I T H E A

International Journal INFORMATION THEORIES & APPLICATIONS

2006 Volume 13 Number 3

International Journal
INFORMATION THEORIES & APPLICATIONS
ISSN 1310-0513
Volume 13 / 2006, Number 3

IJ ITA is official publisher of the scientific papers of the members of the Association of Developers and Users of Intellectualized Systems (ADUIS).

IJ ITA welcomes scientific papers connected with any information theory or its application. Original and non-standard ideas will be published with preferences.

IJ ITA rules for preparing the manuscripts are compulsory.
The rules for the papers for IJ ITA as well as the subscription fees are given on www.foibg.com/ijita.
The camera-ready copy of the paper should be received by e-mail: foi@nlcv.net

Responsibility for papers published in IJ ITA belongs to authors.

General Sponsor of IJ ITA is the Consortium FOI Bulgaria (www.foibg.com).

International Journal "INFORMATION THEORIES & APPLICATIONS" Vol.13, Number 3, 2006
Printed in Bulgaria
Edited by the Institute of Information Theories and Applications FOI ITHEA, Bulgaria, in collaboration with the V.M.Glushkov Institute of Cybernetics of NAS, Ukraine, and the Institute of Mathematics and Informatics, BAS, Bulgaria.
Publisher: FOI-COMMERCE - Sofia, 1000, P.O.B. 775, Bulgaria. www.foibg.com, e-mail: foi@nlcv.net

ISSN 1310-0513

RANDOMIZED PARALLELIZATION – A NEW METHOD FOR SOLVING HARD COMBINATORIAL PROBLEMS

Arkadij Zakrevskij

Abstract: A new method for solving some hard combinatorial optimization problems is suggested, admitting a certain reformulation. Considering such a problem, several different similar problems are prepared which have the same set of solutions. They are solved on computer in parallel until one of them will be solved, and that solution is accepted. Notwithstanding the evident overhead, the whole run-time could be significantly reduced due to dispersion of velocities of combinatorial search in regarded cases. The efficiency of this approach is investigated on the concrete problem of finding short solutions of non-deterministic system of linear logical equations.

Keywords: combinatorial problems, combinatorial search, parallel computations, randomization, run-time, acceleration.

ACM Classification Keywords: G.2.1 Combinatorics – combinatorial problems, combinatorial search, G.3 Probability and Statistics – randomization, G.4 Mathematical software – efficiency, parallel and vector implementations.

Introduction

There exist some hard combinatorial optimization problems that could be solved on computer by many different ways, and it is practically impossible to say in advance which of these ways will lead to the solution quicker. Moreover, the run-times for these ways may differ very much, in tens, hundreds, thousands and more times. Unfortunately, we do not know beforehand, which of them is better for the concrete input data.

In such a case, it would be more profitable to use several competing ways simultaneously and organize parallel computations looking for the solution, terminating them as soon as one of the competitors will find the solution.

In other words, a problem under solution could be changed for a selection of other problems solved in parallel. These new problems could seem quite different, but nevertheless they should be equivalent to the initial problem, that means they will have the same set of solutions. The overhead expenses for constructing these sets of new problems and dealing with them can be compensated with interest by the essential acceleration of the search for solution of the considered problem.

That idea is demonstrated below by the problem of finding a short solution of an undefined system of linear logical equations.

Solving Linear Logical Equations

Let us regard a system of *m linear* Boolean equations with *n variables*

$$\begin{aligned} &a_1^1x_1 \oplus a_1^2x_2 \oplus \ldots \oplus a_1^nx_n = y_1 ,\\ &a_2^1x_1 \oplus a_2^2x_2 \oplus \ldots \oplus a_2^nx_n = y_2 ,\\ &\qquad\qquad \ldots \\ &a_m^1x_1 \oplus a_m^2x_2 \oplus \ldots \oplus a_m^nx_n = y_m , \end{aligned} \qquad (1)$$

where a_i^j are the Boolean coefficients of the system, x_j – unknown variables (a_i^j, $x_j \in \{0, 1\}$, $i = 1, 2, \ldots, m$, $j = 1, 2, \ldots, n$), $\oplus$ is the EXOR operator.

This system can be represented in the compact matrix form:

$$A\,x = y , \qquad (2)$$

where A is a Boolean $m \times n$ coefficient matrix, $x = (x_1, x_2, \ldots, x_n)$ and $y = (y_1, y_2, \ldots, y_m)$ – Boolean vectors. AND operator is used as an internal one, and EXOR as the external one for the matrix multiplication.

$$\bigoplus_{j=1}^{n} a_i^j x_j = y_i . \qquad (3)$$

Solutions (roots) of this system are the sets of variable values, for which all equations will be satisfied. In this case EXOR sum of the columns of matrix A, corresponding to the component-wise sum by modulo 2 of the columns of A, which correspond to the variables having value 1, should be equal to the column y. The system can be deterministic (defined, having the single solution), non-deterministic (undefined, several solutions exist) or contradictory (over-defined), where there are no solutions [1, 2]. Let us assume below that $n > m$ and all rows of A are linearly independent, so the system is non-deterministic. Then the number of possible solutions is equal to 2^{n-m}.

A great property of the linear equations is that they can be solved in regard of any variable. An effective Gauss method of variable exclusion is based on this property [3] and allows finding the only solution for a deterministic system or the set of all possible solutions for a non-deterministic one. But it is not sufficient when some optimal solution should be selected from that set which could be very large.

For instance, solving a non-deterministic system of linear logical equations $A\,x = y$, it is necessary sometimes to find a solution with minimum number of unknown variables $x_1, x_2, \ldots, x_n$ taking value 1 – such a solution is called *shortest*. That is important in many practical cases, for instance, when logic circuits in AND/EXOR basic are synthesized [4-6] or when some problems of information security are considered.

Looking for Short Solutions of Systems of Linear Logical Equations

Evidently, a shortest solution could be found by means of selecting from matrix A one by one all different combinations of columns, consisting first of 1 column, then of 2 columns, etc. and examining each of them to see if their sum equals vector y. As soon as it happens, the current combination is accepted as the sought-for solution. That moment could be forecasted. If the weight of the shortest solution (the number of 1s in vector x) is w, the number N of checked combinations is defined approximately by the following formula, where C_n^i is the number of i-element subsets taken from a set of n elements:

$$N = \sum_{I=0}^{w} C_n^i . \qquad (4)$$

It could be very large, as is demonstrated below.

It was shown [7], that the expected weight γ of the shortest solution of an SLLE with parameters m and n can be estimated before finding the solution itself. We represent this weight as a function $\gamma\,(m, n)$. First we find the mathematical expectation $\alpha\,(m, n, k)$ of the number of solutions with weight k. We assume that the considered system was randomly generated, which means that each element of A takes value 1 with the probability 0.5 and any two elements are independent of each other. Then the probability that a randomly selected column subset in matrix A is a solution equals 2^{-m} (the probability that two randomly generated Boolean vectors of size m are equal). Since the number of all such subsets having k elements equals $C_n^k = \frac{n!}{(n-k)!\,k!}$, we get:

$$\alpha\,(m, n, k) = C_n^k\, 2^{-m}. \qquad (5)$$

Similarly, we denote as $\beta\,(m, n, k)$ the expected number of the solutions with weight not greater than k:

$$\beta\,(m, n, k) = \sum_{i=0}^{k} C_n^i\, 2^{-m}. \qquad (6)$$

Now, the expected weight γ of the shortest solution can be estimated well enough by the maximal value of k, for which $\beta\,(m, n, k) < 1$:

$$\gamma\,(m, n) = k, \quad \text{where } \beta\,(m, n, k) < 1 \le \beta\,(m, n, k+1). \qquad (7)$$

For example, for a system of 40 equations with 70 variables the expected weight γ equals 10, and reaches 31 when $m = 100$ and $n = 130$.

In the last case, the described above simple algorithm should check about 10^{30} combinations, according to formula (4) in which $w = \gamma$. Examining them with the speed of one million combinations per second we need about 30 000 000 000 000 000 years to find the solution. Practically impossible!

A great acceleration can be achieved by using Gaussian method of variables exclusion [3] developed for solving systems of linear equations with real variables and adjusted for Boolean variables [5]. It enables to avoid checking all subsets of columns from A which have up to w columns, when only one of 2^{n-m} regarded combinations presents some roots of the system.

Its main idea consists in transforming the extended matrix of the system (matrix A with the added column y) to the *canonical form*. A maximal subset of m linear independent columns (does not matter which one) is selected from A and by means of equivalent matrix transformations (adding one row to another) is transformed to I-matrix, with 1s only on the main diagonal. That part is called a basic, the rest $n - m$ columns of the transformed matrix A constitute a *remainder*. The column y is changed by that, too.

According to this method the subsets of the remainder are regarded, i.e. combinations selected from the set which has only $n - m$ columns (not all n columns!). It is easy to show that every of these combinations enables to get a solution of the considered system (any sum of its elements can be supplemented with some columns from matrix I to make it equal to y). When we are looking for a shortest solution using this method, we have to consider different subsets of columns from the remainder, find the solution for each such subset and select a subset, which leads to the shortest solution. If it is known that the weight of the shortest solution is not greater than w, then the level of search (the cardinality of inspected subsets) is restricted by w. Note that if $w \geq n - m$, then all 2^{n-m} subsets must be searched through.

Now, for the same example ($m = 100$ and $n = 130$), $N \cong 10^9$, which means that the run-time of Gaussian method is about only 17 minutes.

Unfortunately (for practical application), it rises very quickly when increasing parameters m and n. For example, when $m = 625$ and $n = 700$, the number of checked combinations reaches 2^{75} and the run-time surpasses one milliard years.

It could be significantly reduced when we know that there exists a short solution with weight w less than γ. For example, the following situation could be imagined. A random matrix A is generated, then w columns are selected by random, and their component-wise modulo 2 sum is defined as vector y.

In that case the search (checking different solutions one by one in order to find a shortest one) could be interrupted as soon as the weight v of the current solution satisfies the inequality $v < \gamma$, and we may conclude that the sought-for solution is found. This idea lies in the base of the recognition method [8, 9].

The Level of Search and its Dispersion

Suppose, a short solution exists presented by w columns of matrix A (their sum equals vector y). Let p of them belong to the remainder. Then this solution will be found when such subsets of columns from the remainder are checked which have exactly p elements. In other words, the solution will be found on the *level of search* L when this one equals p.

That quantity depends, first, on the regarded example (how matrix A was generated) and, second, on the way of getting the remainder - how m linearly independent columns were selected from initial matrix A, splitting that matrix into two parts, basic and remainder.

Chances to discover a solution with weight w on the low search level $L = k$ can be estimated as follows. Let us regard an m-component Boolean vector a having n-component subvector a^*. There exist C_m^w different values of vector a with w ones. Suppose all of them are equiprobable. The number of those which have exactly k ones in subvector a^* (by that $k \leq n$) is estimated by formula $C_n^k C_{m-n}^{w-k}$. So, this number comes to $100\, C_n^k C_{m-n}^{w-k} / C_m^w$ percent of the number of values of vector a with w ones.

An experiment was conducted, where 10 different Boolean matrices A were generated at random with parameters n 500 and $m = 430$. In each of them 30 different subsets of linearly independent columns were selected and corresponding canonical forms were constructed, which produced 300 variants of the remainder. A set of 75 columns was chosen from A by random constituting a short solution - their sum was accepted as vector y.

All intersections of that set with remainders were found. The cardinalities of these intersections (numbers of columns in them) define the level of search L on which the considered short solution will be found. The values of L for all 300 variants are shown in Table 1. Its columns correspond to different examples (i) of system and the rows correspond to different remainders in them (j) selected at random.

Table 1. Dispersion of level L in space of examples and remainders

$j \setminus i$	1	2	3	4	5	6	7	8	9	10
1	10	13	12	12	16	11	13	10	12	13
2	8	9	13	9	9	10	11	10	7	9
3	10	10	11	9	11	13	11	10	12	14
4	10	8	11	11	10	8	12	10	13	11
5	10	11	14	6	11	8	17	15	10	6
6	12	6	7	10	12	11	11	12	8	16
7	9	11	8	9	10	12	8	9	8	17
8	8	15	6	13	14	16	15	12	12	12
9	10	13	9	5	14	11	13	6	16	10
10	14	15	9	8	11	10	8	6	13	12
11	14	12	10	12	6	10	12	15	13	9
12	8	9	12	16	16	7	18	11	9	12
13	9	11	6	13	15	9	9	10	10	9
14	14	12	9	14	6	9	7	13	11	13
15	9	5	13	12	11	10	7	13	9	16
16	13	16	6	15	10	7	4	10	11	13
17	6	10	15	13	11	11	13	11	9	12
18	9	9	10	11	10	6	14	8	10	12
19	5	15	7	11	10	16	9	12	9	13
20	11	11	7	8	7	7	11	9	13	10
21	8	12	10	8	8	13	9	6	12	8
22	15	8	11	8	16	6	12	11	15	5
23	10	12	7	14	11	12	13	13	9	9
24	10	9	10	12	4	10	12	11	10	10
25	9	11	10	10	11	11	14	9	7	19
26	11	14	8	10	11	5	14	8	12	10
27	11	12	14	9	10	7	14	11	11	8
28	18	12	13	12	6	8	8	10	12	11
29	12	12	7	9	14	12	7	8	9	12
30	10	7	6	11	13	8	11	15	11	18

As one can see from the table, the level of search L is subjected to a great dispersion, varying from 4 up to 19. But this dispersion is surpassed in many times by the dispersion of the corresponding run-times, as is shown below.

The values of run-time for the same examples were obtained, using PC COMPAC Presario, 1000 MH, and presented in Table 2, being measured in seconds (s), minutes (m), hours (h), days (d) and years (y). The distribution of levels is shown on the right side of the table, N indicating the number of entries with value L in Table 1.

Note that fulfilled experiments were virtual ones: the run-time was not measured directly but was calculated by the method suggested in [10] and ensuring a rather good approximation.

Table 2. Dependence of run-time *T* on level *L*

L	*T*	*N*	
4	1.6s	2	11
5	22s	5	11111
6	4m	16	1111111111111111
7	38m	16	1111111111111111
8	5h	27	111111111111111111111111111
9	1.5d	36	111111111111111111111111111111111111
10	10d	44	11111111111111111111111111111111111111111111
11	55d	44	11111111111111111111111111111111111111111111
12	280d	39	111111111111111111111111111111111111111
13	3.6y	27	111111111111111111111111111
14	15y	16	1111111111111111
15	59y	12	111111111111
16	214y	10	1111111111
17	713y	2	11
18	2213y	3	111
19	6394y	1	1

Method of Randomized Parallelization

It logically follows from the tables 1 and 2 that efficient algorithms for finding short solutions may be constructed which use the presented great dispersion of values *L* and *T* on the sets of examples and remainders.

These are algorithms of randomized parallelization, which instead of one canonical form of the system (*A*, *y*) use *q* canonical forms of the same system with different basics selected at random (and remainders defined by them)..Using these forms one by one, such an algorithm consecutively increases the cardinality of checked combinations of columns. That means that it looks for the short solution at first on level $L = 0$, then on level $L = 1$, then on level $L = 2$, etc., until a solution with weight w will be found in some form, which satisfies inequality $w < \gamma$. In that case the algorithm terminates.

Experiments

Some results of the program virtual implementation of that algorithm for different values of *q* are shown in Table 3. Thirty examples of random systems with parameters $n = 1000$, $m = 900$ and the weight of the short solution $w = 100$ have been generated and solved. The following denotation is used in the table:

No is the number of the regarded example,

N is the number of form where the short solution was found,

L is the level at which it was found, and

T is the time spent for it.

Note that value 1 of parameter *q* corresponds to the pure Gaussian method dealing with only one canonical form of the system. Changing it for pseudo-parallel algorithm on the base of 300 forms we accelerate finding the short solution on an average in 46 million times.

The immense acceleration!

Table 3. Results of solving non-deterministic systems of linear logical equations with parameters $n = 1000$, $m = 900$, $w = 100$.

	$q = 1$			$q = 10$			$q = 30$			$q = 300$		
No	*N*	*L*	*T*	*N*	*L*	*T*	*N*	*L*	*T*	*N*	*L*	*T*
1	1	8	7d	9	6	16h	9	6	18h	33	4	19m
2	1	9	69d	8	6	14h	16	5	2h	254	2	4m
3	1	10	2y	7	7	7d	28	6	2d	234	2	4m
4	1	13	776y	3	6	5h	3	6	8h	117	3	5m
5	1	3	4s	1	3	4s	1	3	5s	1	3	4m
6	1	10	2y	5	7	5d	26	5	3h	56	2	4m
7	1	12	112y	9	4	3m	9	4	3m	57	3	5m
8	1	8	7d	10	5	1h	10	5	1h	106	3	5m
9	1	12	112y	10	5	1h	28	4	10m	141	3	6m
10	1	9	69d	2	6	4h	2	6	6h	95	2	4m
11	1	13	776y	7	8	82d	15	4	5m	50	2	4m
12	1	13	776y	9	8	104d	14	5	2h	117	3	5m
13	1	10	2y	8	6	14h	8	6	16h	35	3	4m
14	1	6	58m	1	6	2h	1	6	4h	39	3	4m
15	1	6	58m	1	6	2h	11	5	1h	134	3	6m
16	1	14	4954y	2	8	27d	28	4	10m	205	3	7m
17	1	6	58m	1	6	2h	14	5	2h	49	3	4m
18	1	10	2y	2	7	2d	27	5	3h	285	2	4m
19	1	13	776y	8	7	8d	8	7	9d	84	3	5m
20	1	10	2y	2	6	4h	2	6	6h	203	4	1,3h
21	1	8	7d	7	5	45m	7	5	52m	93	4	39m
22	1	7	13h	10	6	17h	27	4	9m	190	3	6m
23	1	12	112y	2	5	13m	16	2	4s	16	2	4m
24	1	15	29185y	4	7	4d	24	6	2d	226	2	4m
25	1	10	2y	2	6	4h	2	6	6h	46	3	4m
26	1	13	776y	9	7	9d	16	5	2h	112	4	45m
27	1	10	2y	6	8	71d	16	4	6m	281	3	8m
28	1	12	112y	7	8	82d	18	6	1d	87	3	5m
29	1	12	112y	6	4	2m	6	4	2m	254	2	4m
30	1	12	112y	7	8	82d	22	6	2d	31	4	18m
The sum:			38704y			1,3y			20d			5,3h

Conclusion

A new approach to solving hard combinatorial optimization problems is suggested, demonstrated on the problem of finding a short solution of a non-deterministic system of linear logical equations. Its idea is in changing the regarded problem for a set of other similar problems equivalent to the given one and solving them in parallel. The run-time could be considerably reduced by that, possibly in many millions times, as some results of computer experiments show.

