The information Io is commonly called training information, and the predicates P_j (S) are called elementary predicates.

In this formulation of the problem, it is actually required to construct some algorithm A($I_0(K_1$, . . . , $K_l)$, I(S)) = (α $^A_1(S)$...α $^A_l(S)$), *where* α $^A_1(S)$ =P$_j$ (S), j = 1, 2, . . . , l.

Obviously, the result of solving the problem Z is an algorithm with certain properties. In machine learning, the construction of such an algorithm is done within a scenario:

1)Some parametric family (model) of algorithms is selected;

2)The initial values of the parameters are fixed, and thus a specific algorithm is set;

3)The final setting of the algorithm to the subject domain is carried out during its training based on the training set data.

In this case, the learning process is reduced to the construction of algorithms (decision rules) that ensure the extremum of some criterion. Such a criterion, for example, can be the value of the average risk in a special class of decision rules. That is, at the beginning, the class of decision rules is defined up to parameters, and the training is reduced to finding the values of parameters that provide the extremum for the selected criterion.

Thus, in the most general form, the recognition problem can be written as follows: *The object description space is given, in which it is necessary to construct surfaces separating classes.*

In this formulation, the emphasis is on the construction of the algorithm *(on the construction of surfaces separating the classes)*, and therefore the problem has a pronounced algorithm-centric character.

In a more detailed analysis of the problem statement Z, we can propose an alternative variant of its formulation, when to find a solution the emphasis is shifted to the study of the property of classes. The new problem statement in this case is as follows.

Let two sets I_0, I(S) be given, i. e., admissible training information $I_0(K_1$, . . . , $K_l)$ and descriptions I(S) of admissible objects S $\in$ M, respectively.

It is required, based on the analyses of the of information $I_0(K_1$, . . . , $K_l)$, to find the set of distinguishing qualities of classes Q(K_1, . . . , K_l) such that $K_i \cap K_j = \emptyset$, $\forall$ i = j (where i, j = 1, 2, . . . , l) and using then the set Q(S) to compute the values of predicates Pj (S), j = 1, 2, . . . , l.

In this formulation, the solution of the problem is emphasized on the study of the property of classes and identification of features that provide class distinction. The recognition problem in this formulation is proposed to be called the Knowledge Discovery Classification Problem (KDC problem).

III. Method for Solving the KDC Problem

Let M be a set of objects, called admissible objects, and let it be covered by a finite number of subsets K_1, . . . , K_l : M =$\cup_{i=1}^{l}$ K_i called classes.The partition M is not completely defined. Let an a priori dictionary of features F = $\{f_1$, . . . , $f_n\}$ be given and on its basis only partial information X = $\cup_{i=1}^{l}$ X_i about classesK_1, . . . , K_l is given. Similarly, an admissible object S is defined on the basis of the features of the a priori dictionary.

The classification problem is to compute the values of predicates P_j(S)− S $\in K_j$, j = 1, 2, . . . , l, based on the partial information X about classes K1, . . . , Kl and the description of the admissible object S.

In the framework of the classical approach, the mathematical formulation of the classification problem is as follows: *Let X be the set of object descriptions and Y be the set of admissible classification answers. There is an unknown target dependency y*: X $\rightarrow$ Y, the values X^m = $\{(x_1,y_1)$, . . . ,$(x_m, y_m)\}$ of which are known only for the objects of the training set. It is necessary to construct an algorithm a: X $\rightarrow$ Y, which would approximate this target dependency not only on the objects of the finite set, but also on the whole set X.*

To solve the problem, first the model of algorithms is specified up to parameters, and then training is carried out by finding such values of parameters that provide the extremum of the selected criterion. The experience of practical use of this scenario has revealed a number of problematic points.

The choice of a model of algorithms A = a : X $\rightarrow$ Y is actually a non-trivial task. In this case, it is not so much about science as about the art of algorithm construction [21], [22]. Moreover, learning can be realized only in an automated mode. And the final algorithm a: X $\rightarrow$ Y is a "black box". It approximates an unknown target dependency, which cannot be interpreted in terms of the subject domain.

The above drawbacks are avoided by using an alternative approach, which is based on the idea of the compactness hypothesis that classes form compactly localized subsets in the object space [23]–[25].

