Mining Coherent Logical Regularities of Type 2 via Positional Preprocessing

Alexander Vinogradov[1] and Yury Laptin[2]

[1]Dorodnicyn Computing Centre of the Russian Academy of Sciences,
Vavilov str. 40, 119333 Moscow, Russian Federation
[2]Glushkov Institute of Cybernetics of the Ukrainian National Academy of Sciences,
Academician Glushkov pr. 40, 03680 Kiev, Ukraine

Abstract. A new approach to the development of efficient decision rules based on the use of two types of logical regularities is presented. In the approach positional representation of real values of features is used, and as a result, all conversions of types of logical regularities are performed using fast bit operations.

1 Introduction: Search of Logical Regularities in Data

The concept of logical regularity was proposed as convenient means for operating by compact clusters of data in training sample [4], [5]. This concept possesses a number of important advantages, we here will point to some of them, and also to one inevitable shortcoming. We consider standard formulation of the problem of recognition in feature space R^N in case of K classes. Let $X_k \subset R^N$ is one of the classes.
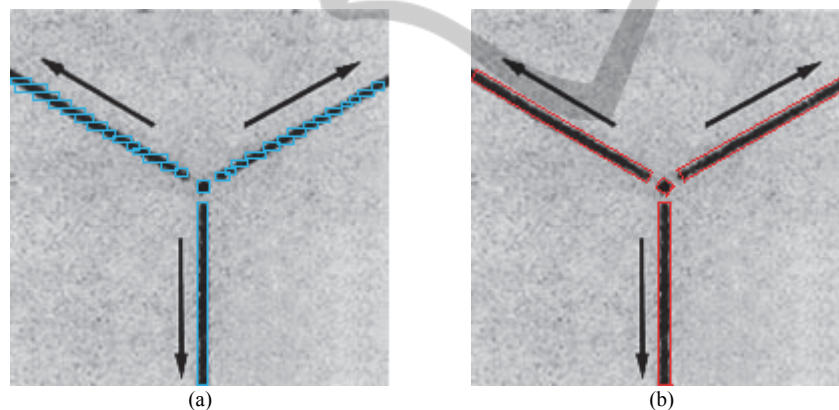
a. *Elementary Logical Regularity of Type 1* is a conjunction $LR = \underset{i}{\&} R_i$, containing not more than N predicates of the form $R_i = A_i < x_i < B_i$. Each conjunction $LR = \underset{i}{\&} R_i$ corresponds to a hyper-parallelepiped in R^N, and each predicate R_i describes the condition of entering the coordinate i of a new object x between i-edges of the hyper-parallelepiped. This reveals the main advantage of the approach: it's very easily to compute that the object x belongs to the class X_k described in terms of logical regularities of Type 1. It requires at most *2N* comparisons of real numbers per conjunction, and any other operation is unnecessary.

b. Logical conjunction (and corresponding hyper-parallelepipeds) can be combined. Disjunction of type $\bigcup_t LR_t$ provides more or less dense covering of objects of a class X_k. Since the values of A_i, B_i are fixed thresholds, only propositional logic is used in each case, and the form of the condition LR has a clear meaning to humans. Thus an appropriate disjunction $\bigcup_t LR_t$ not

only provides an efficient implementation of the decision rule, but may contain important new information that was not previously known, and that information is immediately presented in an easy to use form.

c. The greater the distance between the boundaries A_i, B_i, the greater volume is covered by hyper-parallelepiped, and the more important $\underset{i}{\&}R_i$ gets in the disjunction $\bigcup_t LR_t$. At the same time, the complexity of the decision rule in general decreases, and revealing elementary logical regularities of the maximum volume leads to exclusion some other ones from the decision rule. Significant help in this search lies in the very bit recording of real values of features, where higher digits correspond to the most distant from each other positions of boundaries A_i, B_i on the axis i. One of the motivations of this study was the fact that one can use this ready hierarchy of thresholds in search of precise boundaries for relevant hyper-parallelepipeds in R^N.