Acknowledgements

The work had been performed thanks to the partly support of ISTC (Project B-986) and the Fond of Fundamental Researches of Belarus.

Bibliography

1. Kostrikin A., Manin Y. Linear algebra and geometry. Translated from the second Russian (1986) edition by M. E. Alferieff. Revised reprint of the 1989 English edition. Gordon and Breach, 1997.
2. Lankaster P. Theory of matrices, Academic Press, New York – London, 1969.
3. Gauss C.F.Beitrage zur Theorie der algebraischen Gleichungen, Gött, 1849.
4. Sasao T. Representations of logic functions using EXOR operators. – In "Representations of discrete functions" (ed. by T. Sasao and M. Fujita) – Kluwer Academic Publishers, Boston/London/Dordrecht, 1996, pp. 29-54.
5. Zakrevskij A.D. Looking for shortest solutions of systems of linear logical equations: theory and applications in logic design. – 2.Workshop "Boolesche Probleme", 19./20. September 1996, Freiberg/Sachsen, pp. 63-69.
6. Zakrevskij A.D., Toropov N.R. Polynomial representation of partial Boolean functions and systems. – Minsk, Institute of technical cybernetics, Belarusian NAS, 2001; Moscow: URSS (Second edition), 2003 (In Russian).
7. Zakrevskij A.D., Vasilkova I.V. Fast algorithm to find the shortest solution of the system of linear logical equations. - Problems of logical design. – Minsk: UIIP of NAS of Belarus, 2002, pp. 5-12. (In Russian).
8. Zakrevskij A.D. Randomization of a parallel algorithm for solving undefined systems of linear logical equations. – Proceedings of the International Workshop on Discrete-Event System Design – DESDes'04. – University of Zielona Gora Press, Poland, 2004, pp. 97-102.
9. Zakrevskij A.D. Efficient methods for finding shortest solutions of systems of linear logical equations. – Control Sciences, 2003, No 4, pp. 16-22. (In Russian).
10. Zakrevskij A.D., Vasilkova I.V. Forecasting the run-time of combinatorial algorithms implementation. – Methods of logical design, issue 2. Minsk: UIIP of NAS of Belarus, 2003, pp. 26-32. (In Russian).

Author's Information

Arkadij Zakrevskij – United Institute of Informatics Problems of the NAS of Belarus, Surganov Str. 6, 220012 Minsk, Belarus; e-mail: zakr@newman.bas-net.by

THE BOUNDARY DESCRIPTORS OF THE *n*-DIMENSIONAL UNIT CUBE SUBSET PARTITIONING[1]

Hasmik Sahakyan, Levon Aslanyan

Abstract: The specific class of all monotone Boolean functions with characteristic vectors of partitioning of sets of all true-vertices to be minimal is investigated. These characteristic vectors correspond to the column-sum vectors of special (0,1)-matrices – constructed by the interval bisection method.

Keywords: monotone Boolean functions, (0,1)-matrices.

ACM Classification Keywords: G.2.1 Discrete mathematics: Combinatorics

1. Introduction

The problem of general quantitative description of vertex subsets of n dimensional unit cube E^n, through their partitions, the existence problem and composing algorithms for vertex subsets by the given quantitative characteristics of partitions are considered. Each of these sub-problems has its own theoretical and practical significance. The existence and composing problems are studied intensively [BI, 1988; C, 1986; S, 1986], but the

[1] The research is supported partly by INTAS: 04-77-7173 project, http://www.intas.be

complexity issues are still open [B 1989]. Therefore, studying the problem in different restrictions, for particular type of subsets, and obtaining different necessary and sufficient conditions is essential.

Concerning the problem of general quantitative description - the complete description of the set of all integer-valued vectors - quantitative descriptors of subset partitions is obtained [S 1997]. The description is based on structures, where the characteristic vectors corresponding to the monotone Boolean functions of E^n have an important role. The main result of this research concerns with the partitioning-boundary-cases, which correspond to the special monotone Boolean functions having minimal characteristic partitioning vectors, which, in their turn, correspond to the column sum vectors of (0,1)-matrices constructed by the interval bisection method.

2. Structure

Let $M \subseteq E^n$ be a vertex subset of fixed size $|M| = m$, $0 \le m \le 2^n$. An integer, nonnegative vector $S = (s_1, s_2, \cdots, s_n)$ is called the characteristic vector of partitions of set M, if its coordinates equal to the partition-subsets sizes of M by coordinates $x_1, x_2, \cdots, x_n$ - the Boolean variables composing E^n. s_i equals the size of one of the partition-subsets of M by the i-th direction and then $m - s_i$ is the size of the complementary part of partition. To make this notation precise we will later assume that s_i is the size of the partition subset with $x_i = 1$.

The complete description of all integer-coordinate vectors, which serve as characteristic vectors of partitions for vertex subsets of size m sets is based on structures, where characteristic vectors corresponding to the monotone Boolean functions play an important role.

Let Ξ^n_{m+1} denotes the set of all vertices of n dimensional, $m+1$ valued discrete cube, i.e. the set of all integer-vectors $S = (s_1, s_2, \cdots, s_n)$ with $0 \le s_i \le m$, $i = 1, \cdots, n$. ψ_m denotes the set of all characteristic vectors of partitions of m-subsets of E^n. It is evident, that - $\psi_m \subseteq \Xi^n_{m+1}$. Below, the description of ψ_m has been given in terms of Ξ^n_{m+1} geometry: the vertices are distributed schematically on the $m \cdot n + 1$ layers of Ξ^n_{m+1} according to their weights – sums of all coordinates. The l-th layer contains all vectors $S = (s_1, s_2, \cdots, s_n)$ with $l = \sum_{i=1}^{n} s_i$.

Let $\hat{\psi}_m$ and $\check{\psi}_m$ are subsets of ψ_m, consisting of all its upper and lower boundary vectors, correspondingly:

$\hat{\psi}_m$ is the set of all "upper" vectors $S \in \psi_m$, for which $R \notin \psi_m$ for all $R \in \Xi^n_{m+1}$, greater than S. $\check{\psi}_m$ is the set of all "lower" vectors $S \in \psi_m$, for which $R \notin \psi_m$ for all vectors R from Ξ^n_{m+1}, less than S.

These sets of all "upper" and "lower" boundary vectors have symmetric structures - for each upper vector there exists a corresponding (opposite) lower vector, and vice versa; so that also the numbers of these vectors are equal:

$$\hat{\psi}_m = \{\hat{S}_1, \cdots, \hat{S}_r\} \text{ and } \check{\psi}_m = \{\check{S}_1, \cdots, \check{S}_r\}.$$

Let $\hat{S}_j$ and $\check{S}_j$ be an arbitrary pair of opposite vectors from $\hat{\psi}_m$ and $\check{\psi}_m$ correspondingly. $I(\hat{S}_j)$ (equivalently $I(\check{S}_j)$) will denote the minimal sub-cube of Ξ^n_{m+1}, passing through this pair of vectors. Then,

$$I(\hat{S}_j) = \{Q \in \Xi^n_{m+1} / \hat{S}_j \le Q \le \check{S}_j\} \text{ (the coordinate-wise comparison is used).}$$

The following Theorem states that the minimal sub-cubes passing the pairs of corresponding opposite vectors of the boundary subsets are continuously and exactly filling the vector area ψ_m.

Theorem 1 [S 1997]: $\psi_m = \bigcup_{j=1}^{r} I(\hat{S}_j)$.

3. Boundary Cases

Boundary vectors of ψ_m can be described by the monotone Boolean functions, defined on E^n: the set of all characteristic boundary vectors is a subset of the set of all characteristic vectors of partitions of monotone Boolean functions. This fact is confirmed by the following theorem.

Theorem 2 [S 1997]: $\bar{\psi}_m \subseteq M_m^1$ and $\breve{\psi}_m \subseteq M_m^0$,

where M_m^1 and M_m^0 are the sets of characteristic vectors of those m-subsets of E^n, which correspond to the sets of all one-valued vertices and all zero-valued vertices defined by monotone Boolean functions correspondingly.

Let $\wp_{min}(m,n)$ is the class of monotone Boolean functions for which the partitioning characteristic vectors of $\bar{\psi}_m$ are placed on the minimal possible layer of Ξ_{m+1}^n, - denote this layer by L_{min}. A similar class of functions is related to $\breve{\psi}_m$.

The structure of functions of class $\wp_{min}(m,n)$ is related to the linear ordering of vertices of E^n by decreasing of numeric values of the vectors expressed as $x_n, x_{n-1}, \cdots, x_1$. Call this sequence of vertices "decreasing", denoting it by D_n. D_n has some useful properties. The first 2^{n-1} vertices of D_n compose an $n-1$ dimensional sub-cube (interval), incident to the vertex $x_i = 1, i = \overline{1,n}$. The reminder 2^{n-1} vertices are in the "shadow" of that interval by the variable x_n. The vertex order in both sub-cubes correspond to the orderings in D_{n-1}. In general, for an arbitrary k the first 2^k vertices of the sequences compose k dimensional intervals, where variables $x_n, x_{n-1}, \cdots, x_{k+1}$ accept fixed values; and then the next 2^k vertices are in the "shadow" of that interval by the direction x_{k+1}, and continuing the same way we will receive the recursive structure of the series D_n.

The given construction provides that the initial segments by m of the "decreasing" sequence D_n correspond to the sets of all one-valued-vectors of some monotone Boolean function. We denote this functions by $\mu(m,n)$.

Next theorem confirms that the constructed functions are the required monotone Boolean functions.

Theorem 3: $\mu(m,n) \in \wp_{min}(m,n)$.

The general technique to prove this Theorem consists of partitioning of E^n by one or two directions, the inductions by these parameters and the partitioning of all possible cases into the several sub cases.

The minimal layer L_{min} might be presented by the following formula:

$$L_{min} = \sum_{i=1}^{n} s_i = \sum_{i=1}^{p} \left((n - k_i - (i-1)) \cdot 2^{k_i} + k_i \cdot 2^{k_i - 1} \right),$$

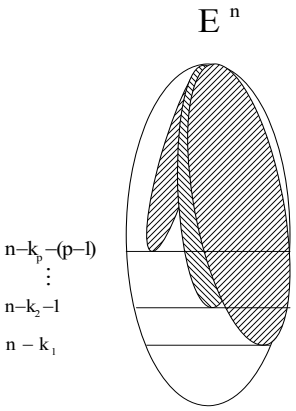
where parameters k_i correspond to the binary presentation of number m:

$$m = 2^{k_1} + 2^{k_2} + \cdots + 2^{k_p}.$$

Numerical expressions of coordinates of characteristic vector $S = (s_1, s_2, \cdots, s_n)$ of function $\mu(m,n)$ are also calculable:

$$s_{k_j+1} = \left(\sum_{l=1}^{j-1} 2^{k_l - 1} \right) + 2^{k_j}, \ j \in \overline{1,p}$$

for digits $k_1, k_2, ..., k_p$ of binary presentation, and



$$s_i = \left(\sum_{l=1}^{j} 2^{k_l - 1} \right) + \left(\sum_{l=j+1}^{p} 2^{k_l} \right) \text{ for } i, k_{j+1} + 2 \le i \le k_j, \ j \in \overline{1, p-1}$$

and

$$s_i = \sum_{l=1}^{p} 2^{k_l - 1} = m/2 \text{ for } i, 1 \le i \le k_p$$

$$s_i = \sum_{l=1}^{p} 2^{k_l} = m \text{ for } i, k_1 + 2 \le i \le n$$

4. Synthesis by Bisections

Subsets $M \subseteq E^n$ might be presented (coded) by $(0,1)$-matrices with m rows and n columns, where the rows correspond to the vertices-elements of M. All the rows are different, and the quantities of 1's in columns (column sums) correspond to the sizes of (one of the two) partition-subsets of M in corresponding directions.

Thus the existence and synthesis issues of vertex subsets by given quantitative characteristics of their partitions might be reformulated as the corresponding existence and synthesis problems for $(0,1)$-matrices.

Let it is given an integer vector $S = (s_1, \cdots, s_n)$, $0 \le s_i \le m$, $i = 1, \cdots, n$. If there exists an $m \times n$ $(0,1)$-matrix with all different rows and with s_i column sums, then by the finite number of row pairs replacements it can be transformed into the equivalent matrix of "canonical form" – where each column consists of continuous intervals of 1's and (then - below) 0's such that they compose bisections of intervals of the previous stage (column). Therefore the problem of synthesis of the given $(0,1)$-matrices in supposition of the existence might be solved, in particular, by the algorithms which compose the matrices in column-by-column bisection fashion.

The first column is being constructed by allocating of s_1 1's to the first s_1 rows-positions followed by the $m - s_1$ 0's in others. Two intervals is the result: – the s_1 interval of 1's, and the $m - s_1$ interval of 0's. Within each of these intervals we have the same row, and pairs of (i, j) rows with i and j from different intervals are different.

The second column has been constructed by a similar bi-partitioning of intervals of the first column - situating first 1's and then 0's on these intervals such that the summary length of 1-intervals is equal to s_2, and the summary length of 0-intervals is equal to $m - s_2$. The current k-th column has been constructed by consecutive and continuous bi-partitioning of intervals of the $k-1$ column – providing only that the summary length of all 1's-intervals is equal to s_k.

When in some column we get all 1 length intervals, then all the rows of matrix become different by the set of constructed columns, and the remainder columns might be constructed arbitrarily.

Partitioning of the intervals in each step can be performed by different ways - following different goals. Let assume that the partitioning of intervals aims to maximize some quantitative characteristics, which might lead to

the matrices with different rows. One of such characteristics - the number of pairs of different rows – is considered in [S, 1995], where is proven that the optimal partitioning is achieved when intervals are partitioned by the equi-difference principle.

In an analogue sophisticated situation when we are not restricted by column sums (or when we allow the whole set of descriptions) and the aim is only in minimization of number of columns for differentiations of rows or in maximization of the number of different row pairs, the best partitioning is known by the "interval bisection" method, which requires the number of columns - $n = \lceil log_2 m \rceil$ [K, 1998].

For the given m let's describe all the possible column sum vectors of matrices composed by the interval bisection method.

In case of $m = 2^n$ it is evident, that $s_i = m - s_i = 2^{n-1}$ for $i = 1, \cdots, n$.

For an arbitrary m, ($m > 1$), an odd length interval may appear during the separate column partitioning stage. Partitioning the odd length interval takes an extra 1 or 0, which leads to the different column sums. Factually it is satisfactory to consider the case when in each stage an extra 1 is allocated on each odd length interval during the partitioning and let this leads to the column sums $s_1, s_2, \cdots, s_n$. Column sums corresponding to all the other allocations of extra 1's and 0's can be received through the inversions and monotone transformation of $s_1, s_2, \cdots, s_n$ [S, 1995].

So the number of odd length intervals of each column is equivalent to the difference of 1's and 0's used in the next column, and the goal becomes to calculate the number of odd length intervals (denote it by d_i for i-th column).

Let $2^{n-1} \le m < 2^n$ and $m = 2^{k_1} + 2^{k_2} + \cdots + 2^{k_p}$ is the binary presentation of number m, where $n - 1 = k_1 > k_2 > \cdots > k_p \ge 0$.

Since $m \ge 2^{n-1}$, then the i-th column for $i = 1, \cdots, n-1$ will create 2^i intervals. These intervals will have the length approximately equal to

$$2^{k_1 - i} + 2^{k_2 - i} + \cdots + 2^{k_p - i}. \quad (1)$$

The same time, the n-th column constructs the m intervals of length 1.

Consider the following cases:

- $k_p - i > 0$

  the i-th column has no intervals of odd length, therefore $s_{i+1} = m - s_{i+1} = m/2$.

- $k_p - i = 0$

  the i-th column will have 2^i odd length intervals, therefore $s_{i+1} - (m - s_{i+1}) = 2^i$ and $s_{i+1} = \frac{m + 2^i}{2}$.

- $k_1, k_2, \cdots, k_j \ge i$ and $k_{j+1}, \cdots, k_p < i$

$$\underbrace{2^{k_1 - i} + \cdots + 2^{k_j - i}}_{\substack{k_j - i \ge 0 \\ (I)}} + \underbrace{2^{k_{j+1} - i} + \cdots + 2^{k_p - i}}_{\substack{k_{j+1} - i < 0 \\ (II)}} \quad (2)$$

1. $k_j - i = 0$,

   Component (I) in (2) provides that each of 2^i intervals of i-th stage is at least of length $2^{k_1 - i} + \cdots + 2^{k_j - i} = q$ (which is odd) and the component (II) - that the length of $2^{k_{j+1} - i} + \cdots + 2^{k_p - i}$ share of 2^i intervals is $q + 1$. Therefore the number of even intervals equals

$\left(\sum_{l=j+1}^{p} 2^{k_l-i}\right)\cdot 2^i = \sum_{l=j+1}^{p} 2^{k_l}$ and the number of odd length intervals is equal to

$d_i = 2^i - \sum_{l=j+1}^{p} 2^{k_l} = 2^{k_j} - \sum_{l=j+1}^{p} 2^{k_l}$

Hence: $s_{i+1} = \dfrac{\sum_{l=1}^{p} 2^{k_l} + 2^{k_j} - \sum_{l=j+1}^{p} 2^{k_l}}{2} = \left(\sum_{l=1}^{j-1} 2^{k_l-1}\right) + 2^{k_j}$ (3)

for $i = k_j$ and the same formula holds for each k_j, $j = 1,2,\cdots,p$.