The mathematical formulation of the *Knowledge Discovery Classification Problem in this case is as follows: Let X be the set of object descriptions and Y be the set of admissible answers for their classification. There is an unknown target dependency y* : X $\rightarrow$ Y, the values of which X^m = $\{(x_1, y_1)$, . . . ,$(x_m, y_m)\}$ are known only for the objects of the training set. It is required to find feature spaces in which classes do not intersect, and on their basis to construct an algorithm a : X $\rightarrow$ Y which would approximate this target dependency not only on the objects of the finite set, but also on the whole set X.*

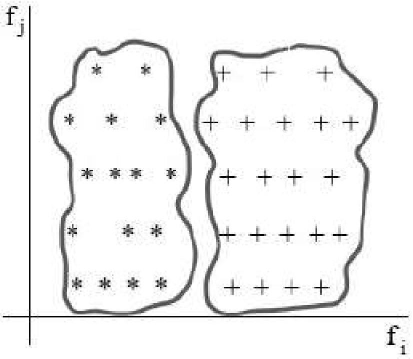
The KDC problem is solved in two steps. First, the feature spaces in which the class patterns do not intersect are searched. After that, the construction of the classification algorithm becomes a simple procedure.

The initial data of the Knowledge Discovery Classification Problem are the alphabet of classes K = $\{K_1, . . . , K_l\}$, a priori dictionary of features F = $\{f_1, . . . , f_n\}$, training set $X_m = \{(x_1, k_1), . . . (x_m, k_m)\}$, where k_i is the label of one of the classes of the alphabet K.

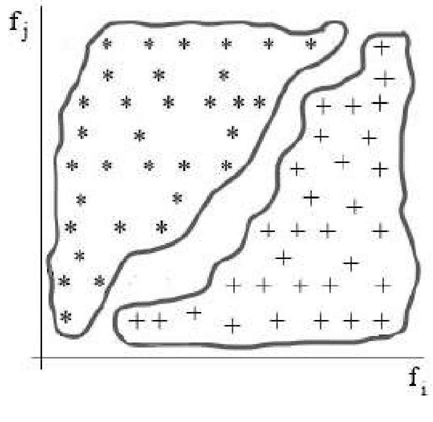
Let us denote by V = $\{v_1, . . . , v_q\}$ the set of all possible combinations of features from F. In total V contains q =$\sum_{i=1}^{n} C_n^i$ = 2n − 1 different subsets.

At first glance, the solution to the KDC problem should involve performing a brute-force search on the set V . However, using the properties of combinations of features the search can be significantly reduced.

Let us demonstrate by concrete examples what properties of class distinction can be possessed by features and combinations of features of the dictionary F. Suppose that two classes of objects * and + are given, and for some pair of attributes f_i andf_j the distribution of objects of these classes is as follows (Fig. 5).



'Figure 5.The first variant of mutual placement of objects.'



'Figure 6.The second variant of mutual placement of objects. '

Obviously, the feature fi in Figure 5 has the property of distinguishing two classes, while the feature fj has no such property. In addition, all combinations of features containing fi , have the property of distinguishing two classes, i. e. the patterns of classes in the corresponding feature space do not intersect.

Figure 6 shows that each of the features fi and fj individually does not have the property of distinguishing between the two classes, while the combination of features fi and fj has such a property.

If each of the features of the original a priori dictionary has the property of class distinction, then the solution of the KDC problem is reduced from the brute-force search to the consideration of n variants.

The algorithm of searching for combinations of features on the set V = {v1 . . . , vq} that provide class distinction, as a result of the following steps:

Step 1. Select a subset $V^+ = \{v_1^+ , . . . , v_i^+\}$ of V , where v_i^+ contains only one feature.

Step 2. For each v^+ i we construct class patterns (class definition areas) and compare their their mutual placement [26].

Step 3. If class patterns do not intersect, then feature v_i^+ is included in the set $V^* = \{v_1^* , . . . , v_k^*$ }.

Step 4. Exclude from the set V = {v1, . . . , vq} the subset $V^+ = \{v_1^+ , . . . , v_n^+$ n} and get V^Δ= {v_1^Δ , . . . , $v_p\Delta$ }.

Step 5. Exclude from V all combinations of v_i^Δ , that contain any combination from $V^* = \{v_1^* , . . . , v_k^*$ }.

Step 6. Take the next combination v_i^Δ from V^Δ and build a feature subspace based on it.

Step 7. In this feature subspace, we construct class patterns and compare their mutual placement.

Step 8. If the class patterns do not intersect, we include the combination of features v_i^Δ in the set V^* , and exclude from V^Δ all combinations that contain v_i^Δ.

Step 9. Repeat the process until V^Δ is empty.