For all the merits of logical regularities of Type 1, it is easy to point out the main drawback of this approach: the form of disjunction $\bigcup_t LR_t$ and its constituent elementary logical regularities strongly depends on the direction of the main axes. If the geometry of the training data shows own characteristic directions that differ from the orientation of the feature axes, a description in terms of logical regularities of Type 1 may be very inefficient. Fig 1a. shows example of such data [7].

(a)   (b)

Fig. 1. a) Numerous elementary logical regularities of Type 1 are necessary for representation of a sample. Each of them covers just few of relevant data, but should be used in calculations; b) Only 4 elementary logical regularities of Type 2 provide exhaustive representation of the same sample.

To overcome this disadvantage *Elementary Logical Regularity of Type 2* are considered. For them, the boundaries of clusters in the training sample are described by set of hyper-planes of general form in R^N. Predicates in conjunction $\underset{i}{\&}R_i$ turn out as restrictions $R_j = (\langle x, n_j \rangle < C_j)$, where n_j is a normal to the hyper-plane j, C_j is bound-

ary value of the distance to the hyper-plane *j*. Fig.1b shows much simpler solutions for the data from Fig. 1a, where logical regularities of Type 2 are used.

2 Coherent Sets of Logical Regularities

Of course, the fewer predicates in $\bigcup_t LR_t$ use incremental directions n_j, the greater the advantages of logical regularities of Type 1 would be represented in calculations. In particular, the products $\langle x, n_j \rangle$ need to be performed only for additional directions n_j. The best case is implemented when for many predicates $R_j = (\langle x, n_j \rangle < C_j)$ normals n_j are the same, and check the conditions of membership of an object *x* to the appropriate clusters is reduced only to a single calculation $\langle x, n_j \rangle$, and then to test only inequalities as in the case of logical regularities of Type 1. Thus, an important task is to find a limited number of additional directions n_j that are most effectively used in the decision rule. Sets of logical regularities of Type 2, for which some directions n_j involved in their description are the same, we call *coherent on* n_j.

Under the new terms, our problem is formulated as finding the maximum coherent sets of logical regularities of Type 2 for the actual training sample *{Xk.}, k=1,...,K*. To compare possible directions n_j in R^N and to assess the suitability of sets of regularities for the role of coherent we will use fast bit preprocessing on the base of positional data representation. It should be said that the issues of optimal organization of such a search and choice of good initial approximation are very promising. They form a separate topic of research and discussion, and we leave it aside.

Reasons for choosing the term *coherent* are the following. Hierarchy of digits is a one-dimensional self-similar structure [1], [2], densely packed on the scale of real numbers. Positional representation transforms the simple binary package of this type in a similar dense packing of hyper-cubes in multidimensional space. However, on each axis *n, n=1,...,N*, representation is still binary, and weights of bits correspond to the basic row of higher harmonics up to the minimum length of the spatial wave on the grid *S*. From this point of view, all the boundary hyper-planes orthogonal to axis *n* are *coherent* (correspond to the same phase) with respect to highest harmonics on *n*, and the same is valid for any level of resolution on the grid *S*.

3 Competing Positional Representations of Coherent Sets

Let a linear transform $g \in GL(R^N)$ is made in R^N. Some of own to the training sample directions can be aligned by with the new directions of the coordinate axes. Then without loss of generality the same notation can be used further in the transformed space $R^N{}' = g(R^N)$.

Positional representation [3] of data in R^N is defined by bit grid $D^N \subset R^N$, where $|D| = 2^d$. For a grid point $x=(x_1, x_2, ..., x_N)$ the positional representation corresponds to effectively performed transformation on a slices in D^N, when the m-bit in binary representation of $x_n \in D$ becomes $p(n)$-bit of binary representation of m-digit in 2^N-ary number representing vector x as whole. Here $m \leq d$, and the function $p(n)$ defines a permutation on $\{1,2, .., N\}$. The result is a linearly ordered scale S of length 2^{dN}, representing one-to-one all the points of the grid in the form of the curve, densely filling the space D^N. For chosen grid D^N an exact solution of the problem of recognition with K classes results in K-valued function f, defined on the scale S. As known, m-digit in 2^N-ary positional representation corresponds to n-dimensional cube of volume $2^{N(m-1)}$. We call this cube *m-point*. It was shown in [6] that structure of the function f can be represented as a list $\bar{f}$, consisting of m-points, where different values m can be present.