2. $k_j - i > 0$,

in this case component (I) in (2) provides that each of 2^i intervals of i-th stage is of even length and the component (II) - that the $2^{k_{j+1}-i} + \cdots + 2^{k_p-i}$ share of 2^i intervals is of odd length.

The number of odd length intervals is equal to

$d_i = \left(\sum_{l=j+1}^{p} 2^{k_l-i}\right)\cdot 2^i = \sum_{l=j+1}^{p} 2^{k_l}$ and $s_{i+1} = \dfrac{\sum_{l=1}^{p} 2^{k_l} + \sum_{l=j+1}^{p} 2^{k_l}}{2} = \left(\sum_{l=1}^{j} 2^{k_l-1}\right) + \left(\sum_{l=j+1}^{p} 2^{k_l}\right)$ (4)

for $i \neq k_j$ and the same formula holds for each k_j, $j = 1,2,\cdots,p$..

Thus (3) and (4) describe column sums of matrices composed by interval bisection method by the given m. Comparing this column sums to the coordinates of characteristic vector of monotone Boolean function $\mu(m,n)$ points out that the $(0,1)$-matrix A, which corresponds to the set of all true-vectors of monotone Boolean function $\mu(m,n)$ has the same set of column sums as the matrix – constructed by the bisection method.

Conclusion

Resuming, - the n-cube subsets and partitioning (Set Systems and characterization by inclusion of a particular element) in specific boundary cases, and the bisection strategy characterization are strongly similar having the same characteristic numerical descriptors, simply related to the binary representations of the set sizes.

Bibliography

[B, 1989] C. Berge, Hypergraphs. North Holland, Amsterdam, 1989.

[BI, 1988] Billington D., Conditions for degree sequences to be realisable by 3-uniform hypergraphs". The Journal of Combinatorial Mathematics and Combinatorial Computing". 3, 1988, 71-91.

[C, 1986] Colbourn Charles J., Kocay W.L. and Stinson D.R., Some NP-complete problems for hypergraph degree sequences. Discrete Applied Mathematics 14, p. 239-254 (1986).

[K, 1998] D. Knuth, The Art of Computer Programming, second edition, vol.3. Sorting and Searching, Addison-Wesley, 1998

[R, 1966] H. J. Ryser. Combinatorial Mathematics, 1966.

[S, 1986] H. A. Sahakyan. Greedy Algorithms for Synthesis of (0,1) Matrices, Doklady Akademii Nauk Arm. SSR, v. 83, 5, pp. 207-209 (1986)

[S, 1997] H. Sahakyan. On a class of (0,1)-matrices connected to the subsets partitioning of E^n, Doklady NAS Armenia, v. 97, 2, pp. 12-16.

[S, 1995] H. Sahakyan. Hierarchical Procedures with the Additional Constraints, II Russia, with participation of NIS Conference, Pattern Recognition and Image Analysis: New Information Technologies, Ulianovsk, 1995, pp.76-78.

[SA, 2001] L. Aslanyan, H. Sahakyan. On the boundary cases of partitioning of subsets of the n-dimensional unit cube, Computer Science & Information Technologies Conference, Yerevan, September 17-20, 2001, 164-166.

Authors' Information

Hasmik Sahakyan – Institute for Informatics and Automation Problems, NAS Armenia, P.Sevak St. 1, Yerevan-14, Armenia; e-mail: hasmik@ipia.sci.am

Levon Aslanyan – Institute for Informatics and Automation Problems, NAS Armenia, P.Sevak St. 1, Yerevan-14, Armenia; e-mail: lasl@sci.am

OPTIMAL CONTROL OF A SECOND ORDER PARABOLIC HEAT EQUATION

Mahmoud Farag, Mainouna Al-Manthari

Abstract: In this paper, we are concerned with the optimal control boundary control of a second order parabolic heat equation. Using the results in [Evtushenko, 1997] and spatial central finite difference with diagonally implicit Runge-Kutta method (DIRK) is applied to solve the parabolic heat equation. The conjugate gradient method (CGM) is applied to solve the distributed control problem. Numerical results are reported.

Keywords: Distributed control problems, Second order parabolic heat equation, Runge-Kutta method, CGM.

ACM Classification Keywords: F.2.1 Numerical Algorithms and Problems; G.4 Mathematical Software

Introduction

In the recent years, optimal control of systems governed by partial differential equations have been extensively studied. We refer for instance to [Lions, 1971], [Farag, 2004] for parabolic problems and to [Wu,2003], [Borzi, 2002] for numerical studies. In this paper, we are concerned with the optimal control boundary control of a second order parabolic heat equation. Using the results in [Evtushenko, 1997] and spatial central finite difference with diagonally implicit Runge-Kutta method of order 2 in 2 stages is applied to solve the parabolic heat equation. The conjugate gradient method (CGM) is applied to solve the distributed control problem. Numerical results are reported.

Consider the second order heat equation

$$(1) \qquad \frac{\partial y(x,t)}{\partial t} = a^2 \frac{\partial^2 y(x,t)}{\partial x^2} + u(x,t)\,,\ (x,t) \in \Omega = (0,l)\,x\,(0,T)$$ where $y(x,t)$ is the temperature at time t and at a point x and $u(x,t)$ is a distributed control.

The initial and boundary conditions are given by

$$(2) \qquad y(x,0) = \varphi(x)\,,\ x \in [0,l]\,,$$

$$(3) \qquad \frac{\partial y(0,t)}{\partial x} = 0\,,\ \frac{\partial y(l,t)}{\partial x} = \nu\,[\,g(t) - y(l,t)\,]\,,\ t \in (0,T)$$ where $g(t)$ is a boundary control.

The problem is to find control functions $u(x,t)$ and $g(t)$ that minimize the cost functional

$$(4) \qquad J = \int_0^l \Phi(y(s,T))\,ds$$

where Φ is continuously differentiable with respect to its argument.

DIRK Method

In this section we present some basic results about the Runge-Kutta methods, the diagonally implicit Runge-Kutta method of order 2 in 2 stages (DIRK). The reader is referred to [Alexander, 1977], [Shamardan, 1998].

In [Alexander, 1977], he has given the A-stable DIRK methods of maximum order in two stages and derived new methods with stronger stability properties, from this work one can extract the following theorem.

Theorem 1: There are exactly two strongly s-stable DIRK formulae of order two in two stages and exactly are strongly s-stable DIRK formulae of order three in three stages. They are

$$(5)\qquad \begin{array}{ccc|c} \alpha & 0 & \alpha & \\ 1-\alpha & \alpha & 1 & \\ - & - & - & \\ 1-\alpha & \alpha & & \end{array}, \qquad \begin{array}{ccc|c} \alpha & 0 & 0 & \alpha \\ c_2-\alpha & \alpha & 0 & c_2 \\ b_1 & b_2 & \alpha & 1 \\ - & - & - & - \\ b_1 & b_2 & \alpha & \end{array}$$

with $\alpha = 1 \pm 0.5\sqrt{2}$ in the first and α is the root of $x^3 - 3x^2 + \frac{2}{3}x - \frac{1}{6} = 0$ lying in $(\frac{1}{6}, \frac{1}{2})$ and,

$b_1 = -0.25\,(6\alpha^2 - 16\alpha + 1)$, $b_2 = 0.25(6\alpha^2 - 20\alpha + 5)$ in the second.

In [Shamardan, 1998], we presented the numerical solution of the linear equation

$$(6)\qquad \begin{cases} \dfrac{\partial y(x,t)}{\partial t} = a\,\dfrac{\partial^2 y(x,t)}{\partial x^2} - c\,\dfrac{\partial y(x,t)}{\partial x} \\ y(x,0) = \sin \pi x\,(x,t), \quad x \in (0,1),\ y(0,t) = y(1,t) = 0,\ t \in (0,1) \end{cases}$$

using DIRKB (Diagonally Implicit Runge-Kutta Backword Method), DIRKC (Diagonally Implicit Runge-Kutta Central Method), DIRKF (Diagonally Implicit Runge-Kutta Forward Method) and given the following figures.

In the figures (1.a),(1.b),(1.c), (2.a),(2.b),(2.c) the wave rapidly decreases as there is no shock wave but in the figures (3.a),(3.b),(3.c), (4.a),(4.b),(4.c) there is a medium boundary region near x=1.

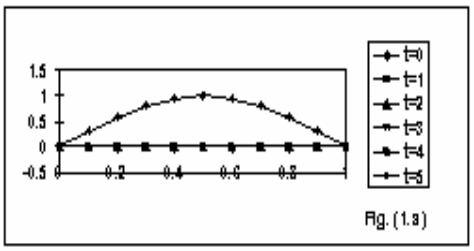


Fig. (1.a)

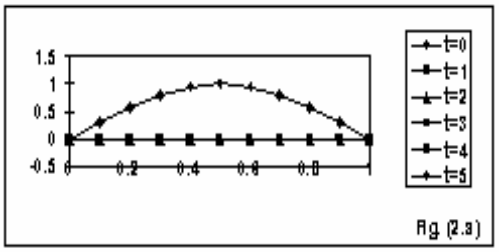


Fig. (2.a)

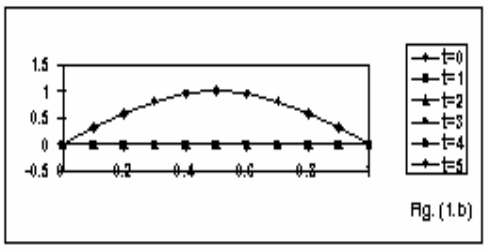


Fig. (1.b)

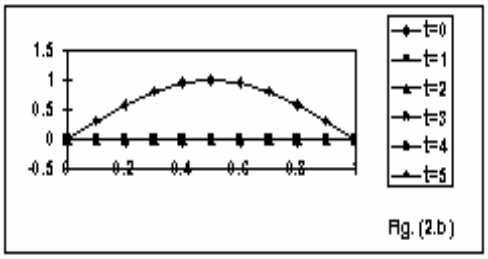


Fig. (2.b)

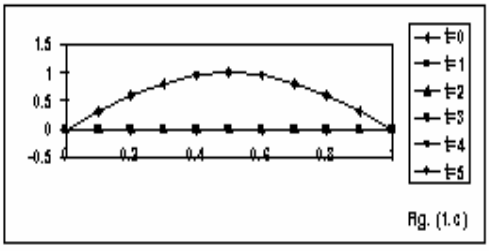


Fig. (1.c)

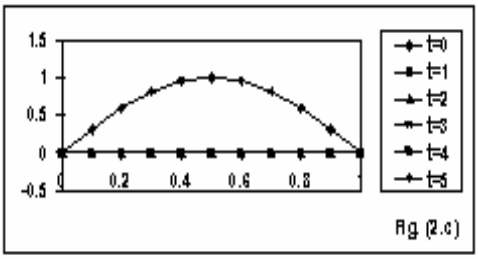


Fig. (2.c)

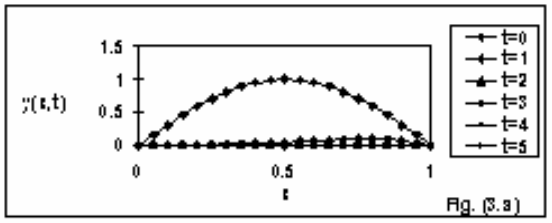


Fig. (3.a)

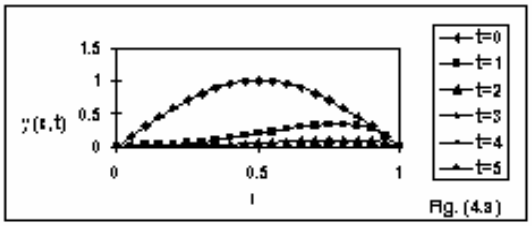


Fig. (4.a)

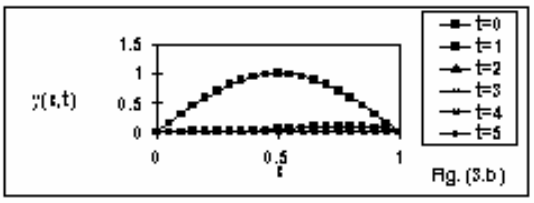


Fig. (3.b)

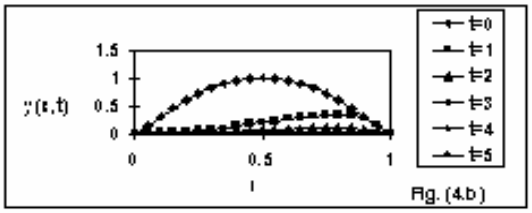


Fig. (4.b)

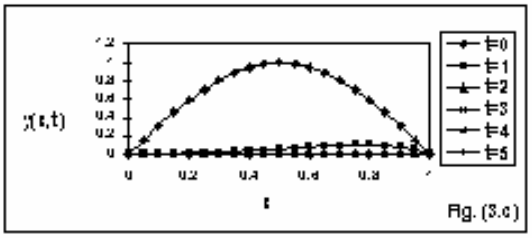


Fig. (3.c)

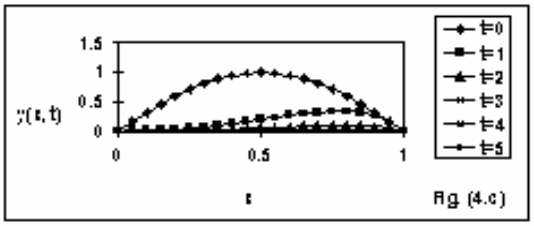


Fig. (4.c)

Differentiation of the Cost Functional

For given control functions $u(x,t)$ and $g(t)$, we solve equation (1) with conditions (2), (3), and then substitute this solution into (4) to evaluate J. This value is a composite function of $u(x,t)$ and $g(t)$. Denote it by $\Theta(u,g)$. Since the optimal control cannot be obtained as an analytic solution of the necessary and sufficient optimality conditions, we attempt to find it numerically by minimizing $\Theta(u,g)$ via a gradient algorithm. We are thus faced with computing the gradient of the cost functional [Evtushenko, 1997]. The problem is discredited by a finite difference approximation scheme. We use a uniform grid denote

$$(7)\quad \begin{cases} x_i = i\Delta x \ , t_j = j\Delta t,\ i = 0,..,\ k,\ j = 0,...,\ m, \\ \Delta x = \dfrac{l}{k},\ \Delta t = \dfrac{t}{m},\ y_i^j = y(i\Delta x, j\Delta t),\ u_i^j = u(i\Delta x, j\Delta t), \\ \varphi_i = \varphi(i\Delta x),\ g^j = g(j\Delta t),,\ i = 0,..,\ k,\ j = 0,...,\ m \end{cases}$$

Using an explicit forward Euler scheme in time, then the cost functional and the system (1)-(3) are replaced by

$$(8)\qquad \overline{J} = \Delta x \sum_{i=0}^{k} \alpha_i\ \Phi(y_i^m),$$

$$(9)\quad y_i^j = \begin{cases} (1-2\lambda)\, y_i^{j-1} + \lambda (y_{i-1}^{j-1} + y_{i+1}^{j-1}) + \Delta t\, u_i^{j-1} & 1 \le i \le k-1 \quad 1 \le j \le m \\ y_1^j & i = 0 \quad 1 \le j \le m \\ \mu y_{k-1}^j + \mu \nu \Delta x\, g^j & i = k \quad 1 \le j \le m \\ \varphi_i & 0 \le i \le k \quad j = 0 \end{cases}$$

where α_i are the quadrature coefficients, $\lambda = \dfrac{a^2 \Delta t}{(\Delta x)^2}$, $\mu = \dfrac{1}{1+\nu\Delta x}$.