The algorithm will result in the set $V = \{v_1^* , . . . , v_t^*\}$, where 0 t q. Based on the combinations $v_i \in V^*$, we formulate the previously hidden and empirically revealed regularities: in the feature space of a subset v_i^* classes do not intersect.

Note that within the framework of solving a particular applied problem, all combinations of features v_i can be interpreted in terms of the subject domain.

Combinations of features $v_i \in V$ define decision spaces in which class patterns do not intersect. In such spaces, the compactness hypothesis condition is satisfied and classification is performed by the rule:

- for each combination of features $v_i^* \in V^*$ (where i = 1, 2, . . . , t) and based on the training set data we build cluster structures $P_!^i$, . . . , P_l^i — patterns of classes K1, . . . , Kl [26];
- investigated object S ∈ K_m if S ∈ P_m^i ∀i = 1, 2, . . . , t.

To exemplify the demonstration of the generality principle in Figure 3, the following variant of the classification algorithm construction can be proposed:

There is an alphabet of classes K={Triangles, Ellipses, Rectangles} and an a priori dictionary of features F={area of figure, perimeter of figure, number of angles}. It is obvious that only the feature "number of angles" has the property of class distinction, because for all figures of class Triangles the value of the feature is equal to 3, for class Ellipses - equal to 0, for class Rectangles - equal to 4. Hence V = {number of angles} and the classification algorithm is as follows: IF (number of angles = 3) THEN Triangles ELSE IF (number of angles = 0) THEN Ellipses ELSE Rectangles

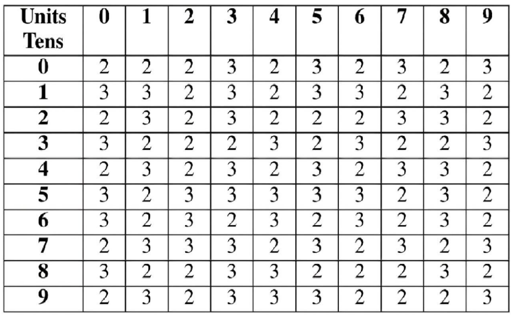
IV. Example of Solving the KDC Problem

Let's demonstrate the results of solving the KDC problem based on model data. Let's say we're given:

- classes **Not3** (there is no digit 3 in the number) and **Yes3** (there is at least one digit 3 in the number);
- a priori dictionary of features $F = \{Units, Tens\}$;
- training set, which consists of 250 two-digit integers, among which 200 have no digit 3 and 50 have at least one digit 3.

Table I shows the feature values of Units and Tens in the training set used in the numerical experiment.

Table I. Values of Units and Tens in the training set



Units Tens	0	1	2	3	4	5	6	7	8	9
0	2	2	2	3	2	3	2	3	2	3
1	3	3	2	3	2	3	3	2	3	2
2	2	3	2	3	2	2	2	3	3	2
3	3	2	2	2	3	2	3	2	2	3
4	2	3	2	3	2	3	2	3	3	2
5	3	2	3	3	3	3	3	2	3	2
6	3	2	3	2	3	2	3	2	3	2
7	2	3	3	3	2	3	2	3	2	3
8	3	2	2	3	3	2	2	2	3	2
9	2	3	2	3	3	3	2	2	2	3

Table II summarizes the results of class pattern intersection study based on the features Units, Tens, where *Not3i* is the number of *Not3i* class representatives for the i-th digit; *Yes3i* is the number of Yes3i class representatives for the i-th digit.

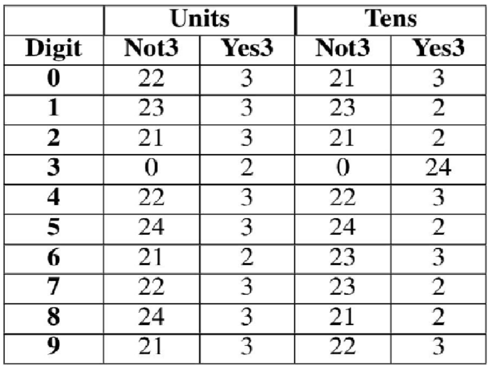
Table II shows that neither the Units feature nor the Tens feature provides an absolute separation between the **Not3** and **Yes3** classes.

Table III summarizes the results of the study on the intersection of class patterns based on the combination of features (Tens, Units).