Lemma 1. [6] When iterating points S the transition from one m-point to the other is for zeroing all younger bits of 2^N-ary positional representation of the current m-point.

Thus, the list $\bar{f}$ is formed by a single pass through S, and it is naturally stratified by the values of indices m and k. Already in this initial form the list $\bar{f}$ is a description of the sample $\{X_k.\}$, $k=1,...,K$ in terms of logical regularities of Type 1 with respect to the new coordinates. This description is not the best, but the structure of $\bar{f}$ contains the information necessary to improve it. In [6] presented a criterion to form of hyper-cubes hyper-parallelepipeds of increasing volume, that is, directly form the list $\bar{f}$-based data description of type $\bigcup_t LR_t$. It will go just about extensions, not about inclusions, since the very process of building the list $\bar{f}$ precludes trivial situations where one hyper-cube completely covers a smaller. The criterion is the following.

Lemma 2. [6] Two m-points C_1, C_2 are adjacent to n-edge iff: a) there is m'-cube C such that $m '> m$; b) record C_1 precedes C_2 in $\bar{f}$; c) in binary notation of 2^N-ary digits m, $m+1$, ..., $m'-1$ bits with the number n are recorded for C_1 (C_2) in the list $\bar{f}$ with values 1 (respectively, 0); d) all of the other bits in the records for the C_1, C_2 in the $\bar{f}$ coincide.

We describe some of the ways to use this criterion, which also formulate as lemmas. Our goal is to choose the simplest representations of the form $\bigcup_t LR_t$, and the main result here is the following.

Lemma 3. Correct extension of the boundaries A, B, C of any elementary logical regularity does not increase the complexity of the decision rule.

Proof follows immediately from the fact that after extension of any elementary logical regularity to the new boundaries exactly the same sequence of calculations is made. Moreover, some of conjunctions can be removed from the decision rule, if their domain of truth is entirely covered by the extension, and the complexity of the decision rule in this case is reduced. □

If the latter is implemented for at least one of elementary logical regularities, we say that the extension of the boundaries of the regularity is *efficient*. Let m-point C_1 and m'-point C_2 included in a set of logical regularities, both belong to the class k, $k = 1, ..., K$, are adjacent to each other along n-edge, and $m \leq m'$.

Lemma 4. If all $2^{m'-m}$ m-points adjacent to C_2 on n-edge belong to the class k, then there exist an efficient extension of any logical regularity that includes a hypercube C_2 as its frontier on the axis n and does not cover C_1.

Proof. In fact, in the hypothesis boundary of such regularity can always be extended with a layer of m-points adjacent to C_2. This extension is efficient because C_1 can now be excluded from the set of logical regularities. □

Lemma 5. For any logical regularity R, which contains C_1 as a boundary on the axis n, the efficient extension along the axis n is possible if and only if in the set of regularities there is some m''-point C_3, $m'' \leq m$, adjacent on n-edge with C_1 or C_2, for which the positional representation in 2^N-ary digits, older than m, is different from the representation for C_1 only in the n-th bits, and for the pair C_3, C_1 conditions of Lemma 4 are realized.