We introduce the adjoint variables Ψ_i^j and the auxiliary function

$$(10)\qquad \mathrm{E}=\bar{J}+\sum_{j=1}^{m}\left[\mu\left(y_{k-1}^{j}+\nu\,\Delta x\,g^{j}\right)\Psi_{k}^{j}+y_{1}^{j}\Psi_{0}^{j}\right]+\sum_{i=0}^{k}\varphi_{i}\,\Psi_{i}^{0}$$
$$+\sum_{i=1}^{k-1}\sum_{j=1}^{m}\left[(1-2\lambda)\,y_{i}^{j-1}+\lambda\left(y_{i-1}^{j-1}+y_{i+1}^{j-1}\right)+\Delta t\,u_{i}^{j-1}\right]\Psi_{i}^{j}$$

Appling {formula 11, in [Evtushenko, 1997]}, we obtain

$$(11)\quad \Psi_{i}^{j}=\begin{cases}(1-2\lambda)\Psi_{i}^{j+1}+\lambda\left(\Psi_{i-1}^{j-1}+\Psi_{i+1}^{j+1}\right) & 2\le i\le k-2 \quad 0\le j\le m-1\\ (1-2\lambda)\Psi_{1}^{j+1}+\lambda\,\Psi_{2}^{j+1}+\Psi_{0}^{j} & i=1 \quad 1\le j\le m-1\\ (1-2\lambda)\Psi_{k-1}^{j+1}+\mu\,\Psi_{k}^{j}+\lambda\,\Psi_{k-2}^{j+1} & i=k-1 \quad 1\le j\le m-1\\ \mu\,\Psi_{k}^{m}\,\delta_{i}^{k-1}+\Psi_{0}^{m}\,\delta_{i}^{1}+\Delta x\;\alpha_{i}\;\Phi(y_{i}^{m}) & 0\le i\le k \quad j=m\\ (1-2\lambda)\,\Psi_{i}^{1}+\lambda\;\Psi_{i+1}^{1} & i=1,i=k-1 \quad j=0\\ \lambda\Psi_{k}^{j+1} & i=0 \quad 0\le j\le m-1\\ \lambda\Psi_{k-1}^{j+1} & i=k \quad 0\le j\le m-1\end{cases}$$

Then using {formula 12, in [Evtushenko, 1997]}, we obtain

$$(12)\qquad \begin{array}{lll}\dfrac{d\,\Theta}{d\,u_{i}^{j}}=\Delta t\;\Psi_{i}^{j+1} & 1\le i\le k-1 & 0\le j\le m-1\\ \dfrac{d\,\Theta}{d\,u_{0}^{j}}=\dfrac{d\,\Theta}{d\,u_{k}^{j}}=0 & & 0\le j\le m-1\\ \dfrac{d\,\Theta}{d\,u_{i}^{m}}=0 & 0\le i\le k & \\ \dfrac{d\,\Theta}{d\,g^{j}}=\mu\;\nu\;\Delta x\;\Psi_{k}^{j} & & 1\le j\le m-1\end{array}$$

If we let $k\to\infty$, $\Delta x\to 0$, $\Delta t\to 0$, then in both cases we find that the function satisfies the following conditions:

$$(13)\qquad \begin{array}{l}\dfrac{\partial\Psi}{\partial t}+a^{2}\dfrac{\partial^{2}\Psi}{\partial x^{2}}=0\;,(x,t)\in(0,1)\,x\,(0,T)\\ \Psi(x,t)=\Phi_{y}(y(x,T))\,,\quad x\in[0,l]\\ \dfrac{\partial\Psi(0,t)}{\partial x}=0,\;\dfrac{\partial\Psi(l,t)}{\partial x}=-\nu\,\Psi(l,t)\;,t\in(0,T)\end{array}$$

The gradients of the cost functional for the continuous problem are given by

$$(14)\qquad \frac{d\,\Theta}{d\,u(x,t)}=\Psi(x,t)\quad,\quad\frac{d\,\Theta}{d\,g(t)}=\nu\,a^{2}\,\Psi(l,t).$$

Solution Algorithm

With the gradient obtained, the following gradient type algorithm [Farag, 2003] can then be developed for the optimal values of u^{*},g^{*} based on the conjugate gradient method (CGM). The direct and adjoint systems are converted to ordinary differential equations and solving by DIRK method. The outlined of the algorithm for solving control problem are as follows:

Step 1: Choose an initial guess $u^{(n)}(x,t),\ g^{(n)}(t)\in U$.

Step 2: Solve the direct problem to obtain $y(x,t,u^{(n)},g^{(n)})$.

Step 3: Solve the adjoint problem to find the gradient of the cost functional

$$(15)\quad \begin{cases} [\dfrac{d\,J}{d\,u(x,t)}]^{(n)} = [\dfrac{d\,\Theta}{d\,u(x,t)}]^{(n)} = \Psi(\,x,t,\,u^{(n)},\,g^{(n)}\,) \\ [\dfrac{d\,J}{d\,g(t)}]^{(n)} = [\dfrac{d\,\Theta}{d\,g(t)}]^{(n)} = \nu\,a^2\,\Psi(l,t\,,u^{(n)}(l,t),\,g^{(n)}(t)). \end{cases}$$

Step 4: Compute the conjugate coefficient by:

$$(16)\quad \chi^{(n)} = \frac{\iint\limits_{\Omega}\{[\dfrac{d\,J}{d\,u(x,t)}]^{(n)}\}^2 dx\,dt \;+\int\limits_0^T\{[\dfrac{d\,J}{d\,g(t)}]^{(n)}\}^2 dt}{\iint\limits_{\Omega}\{[\dfrac{d\,J}{d\,u(x,t)}]^{(n-1)}\}^2 dx\,dt \;+\int\limits_0^T\{[\dfrac{d\,J}{d\,g(t)}]^{(n-1)}\}^2 dt}$$

Step 5 : Calculate the direction of descent :

$$(17)\quad u^{(n)} = [\frac{d\,J}{d\,u(x,t)}]^{(n)} + \chi^{(n)}\,u^{(n-1)}\,,\; g^{(n)} = [\frac{d\,J}{d\,g(t)}]^{(n)} + \chi^{(n)}\,g^{(n-1)}$$

Step 6: Test the optimality of $u^{(n+1)}$, $g^{(n+1)}$.

If $u^{(n+1)}$, $g^{(n+1)}$ are optimum, stop the process. Otherwise, go to Step 7.

Step 7: Set $u^{(n+1)} = u^{(n)}$, $g^{(n+1)} = g^{(n)}$, $n = n+1$ and go to Step 2.

Numerical Example

Let us present a numerical example. The programs were written in FORTRAN. We choose

$$a=1\,,\ \nu=1\,,\ \varphi(t)=x^2\,,\ g(t)=50\,e^t+t+3\,,\ x\in(0,1)\,,\ \ t\in(0,1)$$

In Figure 3.1 the bold curve is the exact optimal boundary control and the other curves are the values of optimal control $g(t)$ via iterations. In figure 3.2 $u(x,t)$ is plotted for the approximation optimal control of the control problem. Figure 3.3 shows the values of cost functional via iterations.

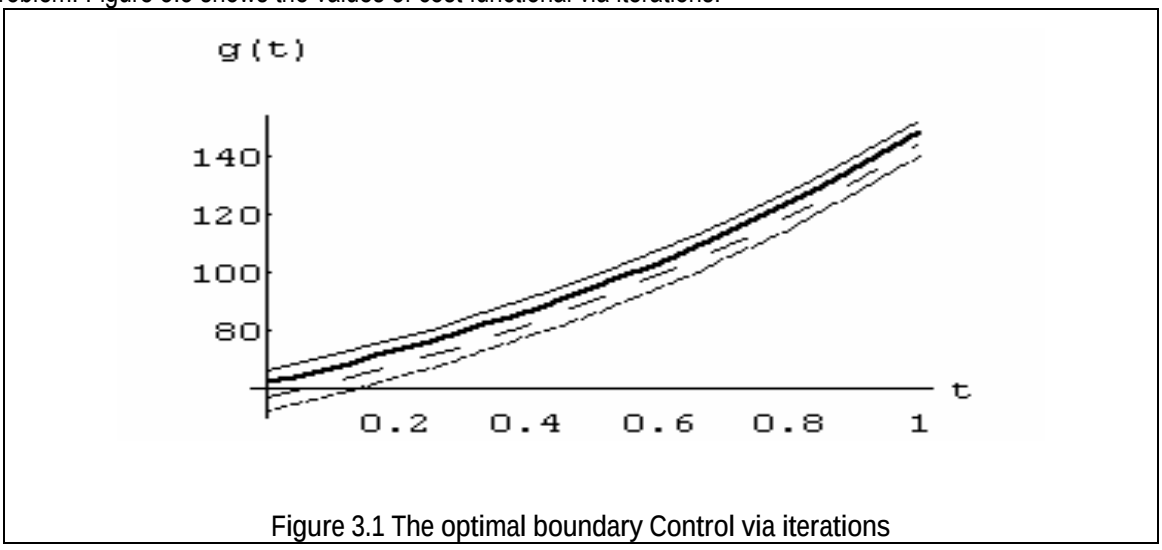


Figure 3.1 The optimal boundary Control via iterations

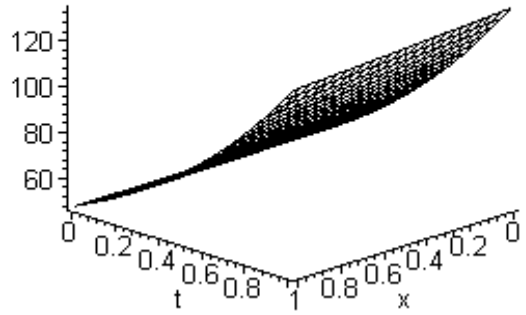


Figure 3.2 The exact optimal control $u(x,t)$

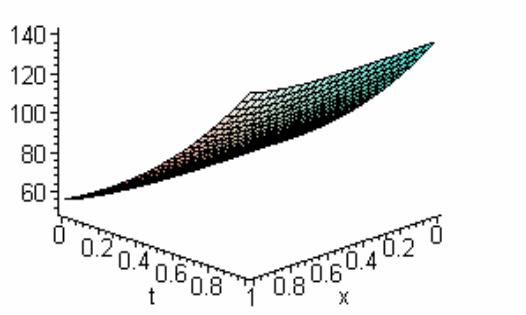
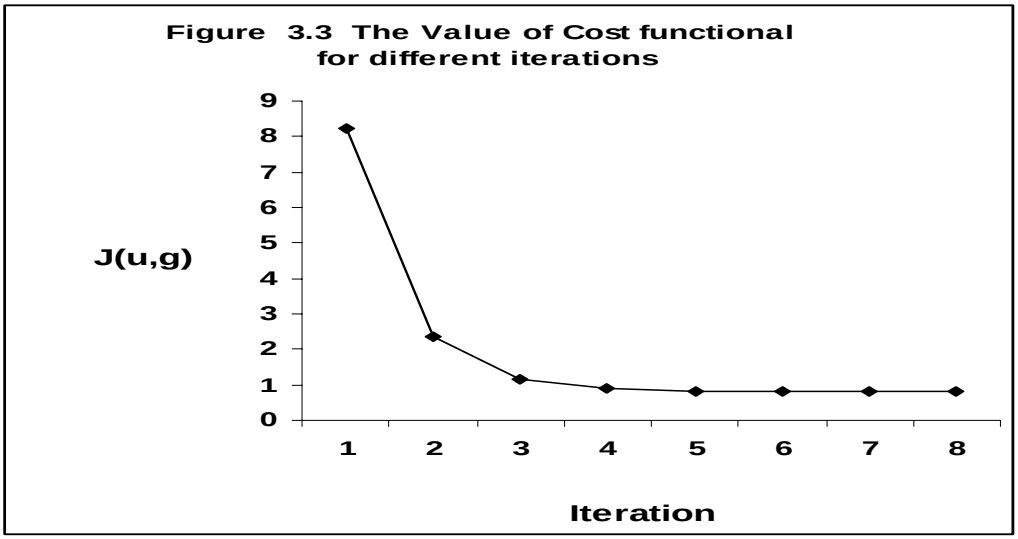


Figure 3.3 The approximation optimal control $u(x,t)$



Conclusion

Optimal control problems for partial differential equations are currently of much interest. A large amount of the theoretical concept which governed by quasilinear parabolic equations has been investigated in the field of optimal control problems. These problems have dealt with the processes of hydro- and gas dynamics, heat physics, filtration, the physics of plasma and others. In this paper; we are concerned with the optimal control boundary control of a second order parabolic heat equation. Using the results in [Evtushenko, 1997] and spatial central finite difference with diagonally implicit Runge-Kutta method (DIRK) is applied to solve the parabolic heat equation. The conjugate gradient method (CGM) is applied to solve the distributed control problem. Numerical results are reported.

Acknowledgments

The authors gratefully acknowledge *Prof. Dr. Krassimir Markov* Editor in chief of IJ ITA, who made useful suggestions and remarks, which helped me to improve the paper.

Bibliography

[Alexander, 1977] R. Alexander. Diagonally implicit Runge-Kutta methods for stiff ordinary differential equations, SIAM J. Numeric. Analysis, 30, 428-451, 19977.

[Borzi, 2002] A. Borzi. Fast multigrid methods for parabolic optimal control problems, Proc. Of the 18th GAMM-Seminar Leipzig, 1-10, 2002.

[Evtushenko, 1997] Yuri Evtushenko. Computation of exact gradients in distributed dynamic systems, Int. J. Control, 6(1), 81—89,1997.

[Farag, 2003] M. H. Farag. A gradient-type optimization technique for the optimal control for Schrödinger equations, I. Tech., Bulgaria, 16-25, 2003.

[Farag, 2004] M. H. Farag. A stability theorem for a class of distributed parameter control systems, accepted in Rocky Mountain J. of mathematics, 2004.

[Lion, 1972] J.-L Lions. Optimal control by systems described by partial differential equations, Mir, Moscow, SSSR, 1972.

[Farag, 2004] M. H. Farag. A stability theorem for a class of distributed parameter control systems, accepted in Rocky Mountain J. of mathematics, 2004.

[Shamardan] Ali B. Shamardan, M. H. Farag. and S. M. Refaat. Spatial central finite difference with DIRK(3,3) temporal schemes for nonlinear dispersive waves, Ricerche di Matematica, Vol. XLVII, fase 2,383-397,1998.

[Wu,2003] Y. H. Wu, M. Chuedoung and P. F. Siew. Optimal boundary control of a linear parabolic evaluation systems, Austra. Math. Soc., 123-134,2003.

Authors' Information

Mahmoud H. Farag – Department of Mathematics and Computer Science, Faculty of Education, Ibri, Sultanate of Oman, e-mail: farag5358@yahoo.com
Mainouna S.M. Al-Manthari – Department of Mathematics and Computer Science, Faculty of Education, Ibri, Sultanate of Oman, e-mail: may78may@hotmail.com

THE MATRIX METHOD OF DETERMINING THE FAULT TOLERANCE DEGREE OF A COMPUTER NETWORK TOPOLOGY

Sergey Krivoi, Miroslaw Hajder, Pawel Dymora, Miroslaw Mazurek

Abstract: This work presents a theoretical-graph method of determining the fault tolerance degree of the computer network interconnections and nodes. Experimental results received from simulations of this method over a distributed computing network environment are also presented.

Keywords: computer network, fault tolerance, coherent graph, regular graph, network topology, adjacency matrix.

ACM Classification Keywords: C.2.1 Network Architecture and Design - network topology, F.2.1 Numerical Algorithms and Problems - matrix methods, B.8.1 Reliability, Testing, and Fault-Tolerance - fault tolerance degree

Introduction

Computer networks plays an extremely important role in today's information technologies, because by its means it's possible to accelerate processes like i.e. transmission, processing and storage of information in computer systems. In such a process the most crucial issues are related with protecting a correct work of a computer network and its interconnection and node fault tolerance. The solution of these problems is related with examining the network topological characteristics and its topological structures. In this work the theoretical-graph method of determining the computer network topology critical points which refers to computer network interconnections and nodes failures is proposed.

1. Preliminary Information and Definitions

Computer network (CN) consisted of $n > 1$ computers connected between themselves is presented as a graph $G = (V, E)$, where $V = \{v_1, v_2, ..., v_n\}$ represents a real number of nodes and $E = \{(u, v): u, v \in V\}$ – a real number of interconnections. Under these assumptions, the network nodes are represented by graph G nodes, and network interconnections corresponds with the given graph connection links. If not assumed differently a graph definition is always meant as an undirected graph. If $e = (u, v) \in E$, than node u (node v) is called an end of a link e, and such nodes – adjacent. If node u turns out to be the end of link e, than link e and node u, is called incident. Note, that the adjacency relation for a graph node turns out to be symmetric. If $u \in V$, than $n(u)$ a number of graph links incidental to node u is called a degree of a node u. A path from a node u to a node v of the graph $G = (V, E)$ with a length k is called a link sequence $(u_1, u_2), ..., (u_k, u_{k+1}) \in E$ such as $u = u_1$, $v = u_{k+1}$.

Definition 1. Graph $G = (V, E)$ is called coherent if from any node u there exists a path to its any other node v (symmetric-relation – inverted path: from a node v to a node u).

An operation of link and node removal is considered on examples of a Cartesian product and an isomorphic joint of two graphs.

Graph $G' = (V', E') = G - v$ is called a graph received from a graph $G = (V, E)$, as a result of applying a node $v \in V$ removal operation if $V' = V \setminus \{v\}$, and E' squares with a number E, from which all links incident to a node v were removed.

Graph $G' = (V', E') = G - e$ is named a graph received from a graph $G = (V, E)$ as a result of applying a link $e \in E$ removal operation, if $V' = V$, and $E' = E \setminus \{e\}$.

Notice, that for both operations the following equations are true:

$$(G - u) - v = (G - v) - u; \quad (G - e) - e' = (G - e') - e,$$

i.e. the result graphs doesn't depend on the sequence of links or nodes removal order .

The nodes number M (or links) of a graph $G = (V, E)$ is called crucial, if as a result of its elements removal from a given graph this graph becomes incoherent.

Let graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ be given. Graph $G = (V, E) = G_1 \times G_2$ is called a graph received as a result of applying a Cartesian product operation on graphs G_1 and G_2, if $V = V_1 \times V_2$ and $E = \{[(u, u'), (v, v')]: u = v$ in a graph G_1 and $(u', v') \in E_2$ or $u' = v'$ in a graph G_2 and $(u, v) \in E_1\}$. As an example, if this operation is applied on graphs presented in the following figure 1 the result graph is called a toroid.

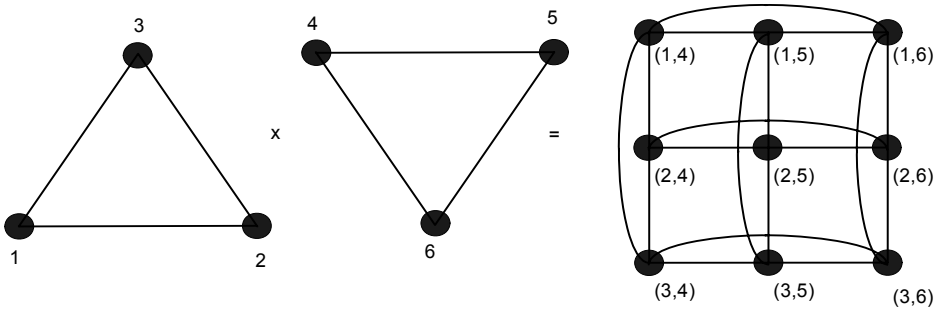


Fig. 1. Example of a Cartesian product $G_1 \times G_2$ operation

Definition 2. Graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ are called isomorphic, if between the sets of nodes V_1 and V_2 exists a bijection mapping $f: V_1 \to V_2$, such that nodes u and v are adjacent to the graph G_1, than nodes $f(v)$ and $f(v)$ are adjacent in the graph G_2. Mapping f is called isomorphic.