Table III shows that:

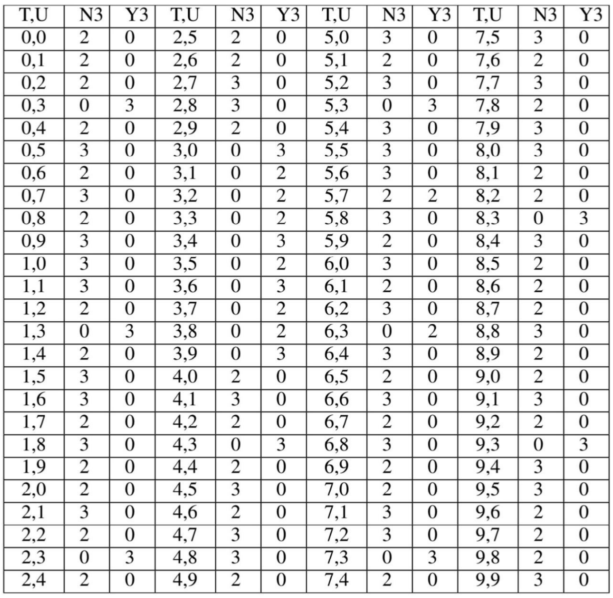
- all numbers of class *Not3* have no digit 3, and all numbers of class *Yes3* have at least one digit 3;
- combination of features *(Tens, Units)* provides absolute separation of *Not3* and *Yes3* classes;

Table II. Results for features Units and Tens



	Units		**Tens**	
Digit	**Not3**	**Yes3**	**Not3**	**Yes3**
0	22	3	21	3
1	23	3	23	2
2	21	3	21	2
3	0	2	0	24
4	22	3	22	3
5	24	3	24	2
6	21	2	23	3
7	22	3	23	2
8	24	3	21	2
9	21	3	22	3

Table III. Results for the combination of features (Tens, Units)



T,U	N3	Y3	T,U	N3	Y3	T,U	N3	Y3	T,U	N3	Y3
0,0	2	0	2,5	2	0	5,0	3	0	7,5	3	0
0,1	2	0	2,6	2	0	5,1	2	0	7,6	2	0
0,2	2	0	2,7	3	0	5,2	3	0	7,7	3	0
0,3	0	3	2,8	3	0	5,3	0	3	7,8	2	0
0,4	2	0	2,9	2	0	5,4	3	0	7,9	3	0
0,5	3	0	3,0	0	3	5,5	3	0	8,0	3	0
0,6	2	0	3,1	0	2	5,6	3	0	8,1	2	0
0,7	3	0	3,2	0	2	5,7	2	2	8,2	2	0
0,8	2	0	3,3	0	2	5,8	3	0	8,3	0	3
0,9	3	0	3,4	0	3	5,9	2	0	8,4	3	0
1,0	3	0	3,5	0	2	6,0	3	0	8,5	2	0
1,1	3	0	3,6	0	3	6,1	2	0	8,6	2	0
1,2	2	0	3,7	0	2	6,2	3	0	8,7	2	0
1,3	0	3	3,8	0	2	6,3	0	2	8,8	3	0
1,4	2	0	3,9	0	3	6,4	3	0	8,9	2	0
1,5	3	0	4,0	2	0	6,5	2	0	9,0	2	0
1,6	3	0	4,1	3	0	6,6	3	0	9,1	3	0
1,7	2	0	4,2	2	0	6,7	2	0	9,2	2	0
1,8	3	0	4,3	0	3	6,8	3	0	9,3	0	3
1,9	2	0	4,4	2	0	6,9	2	0	9,4	3	0
2,0	2	0	4,5	3	0	7,0	2	0	9,5	3	0
2,1	3	0	4,6	2	0	7,1	3	0	9,6	2	0
2,2	2	0	4,7	3	0	7,2	3	0	9,7	2	0
2,3	0	3	4,8	3	0	7,3	0	3	9,8	2	0
2,4	2	0	4,9	2	0	7,4	2	0	9,9	3	0

- the definition areas of *Not3* and emphYes3 classes do not intersect and are presented below (Fig. 7).

The classification algorithm is built on the basis of a rule: *IF (Units = 3 or Tens = 3) THEN Yes3 ELSE Not3.*

So, as a result of solving the KDC problem in automatic mode, the training data set were analyzed. The initially hidden regularity *the combination of features (Tens, Units) provides the distinction between classes Not3 and Yes3 by the rule IF (Units = 3 or Tens = 3) THEN Yes3 ELSE Not3* was found, and the classification algorithm was built on its basis.

V. Conclusion

The paper presents an original approach for solving the problem of learning from examples which is based on the use of the properies generality principle. The method of the principle implementation is proposed which provides