Proof. *Necessity* follows from the fact that at any extension of border there is m such that the layer of $2^{m'-m}$ m-points belonging to class k will be fully included in the volume of extended regularity. *Sufficiency.* Indeed, m''-point C_3 with these properties must be adjacent to n-edge with the point C_1, or with point C_2, but not on their common n-edge. The conditions imposed on the binary structure of positional representation of C_3 limit the possible location of C_3 with respect to C_1 and C_2. Point C_3 can be found only on the n-edge of some m-point C'_1, which differs from C_1 only in the value of coordinate n. If C_3 is adjacent to C_1, then the conditions of Lemma 4 are implemented for the pair C_3, C_1 obviously, where C_3, C_1 act as C_1 and C_2, respectively. Otherwise, the regularity R is made up of m-points of the form C'_1, and it passes through the hyper-cube C_2 at least until its opposite n-edge, as the point C_3 can not be completely covered by the hyper-cube C_2, in accordance to constructing the list $\bar{f}$ This means that the conditions of Lemma 4 are implemented on a pair of C_3, C'_1 for some m-point C'_1, which, like the C_1, is the boundary of the regularity R, but it limits R on opposite side with respect to the axis n. □

Let extensions of this type are made for logical regularities $L^q = \bigcup_t LR^q_t$, $q=1,2,..,Q$, which are built for a set of Q transforms $g^q \in GL(R^N)$ and for corresponding new directions of main axes in R^N. Advantage of the approach is the fact that the results of extensions can be easily combined: universal criterion is the total number of effectively employed elementary logical regularities of Type 1 in each L^q, whatever g^q is

used. The smaller the total number needed to cover the training set, the better the decision rule.

Of course, this criterion is too general, and on this basis it is difficult to formulate direct recommendations on choosing a suitable family of transforms $g^q \in GL(R^N)$, $q=1,2,..,Q$. However, the new approach presented here can combine the merits of the two types of logical regularities and essentially neutralize their weaknesses in the combined solutions. Thus, the solution presented in the example in Fig. 1b corresponds to $Q = 3$, one of the transforms is the identity, and for the new object x it's required to fulfill only two additional linear transforms. After that, all the calculations are reduced to a mere comparison of its coordinates with thresholds.

4 Conclusions

A new approach to the problem of using two types of logical regularities in pattern recognition tasks is presented, which reduces the impact of some of the known short-comings while maintaining the basic benefits. Positional representation of digital data is used that allows implementing fast bit operations in the process of converting types of logical regularities. The approach can be used in various areas of data analysis in pattern recognition and data mining. Mathematical techniques open up the possibility of using the approach in software for specialized processors, based on the implementation of bit transforms, particularly in the processing of the images and video data.

Acknowledgements

References

1. Mark de Berg, Marc van Kreveld, Mark Overmars, and Otfried Schwarzkopf: Computational Geometry (2nd revised ed.). Springer-Verlag (2000) 506 pages.
2. T. Lindeberg: Scale-Space Theory in Computer Vision. Kluwer Academic Publishers (1994) 440 pages.
3. Aleksandrov V.V., Gorskiy N.D.: Algoritmy i programmy strukturnogo metoda obrabotki dannykh. L. Nauka (1983) 208 str.(Russian)
4. Yu.I.Zhuravlev, V.V.Ryazanov, O.V.Senko: RASPOZNAVANIE. Matematicheskie metody. Programmnaya sistema. Prakticheskie primeneniya. Izdatelstvo "FAZIS", Moscow (2006)168 str. (Russian)
5. Ryazanov V.V.: Logicheskie zakonomernosti v zadachakh raspoznavaniya (parametricheskiy podkhod). Zhurnal vychislitelnoy matematiki i matematicheskoy fiziki, T. 47/10 (2007) str.1793-1809.(Russian)

62

6. Vinogradov A., Laptin Yu.: Usage of Positional Representation in Tasks of Revealing Logical Regularities. Rroceedings of the International Conference VISIGRAPP 2010, Workshop IMTA-3, pp.100-104.

7. Shur, V. et al.: Poverhnostnye samopodobnye nanodomennye struktury, inducirovannye lazernym izlucheniem v niobate litiya. Fizika tverdogo tela, tom50, vyp.4 (2008) str.689-695. (Russian)