Let isomorphic graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ and f – isomorphic mapping are given.

Graph $G = (V, E) = G_1 *_f G_2$ is called a graph received as a result of applying an isomorphic joint operation on graphs G_1 and G_2, if $V = V_1 \cup V_2$ and also $E = E_1 \cup E_2 \cup \{(u, u'): u \in V_1, u' \in V_2$ and $f(u) = u'\}$.

As an example, if this operation is applied on graphs presented in the figure 2 with isomorphic *f(i) = i + 8* the result graph is called a hypercube.

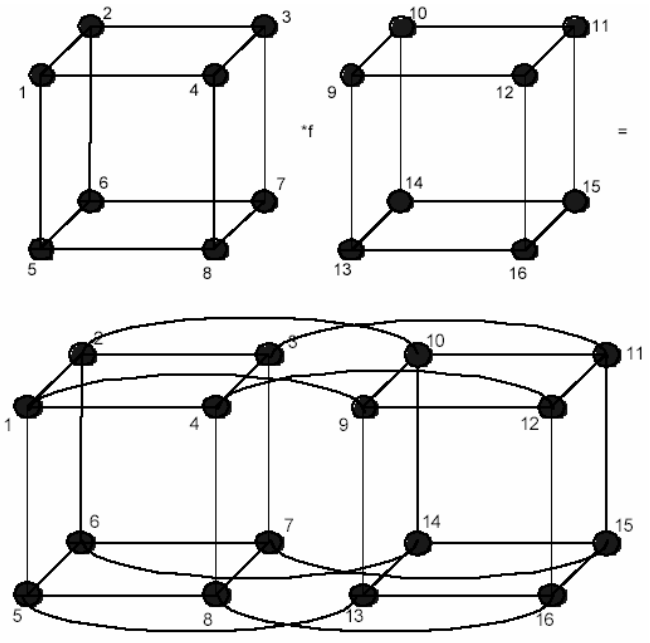


Fig. 2. Example of an isomorphic $G_1 f G_2$ operation

The next theorem is following immediately from definitions.

Theorem 1. If graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ – are coherent, than a graph received as a result of applying a Cartesian product operation or an isomorphic joint operation is also coherent.

It's obvious that mentioned links and nodes removal operations in coherent graphs may create incoherent graphs. With each graph $G = (V, E)$ a matrix $A_G = ||a_{ij}||$ is related, where $i, j = 1, 2, ..., n$, and is called an adjacency matrix and is described as follows:

$$a_{ij} = \begin{cases} 1, \ if \ \ (v_i, v_j) \in E, \\ 0, otherwise. \end{cases}$$

Therefore, graph G with n nodes corresponds to a square matrix $n \times n$, filled up with 0 and 1 values. Introduced links and nodes removal operations in graphs may be carried out in an easy manner on these graphs adjacency matrixes.

2. The Matrix Method of Determining the Fault Tolerance Degree of a CN

Let a CN consists of n computers and is represented as a graph $G = (V, E)$. With a help of a graph G analysis for a given CN the most critical points are specified. That's why formal statements are introduced.

Let $G = (V, E)$ – is the given coherent graph representing CN, while A_G – is the adjacency matrix of this graph. The nodes number $V' \subseteq V$ (links $E' \subseteq E$) of a graph $G = (V, E)$, is called a computer critical point (failure place) if V' (E') the minimal nodes set (links) of a graph G becomes crucial, than any element removal from the graph G causes that this graph becomes incoherent.

Fulfilling these definitions and an adjacency matrix of a graph G, representing a CN, it's possible to create a method of determining the critical points in a CN. In this end the matrix interpretation of introduced before graph operation is considered.

2.1. The Matrix Interpretation of the Graph Operations

As follows from determining critical points this method turns out to be useful in finding the minimal existing subsets. In order to do this a graph adjacency matrix is used because for an inherent graph its adjacency matrix becomes a diagonally-block. With a reorder of an appropriate rows and columns it becomes as follows:

$$A_G = \begin{bmatrix} A \mid 0 \\ 0 \mid B \end{bmatrix},$$

where *A* and *B* are nonzero matrixes, and 0 – represents zero matrixes. On the basis of this simple adjacency matrix property, the matrix method of determining the critical points of a CN is based.

Interpretation of nodes and links removal operation in an adjacency matrix turns out to be very simple. In fact, the removal of links (v_i, v_j) in a graph *G* reduces to changing of an element a_{ij} values from 1 to 0 in a matrix A_G, and a removal of a node v_i in a graph *G* corresponds with a removal of an *i*-ary row of this matrix.

For example, for a graph presented in the figure 3a its adjacency matrix has such a structure (rows and columns of the matrix corresponds to enumeration 1,2,3,4,5,6 from the left to the right and from up to down):

$$A_G = \begin{vmatrix} 011000 \\ 101110 \\ 110000 \\ 010011 \\ 010101 \\ 000110 \end{vmatrix}.$$

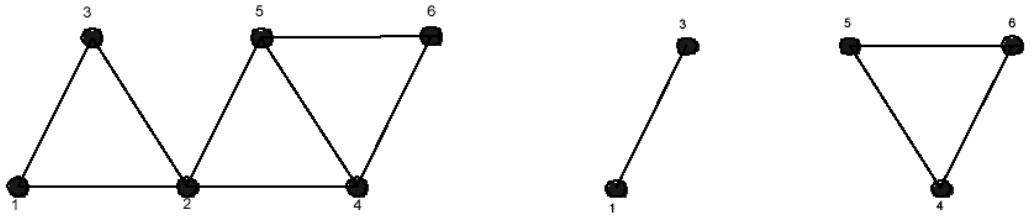


Fig. 3. a) Example of the graph with 6 nodes; b) Graph with removed links and nodes.

The node 2 removal from this graph reduces to a matrix $A_G = \begin{vmatrix} 01000 \\ 10000 \\ 00011 \\ 00101 \\ 00110 \end{vmatrix}$, for which graph is shown in the Fig. 3b.

The Cartesian product operation on graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ on the basis of an adjacency matrix is possible to present on an example. From this operation definition it results that an adjacency matrix of a graph $G = G_1 \times G_2$ has a following form:

$$A_G = \begin{vmatrix} A_{G_1} \mid E_a \mid E_a \\ E_a \mid A_{G_1} \mid E_a \\ E_a \mid E_a \mid A_{G_1} \end{vmatrix},$$

where a size of an adjacency matrix A_{G1} is equal to $|V_2|$, while E_a - a diagonal matrix of the size *a* which on its diagonal accepts values 0 or 1 depending on whether the other pair nodes in a graph G_2 are adjacent to it or not.

For example, an adjacency matrix of a graph $G = G_1 \times G_2$, where graphs G_1 and G_2 are presented in the figure 1, has a following form (matrix rows and columns correspond to an enumeration (1,4), (2,4), (3,4), (1,5), (2,5), (3,5), (1,6), (2,6), (3,6)):

$$A_G = \left|\begin{array}{l} 011\,|\,100\,|\,100 \\ 101\,|\,010\,|\,010 \\ 110\,|\,001\,|\,001 \\ 100\,|\,011\,|\,100 \\ 010\,|\,101\,|\,010 \\ 001\,|\,110\,|\,001 \\ 100\,|\,100\,|\,011 \\ 010\,|\,010\,|\,101 \\ 001\,|\,001\,|\,110 \end{array}\right| .$$

The isomorphic joint $G = G_1 *_f G_2$, of graphs G_1 and G_2 has an adjacency matrix as it follows from the applied operation:

$$A_G = \left|\begin{array}{l} A_{G_1}\,|\,C \\ C\,|\,A_{G_2} \end{array}\right| ,$$

where in a diagonal the adjacency matrixes of graphs G_1 and G_2 are placed respectively, and an element a_{ij} of a matrix C is equal 1, if $f(v_i) = v_j$, or equal 0 otherwise. For example, if this operation is applied on a graph from the figure 2, then an adjacency matrix of this graph has a following form (see on the right side of the text):

$$A_G = \left|\begin{array}{l} 01011000\,|\,10000000 \\ 10100100\,|\,01000000 \\ 01010010\,|\,00100000 \\ 10100001\,|\,00010000 \\ 10000101\,|\,00001000 \\ 01001010\,|\,00000100 \\ 00100101\,|\,00000010 \\ 00011010\,|\,00000001 \\ \cdots\cdots\cdots\cdots\cdots\cdots\cdots \\ 10000000\,|\,01011000 \\ 01000000\,|\,10100100 \\ 00100000\,|\,01010010 \\ 00010000\,|\,10100001 \\ 00001000\,|\,10000101 \\ 00000100\,|\,01001010 \\ 00000010\,|\,00100101 \\ 00000001\,|\,00011010 \end{array}\right|$$

2.2. The Matrix Method of the CN Topology Analysis

Let $G = (V, E)$ – represents a coherent graph describing a CN and A_G – the adjacency matrix of this graph. As follows, with a simple analysis of a graph adjacency matrix it's possible to determine critical points of a CN, which is represented by this graph.

Generally, in a simple way it's possible to determine nodes, which has the lowest degree because the numbers of 1 in a row (or column) is equal to a node degree, which corresponds with this row (or column). After the nodes with the minimal degree are chosen then the starting number of links which might be suspected of critical links point are received. However, it's not guaranteed that they will appear because the minimal degree doesn't causes that the number of incident to them links is the minimal existing number. Suspected of number of nodes (links) is removed from the adjacency matrixes and after this, a received matrix has a block form. If a matrix has such a form then removed number of nodes (links) turns out to be crucial. Such an analysis requires a consideration of 2^n variants.

The general algorithm, which results from the above-presented analysis, might be presented in a following way (A – an adjacency matrix of a graph G, and n – a sequence presenting a CN):

Critical points (A)

1. Determine critical points M of a graph G, consisted of one element of a matrix A.
2. for $i = 2$ to 2^n fulfill
   Determine critical points M' of a graph G, consisted of i elements of a matrix A
   If $/M'/ < /M/$ than $M:=M'$;
   End of cycle.
3. Return (M).

The first operator is processed directly from the same matrix *A* (the before presented manner), while the second operator << determines critical points *M'* of a graph *G*, consisted of *i* elements of a matrix *A,* >> presents by itself computations of a transition closure of an achievement relation in a graph. It's common knowledge that a given algorithm is based on an estimation of *n-i-x* (or n) matrix degree received from the matrix *A* as a result of its node (or link) removal.

Obviously, a cycle described in a second paragraph can be parallelized because computations of different iterations are independent from each other. In the following paragraph an experimental results of computations on a cluster with 10 processors are presented.

3. Experimental Results

In order to parallel ours computations the Parallel Virtual Machine (PVM) was used. Next, the PVM was implemented on a cluster with 10 processors. The PVM is a main parallel library, which may process on heterogeneous computer networks. Created in PVM the Message Passing Interface (MPI) which is a library of procedures and functions became a modern standard of building parallel applications. The MPI is independent of operating systems' platform.

Nodes number	Time [s]
5	0,0028
10	0,0047
50	0,0228
100	0,0431
250	0,34
500	0,6402
750	0,7913
1000	1,4252
1500	2,1958
2500	3,47
5000	7,3218
7500	8,9551
10000	22,8534

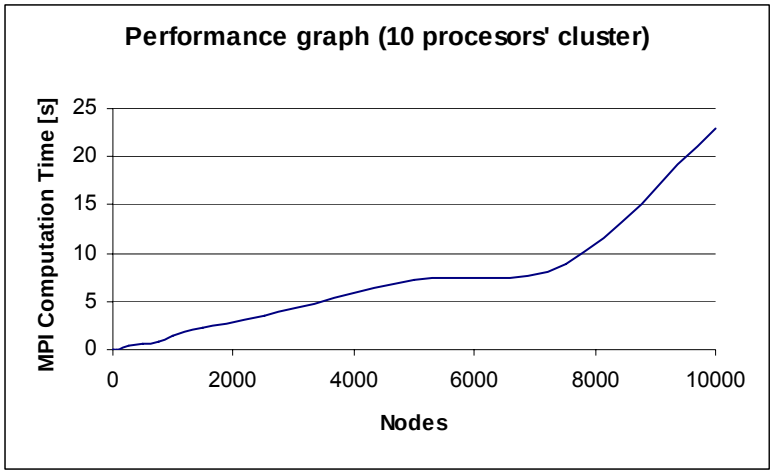


Fig. 4. Experimental results

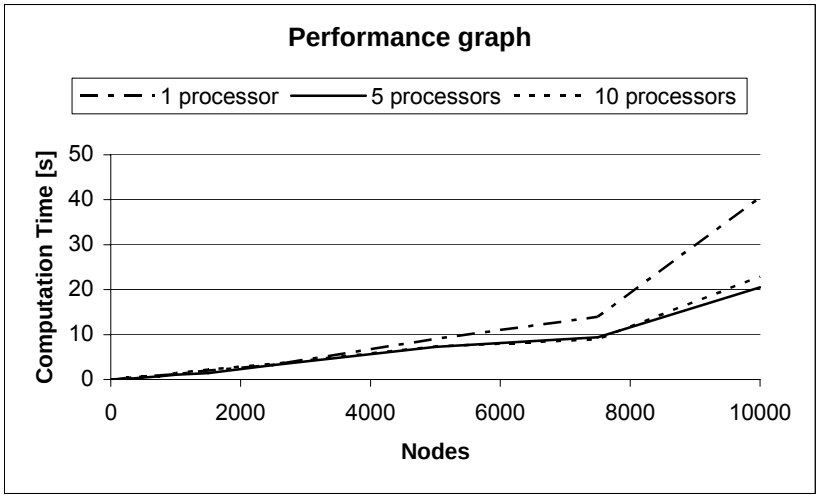


Fig. 5. Experimental results

Conclusions

The presented method of determining the critical points is based on the graph theory and some features of the adjacency matrix, which represents graphs. Searching for critical points in computer networks, as it follows from the above researches is characterized with a large complexity and requires applying of a great computational performance. However, the main advantage of this method is a fact that is uses homogenous structures, and the computations itself are of the same type. Presented experiments were implemented and realized on the multiprocessor cluster and the results of these experiments are presented in the above table. Analyzing these experimental results, the conclusion may be drawn independently.

Bibliography

[Капитонова Ю.В., Кривой С.Л., 1] Капитонова Ю.В., Кривой С.Л., Летичевский А.А., Луцкий Г.М. – *Лекции по дискретной математике*. – Санкт-Петербург: БХВ-Петербург. –2004. 614 стр.

[Свами М., Тхуласираман К., 2] Свами М., Тхуласираман К. Графы, сети и алгоритмы. Москва: Мир. – 1984. 454 стр.

[Hajder M., Loutski G., 3] Hajder M., Loutskij G., Streciwilk W. - *Informatyka*. Rzeszow: WsiZ. - 2002. – 714 s. (на польском языке)

[Hajder M., Dymora P., 4] Hajder M., Dymora P. - *Algorithmical and topological methods of fault tolerace assurance*; Wydawnictwo Uniwersytetu Marii Curie-Skłodowskiej; Annales Informatica, Kazimierz Dolny 2004 – 143-153,

[Hajder M., Dymora P., Bolanowski M., 5] M. Hajder, P. Dymora, M. Bolanowski: *Fault tolerance in multichannel scalability systems;* International Conference of Science and Technology Scalnet'04, Scalable Systems and Computer Networks Design and Aplications, Ministry of Education and Science of Ukraine Kremenchuk State Polytechnical University, Kremenchuk 2004, – pp. 52-57,

[Spohn D.,L., 6] Spohn D. L. - *Data Network Design. New York: McGrew-Hill*. – 1997. – 347 p.

[Harsall F., 7] Harsall F. - *Data Communications, Computer Design Networks and Open Systems*. 4-th Eddition. – Addison-Wesley Company. – Harlow. – 1996. – 412 p.

[Mazurek M., 8] Mazurek M. - *Hyper-deBruijn topology as a result of the Network Product*, International Conference of Science and Technology Scalnet'04, Scalable Systems and Computer Networks Design and Aplications, Ministry of Education and Science of Ukraine Kremenchuk State Polytechnical University, Kremenchuk 2004, pp. 62-65.

[Hajder M., Mazurek M., 9] Hajder M., Mazurek M. - *Embending as a method to improve multichanel bus topology parameters*, Annales Universitatis Mariae Curie-Skłodowska, Sectio Ai Informatica, Lublin, 2004.

Authors' Information

Krivoi Sergey – Glushkov Institute of Cybernetics NAS of Ukraine, Ukraine, Kiev, 03187, 40 Glushkova Street, Institute of Cybernetics, e-mail: krivoi@i.com.ua

Hajder Miroslaw – Technical University of Rzeszow, Rzeszow, Poland, e-mail: mhajder@prz.rzeszow.pl

Dymora Pawel – Technical University of Rzeszow, Rzeszow, Poland, e-mail: pdymora@prz.rzeszow.pl

Mazurek Miroslaw – Technical University of Rzeszow, Rzeszow, Poland, e-mail: mmazurek@prz.rzeszow.pl

RKHS-METHODS AT SERIES SUMMATION FOR SOFTWARE IMPLEMENTATION

Svetlana Chumachenko, Ludmila Kirichenko

Abstract: Reproducing Kernel Hilbert Space (RKHS) and Reproducing Transformation Methods for Series Summation that allow analytically obtaining alternative representations for series in the finite form are developed.

Keywords: The reproducing transformation method, Hilbert space, reproducing kernel, RKHS, Series Summation Method.

