] Gonzalez R.C., Tou J.T. (1978). *Pattern Recognition Principles.* Mir, Moscow.

[3] Rjabtsev A.V. (2002). Algebras for representation of prior knowledge in pattern recognition. *Digital Processing.* Vol. **6**, pp. 80-94.

[4] Yablonsky S.V. (1986). *Introduction to discrete mathematics.* Nauka, Moscow.