ACM Classification Keywords: G.1.10 Mathematics of Computing: Applications

Introduction

Operating speed of digital logic devices depends on type of silicon: PLD, Gate Array or ASIC. FPGAs are the lowest in risk, low in project budget but have the highest cost per unit. Gate Arrays utilize less custom mask making than standard cell and stand in the middle from all of three and fallen from wide use today. Cell based ASICs have the highest performance and lowest cost per unit in case of mass production, but they also have the longest and most expensive design cycle. Also, digital designs can be divided on CPU based systems on chip (SoC) and non-CPU logic devices. CPU as universal processing unit can solve broad spectrum of various tasks from all areas of human activity. Nevertheless, there exist bottlenecks where CPU can't satisfy required performance. Usually it happens during implementation of mathematical tasks that require big number of iterations and hence big time expenses to obtain desired result with desired accuracy.

To increase efficiency of solving of computational tasks there are used mathematical co-processors, which implement most efficient ways of computing equations, integrals, differential coefficients. It is obvious that after discovering of new methods of increasing computation accuracy and decreasing computation time it is necessary to re-implement mathematical co-processors or use new generation of IP-cores in PLD, Gate Array, ASIC designs. It is presented, easy to implement as IP-core, method of reduction of computation of certain types of series to exact function that is widely used during calculation of parameters of high radio frequency devices. Presented method decrease computation time of such tasks in tens and hundred times and its inaccuracy is equals to zero.

Statement of the Problem

The investigation is based on fundamental works of Aronzajn [1], Razmahnin, Yakovlev [2]. It develops the following research [10-12] on Series Summation in Reproducing Hilbert Space and their approbation [19-22]. Modern papers of Saitoh, Laslo Mate, Daubeshies and others [3-9] are used for revues and staying problem. Classical papers of Tranter, Doetsch, Ango, Titchmarch, Ahiezer [13-18] are used for inter-comparison of results.

Mathematical models based on Reproducing Kernel Hilbert Space methods are used in Wavelets Analysis, namely: at Pattern Recognition, Digital Data Processing, Image Compression, Computer Graphics; and also in Learning theory: for example, at Exact Incremental Learning, in Statistical Learning theory, in Regularization theory and Support Vector Machines. In mentioned arias we have not deal with exact Series Summation because it isn't necessary for considered cases. We use sum and finite summation, not series. But there are areas of scientific study where exact series summation it is necessary.

For such problems Reproducing Transformations Method and its part – Series Summation Method in RKHS – can be useful [20, 22]. We are going to point out these areas.

The purpose of the investigation is to originate a new Series Summation Method based on RKHS-theory and to demonstrate the new results which develop theoretical statements of Series Summation Method in RKHS.

The research problems are:

- Series Summation in RKHS
- Applications of Series Summation in RKHS
- Reproducing Transformations Method as a perspective of this research

Base Theoretical Statements and Investigation Essence

Reproducing Kernel Hilbert Space is a subspace of Hilbert space with Reproducing Kernel (RK). RK is a function Ker of two variables with two properties: 1) $\forall t \in T \quad Ker(s_0,t) \in T$; 2) $\forall f \in H \quad f(t) =< f(s), Ker(s,t) >$, where <...> – inner (scalar) product can be represented as a series on selective values. There is an operator G, which transfers any function from Hilbert space L_2 into function from RKHS and leaves without change function from RKHS H.

Thus, there is an operator G, which transfers any function from Hilbert space L_2 into function from RKHS and doesn't change any function from RKHS H.

For example, the functions with finite spectrum of cosine- and sine-transformations and Hankel-transformation form RKHS. The basic research of expansion problem on selective values was executed by K. Shannon and

V.A. Kotelnikov. There are statements determining particular cases RKHS. Thus, any function from RKHS can be represented as selective value expansion. If there is series where the common summand can be reduced to a standard form, – it means to extract reproducing kernel by equivalent transformations, – then for any series one can put in accordance a function from RKHS. In other words, a series can be summarized by known formulas.

Thus, the main idea of proposed method is to obtain and to use the following relation:

$$f(s) = \sum_k f(t_k)\,Ker(s,t_k),$$

in right-hand side of this relation we can see a series on selective values of function f(t); left-hand side represents value of function f in point s.

We use four base kinds of Reproducing Kernels, which originate four RKHS accordingly [21]:

1. RKHS H_1 is a space of functions, which have finite Fourier-transformations.

2. Space H_2 contains a class of functions with finite Hankel-transformation.

3. Space H_3 consists of functions with finite sine-transformations.

4. Space H_4 has all functions, those cosine-transformations are finite.

For these spaces there are four Kernel Functions and Series on selective values accordingly [21].

Based RKHS-theory the new approach to definition of series sum is proposed. It is called *Series Summation Method in RKHS.*

It allows analytically obtaining alternative representations for some kinds of series in the finite form [10].

The new formulas for calculating the sum of series (including alternating) have been obtained by proving several theorems [10, 12].

Reproducing Transformations Method are generalization and extension of Series Summation method in RKHS [20, 22]. It can be useful at solving mentioned problems and other important points. It needs further evolution and consideration.

For example, we can see proving the following formula.

Theorem 1. There is the following relation for alternating series

$$\sum_{k=1}^{\infty}(-1)^k \frac{kF(k)}{a^n - k^n} = \begin{cases} \dfrac{\pi a F(a)}{a-k}, & n=1; \\ \dfrac{\pi F(a)}{2pa^{2p-2}\sin\pi a}, & n=2p, p=1,2,3,...; \\ \dfrac{\pi F(a)}{(2p-1)a^p \sin\pi a}, & n=2p-1, p=2,3,.... \end{cases} \quad (1)$$

for any $F(x)$ from RKHS, $a \neq 0,\pm1,\pm2,...$.

Proof. Let's consider alternating series the common member of which contains the difference of powers in denominator:

$$\sum_{k=1}^{\infty}(-1)^k \frac{kF(k)}{a^n - k^n}.$$

Let's define its sum. For this purpose we would consider the cases, when n is equal to natural number.

1) Let $n=1$. The common member of series transforms to kind [10] that yields:

$$\sum_{k=1}^{\infty}(-1)^k \frac{kF(k)}{a-k} = \sum_{k=1}^{\infty}F(k)\frac{\cos(k\pi)\sin(\pi a)}{(a+k)\sin\pi a}\frac{a+k}{a-k}\frac{2\pi k}{2\pi} = \frac{\pi}{2\sin\pi a}\sum_{k=1}^{\infty}\frac{2k}{a+k}\frac{\sin\pi(a-k)}{\pi(a-k)}(a+k)F(k) =$$

$$= \frac{\pi}{2\sin\pi a}[(a+k)F(k)]\big|_{k=a} = \frac{2\pi a F(a)}{2\sin\pi a} = \frac{\pi a F(a)}{\sin\pi a}.$$

2) For $n=2$ the result obtained in [10]:

$$\sum_{k=1}^{\infty}(-1)^k\frac{kF(k)}{a^2-k^2}=\frac{\pi}{2\sin\pi a}F(a). \qquad (2)$$

3) For $n=3$ we can obtain result by recurrent way with accounting formula (2) and using the decomposition of difference of cubes:

$$\sum_{k=1}^{\infty}(-1)^k\frac{kF(k)}{a^3-k^3}=\sum_{k=1}^{\infty}\frac{(-1)^k k}{(a-k)}\frac{F(k)}{(a^2+ak+k^2)}=\sum_{k=1}^{\infty}\frac{(-1)^k k}{(a^2-k^2)}\frac{(a+k)F(k)}{(a^2+ak+k^2)}=$$

$$=\left[\Phi(k)=\frac{(a+k)F(k)}{(a^2+ak+k^2)}\right]=\frac{\pi}{2\sin\pi a}\Phi(a)=\frac{\pi}{2\sin\pi a}\frac{2aF(a)}{3a^2}=\frac{\pi F(a)}{3a\sin\pi a}.$$

4) For $n=4$ we can obtain:

$$\sum_{k=1}^{\infty}(-1)^k\frac{kF(k)}{a^4-k^4}=\sum_{k=1}^{\infty}\frac{(-1)^k k}{(a^2-k^2)}\frac{F(k)}{(a^2+k^2)}=\left[\Phi(k)=\frac{F(k)}{(a^2+k^2)}\right]=\frac{\pi}{2\sin\pi a}\Phi(a)=$$

$$=\frac{\pi}{2\sin\pi a}\frac{F(a)}{2a^2}=\frac{\pi F(a)}{4a^2\sin\pi a}.$$

5) For $n=5$ we can obtain with decomposition by difference of fifth powers the following result:

$$\sum_{k=1}^{\infty}(-1)^k\frac{kF(k)}{a^5-k^5}=\sum_{k=1}^{\infty}(-1)^k\frac{kF(k)}{(a-k)(a^4+ka^3+k^2a^2+k^3a+k^4)}=$$

$$=\sum_{k=1}^{\infty}\frac{(-1)^k k}{(a^2-k^2)}\frac{(a+k)F(k)}{(a^4+ka^3+k^2a^2+k^3a+k^4)}=$$

$$=\left[\Phi(k)=\frac{(a+k)F(k)}{(a^4+ka^3+k^2a^2+k^3a+k^4)}\right]==\frac{\pi}{2\sin\pi a}\Phi(a)=\frac{\pi}{2\sin\pi a}\frac{2aF(a)}{5a^4}=\frac{\pi F(a)}{5a^3\sin\pi a}.$$

6) For $n=6$ we can analogically obtain:

$$\sum_{k=1}^{\infty}(-1)^k\frac{kF(k)}{a^6-k^6}=\frac{\pi F(a)}{6a^4\sin\pi a}.$$

Thus, based on mentioned transformations we can conclude:

$$\sum_{k=1}^{\infty}(-1)^k\frac{kF(k)}{a-k}=\frac{\pi aF(a)}{a-k},\ n=1;$$

$$\sum_{k=1}^{\infty}(-1)^k\frac{kF(k)}{a^{2p}-k^{2p}}=\frac{\pi F(a)}{2pa^{2p-2}\sin\pi a},\ n=2p, p=1,2,3,...;$$

$$\sum_{k=1}^{\infty}(-1)^k\frac{kF(k)}{a^{2p-1}-k^{2p-1}}=\frac{\pi F(a)}{(2p-1)a^p\sin\pi a},\ n=2p-1, p=2,3,....$$

Thus, the theorem 1 has been proof.

The following examples illustrate application of the theorem 1.

Example 1. To proof of the following formula truth:

$$\sum_{k=1}^{\infty}(-1)^k\frac{k\sin kx}{a^2-k^2}=\frac{\pi}{2}\frac{\sin ax}{\sin\pi a},\ -\pi<x<\pi,\ a>0, a\neq 1,2,...,$$

the residues theory is used in [23]. However, application of the theorem 1 gives the same result:

$$\sum_{k=1}^{\infty}(-1)^k\frac{k\sin kx}{a^2-k^2}=\frac{\pi}{2\sin\pi a}\sin kx\big|_{k=a}=\frac{\pi\sin ax}{2\sin\pi a},\ a>0, a\neq 1,2,....$$

Example 2. To proof of the following identity truth Laplace transformation is used [24]. However application of the theorem 1 reduces to the same result but it is simpler solution:

$$\sum_{k=1}^{\infty}\frac{(-1)^{k+1}k}{k^2-a^2}J_{2n+1}(kx)=\sum_{k=1}^{\infty}\frac{(-1)^{k}k}{a^2-k^2}J_{2n+1}(kx)=\frac{\pi}{2}J_{2n+1}(ax)\cos ec(a\pi),\ -\pi<x<\pi.$$

Example 3. To proof of the following identity truth

$$\sum_{k=1}^{\infty}(-1)^k\frac{k\cos kx}{a^4-k^4}=\frac{\pi\cos ax}{4a^2\sin\pi a}, \tag{3}$$

we can apply the theorem 1. Also we can show numerically this result (see fig. 1, 2). On Fig. 1 there are two diagrams in the equal co-ordinates for parameter $a=2{,}5$. Graphs of function from right-hand side of (3) (the bold curve) and left-hand side (thin curve) of (3). Fig. 2 demonstrates the absolute uncertainty.

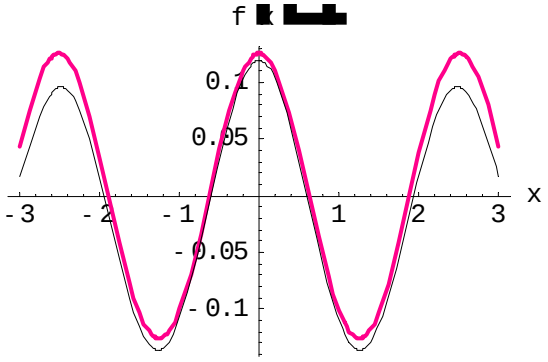


Fig. 1. Results comparison

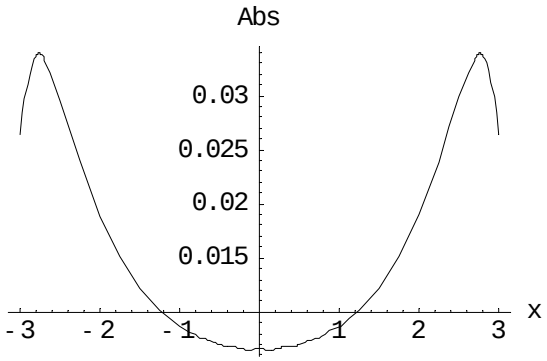


Fig. 2. Absolute error

Conclusion

Thus, we can direct the following *areas* for applications of new Series Summation Method:

– Exact summation of series;

– solving summatory equations and its systems;

– solving integral equations and its systems;

– solving integral-summatory equations and systems of complex form;

– proving integral identities.

Mentioned areas can be used at solving some problems of: antenna theory; diffraction theory; electrodynamics and can be useful at Software/Hardware implementations (See Fig. 3).

The obtained results allow making the following *conclusions*:

1) RKHS-theory can be used for summation of selected series. For this purpose, *Series Summation Method in RKHS* has been proposed.
2) *Advantages of this method* consist of:
   – application of equivalent transformations to the common member of a series, that enables to obtain the *analytical solution* for smaller quantity of steps;
   – in absence of necessity to use the tables of integral transformations and to use the integration in complex area.
3) The application of obtained results of RKHS-theory for *solving the boundary electrodynamics problems* gives possibility *to simplify known methods* and to receive on their basis the *analytical solutions*, that is represented as essential for the further numerical experiment;
4) New mathematical results for solving the summatory and integral equations are obtained by proving some theorems.

5) The obtained results can be included into the reference mathematical library and implemented into Mathematics program products, MathCAD, Math Lab means. It can be useful for scientists, engineers, mathematics at solving the different problems.

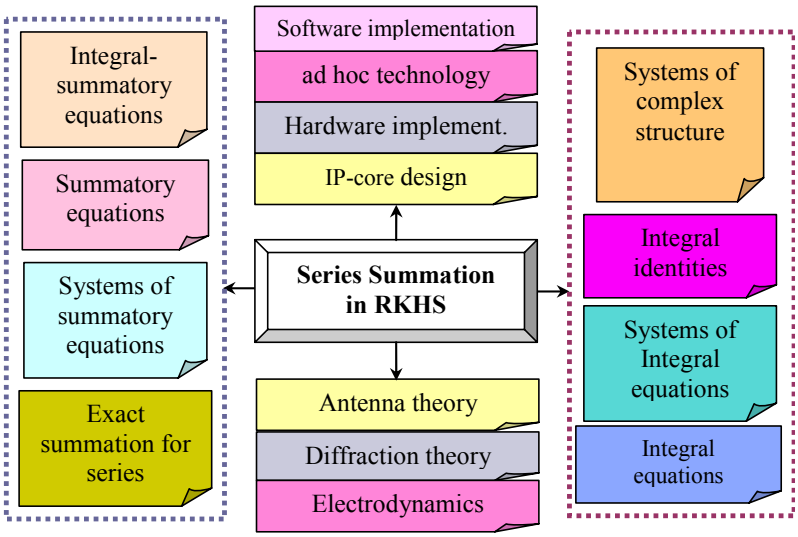


Fig. 3. Application arias of Series Summation in RKHS

Bibliography

[1] *Aronzajn A.* Theory of reproducing kernels // Trans. American Math. Soc., 68:337-404, 1950.

[2] *Functions* with double orthogonality in radio electronics and optician / By edition M.K. Razmahnin and B.P. Yakovlev. M.: Sov. radio, 1971. 256p. (in Russian).

[3] *Saitoh S.* Integral Transforms, Reproducing Kernels and Their Applications, Pitman Res. Notes in Math. Series, 369 (1997), Addison Wesley Longman, UK.

[4] *Laslo Mate*. Hilbert Space Methods in Science and Engineering, 1990. 200 p.

[5] *Daubeshies I.* Ten lectures on wavelets. CBMS-NSF. Regional conference series in applied mathematics. SIAM, Philadelphia, 1992.

[6] *Sethu Vijayakumar, Hidemitsu Ogawa.* RKHS based Functional Analysis for Exact Incremental Learning / Neurocomputing: Special Issue on Theoretical analysis of real valued function classes, Vol.29, No.1-3, pp.85-113, Elsevier Science (1999). http://www-clmc.edu/sethu/research_detail.1html.

[7] *Saitoh S.* New Norm Type Inequalities for linear mappings // Jornal of Inequalities in Pure and Applied Mathemathics. Victoria Univ., 2003. Volume 4. Issue 3. Article 57. http: // jipam.vu.edu.au.

[8] *Girosy F.* An equivalence between sparse approximation and support vector machines // Neural Computation, 10(6):155-1480, 1998.

[9] *Vapnik V.* The nature of Statistical Learning Theory. Springer-Verlag. New York, 1995.

[10] *Chumachenko S.V.* Summation method of selected series for IP-core design // Radioelectronics and Informatics. 2003. N3. C. 197-203 (in Russian).

[11] *Chumachenko S.V.* Theorem about some integral identities based on Series Summation Method in RKHS // Radioelectronics and Informatics. 2004. №1. C. 113-115 (in Russian).

[12] *Chumachenko S.V.* Reproducing kernel Hilbert spaces and some theirs applications // Radioelectronics and Informatics. 2003. № 4. C. 141-144 (in Russian).

[13] *Tranter C.J.* Integral Transforms in Mathematical Physics. London: Methuen & Co. Ltd., New York: John Wiley & Sons, Inc. 1951.

[14] *Doetsch G.* Anleitung Zum Praktischen Gebrauch Der Laplace-transformation und Der Z-transformation. R. Oldenbourg. Munhen, Wien, 1967.

[15] *Angot A.* Complements de Mathematiques. A L'usage Des ingenieurs de L'elektrotechnique et Des Telecommunications. Paris, 1957.

[16] *Watson G.N.* A Treatise on the Theory of Bessel Functions. Cambridge: Cambridge University Press. 1952.

[17] *Titchmarch E.C.* The theory of functions. Oxford Univ. Press, 1939.

[18] *Ahiezer N.I.* The lectures of the approximating theory. Moscow: Nauka, 1965. 408p. (in Russian).

[19] *Chumachenko S.V., Govhar Malik, Imran Saif Chattha.* Series Summation Method in HSRK // Proc. of the international Conference TCSET'2004 "Modern Problems of Radio Engineering Telecommunications and Computer Science". February 24-28. 2004. Lviv-Slavsko, Ukraine. P.248-250.

[20] *Chumachenko S.V.* Solving Electrodynamics Problems by Reproducing Transformations Method // Proc. BEC 2004. Tallinn. October 3-6, 2004. PP. 319-322.

[21] *Chumachenko S.V., Gowher Malik, Khawar Parvez*. Reproducing Kernel Hilbert Space Methods FOR CAD Tools // Proc. EWDTW, 2004. Yalta-Alushta, September 23-26. PP. 247-250.

[22] *Svetlana Chumachenko, Vladimir Hahanov.* Reproducing Transformations method for IP-core of summatory and integral equations solving // Proc. of DSD 2004 Euromicro Symposium on Digital System Design: Architectures, Methods and Tools. August 31 - September 3, 2004, Rennes – France (Work in progress).

[23] *Titchmarch E.C.* The theory of functions. Oxford Univ. Press, 1939.

[24] *Prudnikov A.P., Bryichkov Yu.A., Marichev O.I.* Integrals and series. Moscow: Nauka, 1981. 800p. (in Russian).

Authors' Information

Chumachenko Svetlana – Ph. D. Senior Scientist and a professor assistant. Address: Kharkov National University of Radio Electronics, Ukraine, 61166, Kharkov, Lenin Avenue, 14, Phone: (+380)-57-7021-326, e-mail: ri@kture.kharkov.ua.

Kirichenko Ludmila – Ph. D. Senior Scientist and a professor assistant. Address: Kharkov National University of Radio Electronics, Ukraine, 61166, Kharkov, Lenin Avenue, 14, Phone: (+380)-57-7021-335, e-mail: ludmila@kture.kharkov.ua.

FUZZY SETS: ABSTRACTION AXIOM, STATISTICAL INTERPRETATION, OBSERVATIONS OF FUZZY SETS

Volodymyr Donchenko

Abstract: The issues relating fuzzy sets definition are under consideration including the analogue for separation axiom, statistical interpretation and membership function representation by the conditional Probabilities.

Keywords: fuzzy sets, membership function, conditional distribution

ACM Classification Keywords: I.5.1. Pattern Recognition: Models - Fuzzy set

Introduction

Fussy sets introduced by Lotfi Zadeh [Zadeh, 1965] (see also [Kaufmann, 1982]) were considered on the one hand as the modeling method alternative to statistical ones which ought to realized the idea of uncertainty, vagueness in the situation under consideration, on the other – as the theory which should to generalize the classical Set theory.

Such generalized pretensions syringed up on the fuzzy logic, which considered as the algebra with two binary operations namely max and min on the real numbers from the interval [0,1] for which all the properties of classical Boolean operations are valid excepting the low of excluded middle. Nevertheless nothing out of this frame in the fuzzy logic was considered. Particularly, such axiom of paramount importance known as abstraction [Stoll, 1960] or separation [Kuratovski K, Mostowski A.б 1967] principle was out of consideration. As is well known this

principle establishes the correspondence between classical subsets and the properties of the elements of the universal set – namely, predicates on the universal set. In the Theory of Fuzzy sets this axiom is out of consideration by the reference to the notice that only subsets is under consideration. And nevertheless having the notation $\mu_{\underline{A}}(e)$ with reference to $\underline{A}$ as to set (here fuzzy) it means nothing because denote fuzzy subset as the graphic of membership function [Kaufmann, 1982]:

$$\underline{A} = \{(e, \mu_{\underline{A}}(e)) : e \in E)\}, \; 0 \leq \mu_{\underline{A}} \leq 1.$$

This gap with the absence of an analogue of separation cause the question about what is just the object or property characterized uncertainly by the fuzzy subset (E, $\mu_{\underline{A}}(e)$).

Implicitly and only partially the problem of the gap between subsets and properties are bringing about in the conception of the linguistic variables and its values.

It is just within the framework of linguistic variables the separation principle is realizing implicitly, i.e. the equivalency of the properties and subsets is adopted. Such latent realization discover itself by consideration for this or that value: for example for the "distance" – defined properties, for example: "long", "middle", "short". This properties correspond with certain interval of the "distance, and these intervals are just the supporters for defining the appropriate membership functions. Each of these membership functions namely realizes the uncertain description of the correspondent property. Thus the linguistic variable is not simply a fuzzy set or collection of such sets. It is fuzzy set plus property described implicitly by this variable. In such a way linguistic variable embodies tacitly the idea of the object for uncertainty. In the example with a "distance" these objects of uncertainty are the properties "long", "middle", "short". Besides, linguistic variable establishes the relations between fuzzy sets and correspondent properties.

Such approach: approach by the mean of linguistic variable is realizing the relation between property and fuzzy sets only partially, because all the supporters of these sets are different from each others. In the example with a "distance" the supporters of the correspondent fuzzy sets are the subintervals of the general interval for the "distance".

In the author's publications [Donchenko, 1998, №3], [Donchenko, 1998, №3] probabilistic interpretation for fuzzy sets is proposed: the probabilistic interpretation for membership function to be more explicit.

In the publication [Donchenko, 2004] the more precise definition for fuzzy set is given. In that one appropriate predicate P on the supporter is introduced in the definition by itself under retaining the general approach to definition of fuzzy sets. A membership function is dependent of the supporter's element and the predicate P within the notation $\mu^{\{P\}}(e)$.

Generalized Logit- and Probit – regressions exemplify such approach. In the ones membership function is associated with certain property – event: purchasing of that or this goods for example. An element from the supporter (appropriate R^p) describes additional conditions: income and so on. The property – event is inhered originally. It is fixed by event $\{Y = 1\}$, correspondent to purchasing or something of that type, with some Bernoulli's random variable Y.

1. Modified Determination of Fuzzy Sets

The way for the solving the problem of constructing the analogue of the separation principle may be on the author opinion the straight reference on the object or property described uncertainly. This reference ought to be reflected evidently in the definition of the membership function: $\mu^{\{P\}}(e)$, where P – correspondent property (predicate) on E. Taking into account that for classical sets separation principle take place the property may be replaced by appropriate subset(classical surely). Two membership functions $\mu^{\{P_1\}}(e)$ and $\mu^{\{P_2\}}(e)$ c $P_1 \neq P_2$, specify two different Fuzzy sets, even if they are equal as the function of e, $e \in E$.

Definition. By the Fuzzy subset (modified) of the E, which uncertainly describe P on E (or subset $P_E \subseteq E$), will be called the pair (E, $\mu^{(P)}(e)$) or (E, $\mu^{(P_E)}(e)$), where:

- E – is the abstract set, which will be referenced to as universal or supporter;
- P – predicate on E, P_E – subset of E, which corresponds to P;
- $\mu^{(P)}(e)$ – function of two arguments: e, e∈E and P from the set of all predicates on E. This function just as in classical theory of Fuzzy sets will be referenced to as membership function, adding that it characterize uncertainly property P(or P_E) .

2. Probabilistic Interpretation for Fuzzy Sets

Generally speaking, the authors – founders of the Fuzzy set Theory time and again underlined distinction and fundamental alternatively distinction relatively Statistics.

The thesis that the Fuzzy set theory is alternative to statistical methods underlined emphatically. The opinion that the membership function reflects only the degree of the expert subjective confidence became commonplace. But the object of the characterization stayed out of Fuzzy sets definition.

This example of logit- and probit– likely regressions mentioned above is not universally kind example because special character of the supporter.

Nevertheless the practices of mathematical modeling using Fuzzy sets demand emphatically such interpretation. The one ought to bring out Fuzzy sets out subjective confidence that or those experts and put at the researchers disposal possibilities to establish the correspondence between fuzzy set and the correspondent object under the modeling. It ought to answer the question also what is the observation of Fuzzy set.

2.1. Probabilistic Interpretation for Fuzzy Sets: Discrete Supporter

This subsection deals with the probabilistic interpretation for the classical variant of the Fuzzy Set: Fuzzy Set with discrete supporter – discrete E. This interpretation keeps true for modified variant also.

The interpretation namely is the consequence of the next theorem.

Theorem 1. For any classical Fuzzy Set (E, $\mu_{\underline{A}}(e)$) with discrete support E exist such discrete probability space (Ω, B_Ω, P), event $A \in B_\Omega$ and complete collection of events H_e: $H_e \in B_\Omega$, e∈E, – within this probability space such that membership function $\mu_{\underline{A}}(e)$ is represented by the system of conditional probabilities in the next form:

$$\mu_{\underline{A}}(e) = P\{A|H_e\},\ e \in E .$$

Proof. Let's choose and fix any two-element set with the elements say α and $\overline{\alpha}$. Let's also construct the Ω in the next way:

$$\Omega = \{\alpha, \overline{\alpha}\} \times E .$$

Any element $\omega \in \Omega$ is the pair (α ,e) or ω=($\overline{\alpha}$,e) for appropriate $e \in E$. As the E is discrete Ω is also discrete. Let p_e, $p_e > 0$, e∈E – be the probabilities of any distribution row with the only constraint: all of them are positive. One may determine on the Boolean of the Ω probability P by the distribution row $\overline{p}_\omega, \omega \in \Omega$ in the next way:

$$\overline{p}_\omega = \begin{cases} \mu_{\underline{A}}(e)p_e, \text{ для } \omega = (e,\alpha) \\ (1-\mu_{\underline{A}}(e))p_e, \text{ для } \omega = (e,\overline{\alpha}) \end{cases}$$

Indeed:

- For any $\omega \in \Omega$ $\overline{p}_\omega \geq 0$;
- $\sum_{\omega\in\Omega} \overline{p}_\omega = \sum_{e\in E} p_{(\alpha,e)} + \sum_{e\in E} p_{(\overline{\alpha},e)} = \sum_{e\in E} \mu_{\underline{A}}(e)p_e + \sum_{e\in E}(1-\mu_{\underline{A}}(e))p_e = 1$

We also determine the event $A \in B_\Omega$ and complete collection of the events $H_e \in B_\Omega$, e∈E by the next relations correspondingly:

$$A = \{\alpha\} \times E$$

$$H_e = \{(\alpha, e), (\bar{\alpha}, e\}, e \in E.$$

It is obviously, that $A \cap H_e = \{(\alpha, e)\}$.
Besides,

$$P\{H_e\} = P\{(\alpha, e) + P\{(\bar{\alpha}, e) = p_e > 0,$$

$$P(A \cap H_e) = P\{(\alpha, e)\} = \mu_{\underline{A}}(e) \cdot p_e.$$

Also:

$$P(A \mid H_e) = \frac{P(A \cap H_e)}{P(H_e)} = \frac{P\{(\alpha, e)\}}{p_e} = \frac{\mu_{\underline{A}}(e) p_e}{p_e} = \mu_{\underline{A}}(e).$$

Then we get $\mu_{\underline{A}}(e) = P(A \mid H_e)$, and the proof finishes.

The result of the Th1 may be extended on the special collection of the Fuzzy Sets which we will designate as complete in the sense of the next definition.

Definittion. A collection $(E, \mu_{\underline{A_i}}(e))$, $i = \overline{1,n}$ of the classical subset will be called complete if the next relation take place for any $e \in E$:

$$\sum_{i=1}^{n} \mu_{\underline{A_i}}(e) = 1.$$

Theorem 2. For any complete collection of Fuzzy subsets $(E, \mu_{\underline{A_i}}(e))$, $i = \overline{1,n}$ with the equal supporter E one may find discrete probability space (Ω, B_Ω, P), collection of the evens $A_i \in B_\Omega$, $i = \overline{1,n}$ and complete collection of the events H_e: $H_e \in B_\Omega$, e∈E, – within this probability space, that all of the membership functions $\mu_{\underline{A_i}}$, $i = \overline{1,n}$ may be represented as the systems of conditional probabilities in the next way:

$$\mu_{\underline{A_i}}(e) = P\{A_i \mid H_e\}, \text{ for any } e \in E, \ i = \overline{1,n}.$$

Proof. Just as in previous case let's choose and fix set $\Re$ but with n elements now: $\Re = \{\alpha_1, \ldots, \alpha_n\}$ and construct the Ω as the next Cartesian product $\Omega = \Re \times E$. An element $\omega \in \Omega$ is the pair of the type $\omega = (\alpha, e)$ for appropriate $e \in E$, $\alpha \in \Re$. The set Ω is discrete again. Let's p_e, $p_e > 0$, e∈E are chosen in just the same way as in the proof of the Th1, i.e. are the probabilities of the discrete distribution row on E. The probability P on the Boolean of Ω we construct by the membership function $\mu_{\underline{A_i}}(e)$, $i = \overline{1,n}$ and probabilities p_e, e∈E describing the distribution row $\bar{p}_\omega, \omega \in \Omega$ in the next way now:

$$\bar{p}_\omega = \begin{cases} \mu_{\underline{A_1}}(e) p_e, \text{ for } \omega = (e, \alpha_1) \\ \cdots \\ \mu_{\underline{A_n}}(e)) p_e, \text{ for } \omega = (e, \alpha_n) \end{cases}$$

Indeed:

- For any $\omega \in \Omega$ $\bar{p}_\omega \geq 0$;
- $\sum_{\omega \in \Omega} \bar{p}_\omega = \sum_{i=1}^{n} \sum_{e \in E} \bar{p}_{(\alpha_i, e)} = \sum_{i=1}^{n} \sum_{e \in E} \mu_{\underline{A_i}}(e) p_e = \sum_{i=1}^{n} \mu_{\underline{A_i}}(e) \sum_{e \in E} p_e = \sum_{i=1}^{n} \mu_{\underline{A_i}}(e) = 1.$

We determine the events $A_i \in B_\Omega$, $i = \overline{1,n}$ and complete collection of events $H_e \in B_\Omega$, $e \in E$ then by the next relations correspondingly:

$$A_i = \{\alpha_i\} \times E, i = \overline{1,n},$$
$$H_e = \Re \times \{e\}, e \in E.$$

Once again

$$A_i \cap H_e = \{(\alpha_i, e)\}, \ i = \overline{1,n}, \ e \in E.$$

and

$$P\{H_e\} = \sum_{\omega \in H_e} \bar{p}_\omega = \sum_{i=1}^{n} \bar{p}_{(\alpha_i, e)} = \sum_{i=1}^{n} \mu_{\underline{A_i}}(e) p_e = p_e > 0,$$
$$P(A_i \cap H_e) = P\{(\alpha_i, e)\} = \mu_{\underline{A_i}}(e) \cdot p_e.$$

Thus

$$P(A_i \mid H_e) = \frac{P(A_i \cap H_e)}{P(H_e)} = \frac{P\{(\alpha_i, e)\}}{p_e} = \frac{\mu_{\underline{A_i}}(e) p_e}{p_e} = \mu_{\underline{A_i}}(e).$$

And so

$$\mu_{\underline{A_i}}(e) = P(A_i \mid H_e), i = \overline{1,n}, \ e \in E.$$

And this is the end of the proof.

2.2. Probabilistic Interpretation for Fuzzy Sets : Non - Discrete Supporter

The result of the previous subsection may be improved noticeably and to extend to non-discrete case if the supporter E possesses certain structure, namely is the space with a measure.

Theorem 3. Given the:

- (E, $\Im$, m)- is the space with a measure;
- $(E, \mu_{\underline{A_i}}(e))$, $i = \overline{1,n}$ μ_i(e), i > 0 is the complete collection on Fuzzy subsets with the equal supporters E;
- all of the membership functions $\mu^{(A_i)}(e)$, $i = \overline{1,n}$ are $\Im$, £, measurable (£– Borel σ -algebra on R^1),

then:

- exist probability space (Ω, B_Ω, P),
- exist ξ discrete random S_p – valued random variable on (Ω, B_Ω, P) with S_p is n-element set with elemens say S_i, $i = \overline{1,n}$;
- exist η random E – valued random variable on (Ω, B_Ω, P)

such, that for any $i = \overline{1,n}$

$\mu^{(A_i)}(e) = P\{\xi = S_i \mid \eta = e\}$,

where $P\{\xi = S_i \mid \eta\}$ – conditional distribution of r.v. ξ respectively r.v.. η.

The conditional distribution is regular:for any $e \in E$ P(B | η= e } is a probability respectively B.

Proof. The proof the result extend the idea of previous states of the kind and being technically valuable is ommitted.

2.3. Observations of the Modified Fuzzy Sets

The modification of the definition of Fuzzy set introduced earlier in the paper imparts the objectivity to the Fuzzy set and it is possible now to say about observations of Fuzzy sets for modified ones. It's very important ontologic aspect for mathematical modeling using Fuzzy sets. The observation of modified Fuzzy sets is the pair (e, P(e)) – e, $e \in E$ – element from the supporter and P(e) is the predicate P( ) mean on this element. Namely e is the displayed element and P(e) is the fixed information about property P( ). It is just in such a way the observations are interpreting in the logit- and probit – regressions and in its generalizations.

The membership function may be estimated statistically by MLM for example as it is used to be in the regressions mentioned above. And experts may be used to estimate the function too but in new way additionally. Namely, they may be questioned about $\mu^{(A_i)}(e)$ as it is now, but about fulfilling P(e) also. So the combination of LSM and MLM may be used for the membership function to estimate.

Conclusion

The statistical interpretation and modification of the Fuzzy sets conception represented in the paper earlier make it possible to use the Fuzzy set characterization as the one existent objectively. This approach solves the question of the analogue of separation principle for the classical Fuzzy sets when these ones are considered in modified way.

Bibliography

[Zadeh, 1965] Zadeh, Lotfi. Fuzzy Sets/ Information and Control, 8(3). June 1965. pp. 338-53.

[Kaufmann, 1982], Kaufmann Arnold, Introduction in the Theory of Fuzzy Sets, - Moscow. 1982 - 322 p. (in Russian)

[Stoll, 1960] Stoll, Robert R, Sets,.Logic.and Axiomatic Theories. Freeman and Company, San Francisco,1960.

[Kuratovski K, Mostowski A.б 1967] Kuratovski K, Mostowski A. Set theory – North Holland Publishing Company, Amsterdam, 1967.

[Donchenko, 1998, №3] Donchenko V.S. Conditional distributions and Fuzzy sets. // Bulletin of Kiev University. Series: Physics and Mathematics, №3, 1998. (In Ukrainian)

[Donchenko, 1998, №4] Donchenko V.S. Probability and Fuzzy sets. // Bulletin of Kiev University. Series: Physics and Mathematics, №4,1998. (In Ukrainian)

[Donchenko, 2004] Donchenko V.S. Statistical models of observations and Fuzzy sets. // Вестник Киевского университета, №1, 2004.

Authors' Information

Volodymyr S. Donchenko – Kyiv National Taras Shevchenko Universiry, Professor. 2, Akademician Glushkov prospectus, building 6, Kyiv, 03680, Ukraine; e-mail: vsdon@unictb.kiev.ua

DEVELOPING AGENT INTERACTION PROTOCOLS WITH PRALU

Dmitry Cheremisinov, Liudmila Cheremisinova

Abstract. The purpose of the paper is to explore the possibility of applying existing formal theories of description and design of distributed and concurrent systems to interaction protocols for real-time multi-agent systems. In particular it is shown how the language PRALU, proposed for description of parallel logical control algorithms and rooted in the Petri net formalism, can be used for the modeling of complex concurrent conversations between agents in a multi-agent system. It is demonstrated with a known example of English auction on how to specify an agent interaction protocol using considered means.

Keywords: multi-agent system, interaction protocol, parallel control algorithm

ACM Classification Keywords: I.2.11 [Computer Applications]; Distributed Artificial Intelligence, Multiagent systems; D.3.3 [Programming Languages]: Language Constructs and Features – Control structures, Concurrent programming structures

Introduction

Agents are becoming one of the most important topics in distributed and autonomous decentralized systems, and there are increasing attempts to use agent technologies to develop large-scale commercial and industrial software systems. Over the last decade, the specification, design, verification and application of a particular type of agents, called BDI (belief, desire, intention) agents [1], have received a great deal of attention. BDI agents are systems that are situated in changing environment receive continuous perceptual input and take actions to affect their environment based on their internal state. In particular, there are many efforts aimed at developing agent-oriented designs that are typically structured as multi-agent systems (MASs). MAS is a computational system in which two or more agents interact or work together to perform a set of tasks or to achieve a set of goals [2]. Agents of a MAS interact with others toward their common specific objective or individual benefit. Agent interactions are established through exchanging messages that specify the desired performatives of other agents (such as notice, request) and declarative representations of the content of messages.

MAS is usually specified as a concurrent system based on the notion of autonomous, reactive and internally-motivated agents acting in a decentralized environment. One key reason of the growth of interest in MAS is that the idea of an agent as an autonomous system, capable of interacting with other agents in order to satisfy its design objectives, is a naturally appealing one for software designers. Agents are used in an increasingly wide variety of applications such as distributed and autonomous decentralized systems. The complexity of MASs suggests a pressing need for system modeling techniques to support reliable, maintainable and extensible design. Although there are many efforts aimed at developing such MASs, there is sparse research on formal specification and design of such systems.

Agent system can operate if the agents are able to exchange information in the form of messages and if they have a common understanding of the possible types of messages that are connected with message “content”. This shared understanding is referred to as ontology [3]. A great deal of agent-based research has devoted to the development of techniques for representing ontology’s. Multi-agent conversations are built upon two components: agent communication language and interaction protocol. There are a number of agent communication languages, such as Knowledge Query and Manipulation Language (KQML) [4] and the Foundation for Intelligent Physical Agents (FIPA) ACL [5] and others designed for special purposes and that are like mentioned ones. These agent communication languages specify a domain specific vocabulary (ontology) and the individual messages that can be exchanged between agents.

Interaction protocols [5] specify the sequences in which these messages should be arranged in agent interactions. A group of rational agents complies with an interaction protocol in order to engage in task-oriented sequences of message exchange. Thus, when an agent sends a message, it can expect a receiver's response to

be among a set of messages indicated by the protocol and the interaction history. With a common interpretation of the protocol, each member of the group can also use the rules of the interaction in order to satisfy its own goals. In other words protocol constrains the sequences of allowed messages for each agent at any stage during a communicative interaction (dialogue), i.e. it describes some standard pattern messages exchanged between agents need to follow. Protocol plays a central role in agent communication. It specifies the rules of interaction between communicating agents and the first thing should be done by the designer developing any particular real-time system is to impose interaction protocol.

This paper explores the possibility of applying existing formal theories of description of distributed and concurrent systems to interaction protocols for real-time multi-agent systems. In particular it is shown how the language PRALU [6, 7], proposed for description of parallel logical control algorithms and rooted in the Petri net formalism, can be used to describe agent interaction protocols. The described approach can be used for the modeling of complex, concurrent conversations between agents in a multi-agent system. It can be used to define protocols for complex conversations composed of a great number of simpler conversations. With the language PRALU it is possible to express graphically the concurrent characteristics of a conversation, to capture the state of a complex conversation during runtime, and to reuse described conversation structure for processing multiple concurrent messages. It is demonstrated with a known example of English auction [5] on how to specify an agent interaction protocol using considered means. Finally, using PRALU language we can verify some key behavioral properties of our protocol description that is facilitated by the use of existing software for the language PRALU [6, 7, 8].

Agent Interaction Protocols

The application domains of MASs are getting more and more complex. Firstly, many current application domains of MASs require agents to work in changing environment (or world) that acts on or is acted on by the system. A closed system is one that has no environment; it is completely self-contained in contrast to an open (uncertain) system, which interacts with its environment. Any real system is open. The MAS must decide what to do and develop a strategy in order to achieve its assigned goals. For this, the MAS must have a representation or a model of the environment in which it evolves. The environment is composed of situations. A situation is the complete state of the world at an instant of time.

The application of multi-agent systems to real-time environments can provide new solutions to very complex and restrictive systems such as real-time systems. A suitable method for real-time multi-agent system development must take into account the intrinsic characteristics of systems of this type. As a rule they are distributed, concurrent systems with adaptive and intelligent behaviour. For agent-based systems to operate effectively, they must understand messages that have a common ontology underlying them. Understanding messages that refer to ontology can require a considerable amount of reasoning on the part of the agents, and this can affect system performance.

BDI agents are systems that are situated in a changing environment, receive continuous perceptual input, and take actions to affect their environment, all based on their internal mental state. Within the BDI architecture agents are associated with beliefs (typically about the environment and other agents), desires or goals to achieve, and intentions or plans to act upon to achieve its desires. In practical terms, beliefs represent the information an agent has about the state of the environment. It is updated appropriately after each action. The desires denote the objectives to be accomplished, including what priorities are associated with the various objectives. Intentions reflect the actions that must be fulfilled to achieve the goal (the rules to be fired).

To reduce the search space of possible responses to agent messages, interaction protocols can be employed. They specify a limited range of responses that are appropriate to specific message types for a given protocol. When an agent is involved in a conversation that uses an interaction protocol, it maintains a representation of the protocol that keeps track of the current state of the conversation. After a message is received or sent, it updates the state of the conversation in this representation.

By the very nature of protocols as public conventions, it is desirable to use a formal language to represent them. When agents are involved in interactions where no concurrency is allowed, conversation protocols are traditionally specified as deterministic finite automata (DFA) of which there are numerous examples in the literature. DFA consists of a set of states, an input alphabet and a transition function, which maps every pair of state and input to next state. In the context of interaction protocols [9], the transitions specify the communicative

actions to be used by the various agents involved in a conversation. A protocol based on such a DFA representation determines a class of well-formed conversations. Conversations that are defined in this way have a fixed structure that can be laid down using some kind of graphical representation.

Protocols can be represented as well in a variety of other ways. The simplest is a message flow diagram, as used by FIPA [5]. More complex protocols will be better represented using a UML sequence (Uniffied Modelling Language) [10] and AUML [11], interaction diagram, statechart [12] and Colored Petri Net (CPN) [13]. UML is one of the currently most popular graphical design languages that are de facto standard for the description of software systems. AUML extends UML sequence diagrams to represent asynchronous exchange of messages between agents. The advantage of AUML is its visual representation. Statecharts [12] are an extension of conventional DFAs. However, expressing protocols of realistic complexity using statecharts or AUML requires substantial efforts for developing, debugging and understanding.

A CPN model of a system describes the states, which the system may be in, and the transitions between these states. CPNs provide an appropriate mathematical formalism for the description, construction and analysis of distributed and concurrent systems. CPNs can express a great range of interactions in graphical representations and well-defined semantics, and allow formal analysis and transformations [14]. By using CPNs, an agent interaction protocol can be modeled as a net of components, which are carriers of the protocol structure. Using CPNs to model agent interaction protocol, the states of an agent interaction are represented by CPN places. Each place has an associated type determining the kind of data that the place may contain. Data exchanged between agents are represented by tokens, and the colors of tokens indicate the data value of the tokens. The interaction policies of a protocol are carried by CPN transitions and their associated arcs. A transition is enabled if all of its input places have tokens, and the colors of these tokens can satisfy constraints that are specified on the arcs. A transition can be *fire*d, which means the actions of this transition can occur, when this transition is enabled. When a transition occurs, it consumes all the input tokens as computing parameters, conducts conversation policy and adds new tokens into all of its output places. After a transition occurs, the state (marking) of a protocol has been changed and a protocol will be in terminal state when there is no enabled or fired transition.

There are a number of works of using Petri Nets or CPNs to model agent interaction protocols [15, 16], there have been also some works on the investigation of protocols' flexibility, robustness and extensibility [17]. It is becoming consensus [14] that a CPN is one of the best ways to model agent interaction protocols. However the notion of an agent executing an action with Petri Net is not explicit in the notation [18]. Different PNs can be assigned to each agent role, raising questions about how the entire protocol is inferred and the reachability and consistency of shared places. If a single Petri net is partitioned for each role, this leads to a complex diagram where a partition is required for each agent identified. Furthermore, alternative actions and states such as either agree or reject but not both, cannot be expressed in standard Petri nets.

Keeping in mind complex systems characterized by complex interaction, asynchronism and concurrency we propose to use for the purposes of their description a special language PRALU [6, 7] having its background in the Petri net theory (expanded nets of free choice – EFC–nets investigated by Hack [19]) but possessing special means for keeping track of the current states of the conversation, receiving messages and initiating responses. The formal language PRALU combines properties of "cause-effect" models with Petri nets. It is intended for a wide application in engineering practice and is well suited for representation of the interactions involved in concurrent system, synchronization among them and then it is simple enough for understanding. The language PRALU supports hierarchical description of the algorithms that is especially important in the case of complex systems. At last, powerful software has been developed that provides correctness verifying, simulation, hardware and software PRALU-description's implementation [7, 8]. The review of obtained results it can be found in [8].

PRALU Language

Any algorithm in PRALU consists of sequences of operations to be executed in some pre-determined order. There exist two basic kinds of operations used in PRALU: acting operations "$\rightarrow A$" and waiting operations "$- p$". Action operation changes the state of the object under control, whereas a waiting operation is passive waiting for some event without affecting anything. In simple case A and p are conjunctive terms, so acting and waiting operations can be interpreted as waiting for event $p = 1$ and producing event $A = 1$. But A can be understood

too as a formulae defining operations to be performed and p as a predicate defining condition to be verified [20]. For example, acting and waiting operations could be specified by the expressions such as

"$-(a > b + c)$" and "$\rightarrow (a := b + c)$".

The sequences consisting of action and waiting operations are considered to be linear algorithms. For instance, the following expression means: wait for p and execute A, execute B, then wait for q and execute C:

$$-p \rightarrow A \rightarrow B -q \rightarrow C$$

In general, a logical control algorithm can be presented as an unordered set of chains α_j in the form

$$\mu_j : - p_j L_j \rightarrow \nu_j ,$$

where L_j is a linear algorithm, μ_j and ν_j denote the initial and the terminal chain labels represented by some subsets of integers from the set $M = \{1, 2, ..., m\}$: $\mu_j, \nu_j \subset M$ and the expression "$\rightarrow \nu_i$" presents the transition operation: to the chains with labels from ν_j.

Chains can be fulfilled both serially and in parallel. The order in which they should be fulfilled is determined by the variable starting set $N_t \subseteq M$ (its initial value $N_0 = \{1\}$ as a rule): a chain $\alpha_j = \mu_j : - p_j L_j \rightarrow \nu_j$ (that was passive) is activated if $\mu_j \subseteq N_t$ and $p_j = 1$. After executing the operations of the algorithm L_j, N_t gets a new value $N_{t+1} = (N_t \setminus \mu_j) \cup \nu_j$. The algorithm can finish when some terminal value of N is reached (one-element as a rule), at which time all chains became passive. But the algorithms can also be cyclic; they are widely used when describing production processes.

When the conditions $\mu_j \subseteq N_t$ and $p_j = 1$ are satisfied for several chains simultaneously these chains will be fulfilled concurrently. On the contrary chains with the same initial labels are alternative (only one of them can be fulfilled at a time), they are united in a sentence with the same label as will be shown below.

Thus PRALU allows concurrent and alternative branching, as well as merging concurrent and converging alternative branches. These possibilities are illustrated with the following examples of simplified fragments [20]:

Concurrent branching	Merging concurrent branching	Alternative branching	Converging alternative branching
1: ...→2.3	2: ...→4	1: – a...→2	2: ...→4
2: ...	3: ...→5	– $\bar{a}$ ...→3	3: ...→4
3: ..	4.5: ...		4: ...

In PRALU there are two syntactic constraints on chains that restrict concurrent and alternative brunching. If some chains are united in the same sentence (they have equal initial labels) they should have orthogonal predicates in the waiting operations opening the chains:

$$(i \neq j) \,\&\, (\mu_i \cap \mu_j \neq \varnothing) \rightarrow (p_i \,\&\, p_j = 0).$$

The other constraint is similar to the corresponding condition specific for extended nets of free choice (Hack [19]):

$$(i \neq j) \,\&\, (\mu_i \cap \mu_j \neq \varnothing) \rightarrow (\mu_i = \mu_j).$$

PRALU language has some more useful properties that come in handy for description of complex interaction protocols.

1. PRALU algorithms can be expressed both in graphical and symbolic forms.

2. PRALU language permits hierarchical descriptions. The two terminal algorithms (having the only terminal label) may be used as blocks (invoked as complex acting operations) in hierarchical algorithms.

3. In PRALU there exist some additional interesting operations that can be useful for description of interaction protocols. Those are suppression operations and some arithmetic operations. The first ones provide response on special events that can take place outside or within control system. Suppression operation ("$\rightarrow *$", "$\rightarrow * \gamma$", "$\rightarrow ' \gamma$") interrupts the execution all concurrently executed chains of the algorithm or those ones mentioned in γ, (or do not mentioned in γ – in the case of "$\rightarrow ' \gamma$"). This operation breaks the normal algorithm flow. Among arithmetic

operations it ought to be mention timeout operations (waiting for *n* unit times "– *n*") and counting operation that counts event occurrences.

Specifying Protocols in PRALU

For an example of an interaction protocol, consider an English auction [5]. The auctioneer seeks to find the market price of a good by initially proposing a price below that of the supposed market value and then gradually raising the price. Each time the price is announced, the auctioneer waits to see if any buyers will signal their willingness to pay the proposed price. As soon as one buyer indicates that it will accept the price, the auctioneer issues a new call for bids with an incremented price. The auction continues until no buyers are prepared to pay the proposed price, at which point the auction ends. If the last price that was accepted by a buyer exceeds the auctioneer's (privately known) reservation price, the good is sold to that buyer for the agreed price. If the last accepted price is less than the reservation price, the good is not sold.

In the case of the auction there are participants of two types: the initiator of the conversation – Auctioneer, and others – Buyers. So, we have two kinds of interaction protocols – those of Auctioneer and of Buyers. The last participants are peer and should be described with identical interaction protocols modeled in our case by the operation "Buyer".

Interaction protocol as a whole can be represented in PRALU as three complex acting operations – blocks. PRALU-blocks are exchanging with values of logical (binary) variables, only such variables are mentioned in them. Each block has some sets of input and output variables that are enumerated in brackets following the block name (the other variables of a block are its internal). Initialization of a complex acting operation is depicted by the fragment such as "$\rightarrow *$Buyer". The operation Buyer exists in as many copies as the number of participants of the auction, so the copies of the operation differ in their indexes only.

The modeling of the process of auction begins with the execution of "Main_process" triggering event that initiates the interaction protocol execution. Here the processes Auctioneer and $Buyer_n$s are executed concurrently. For the sake of simplicity we limit the number of buyers to two (in principle it can be simply increased). The process Auctioneer starts with sending the first message (start_auction) that is waited by others participants to continue communication.

Below PRALU description of the auction interaction protocol is shown. Here we cite the only block $Buyer_n$, but for real application (intending to simulate the process of auction, for example) we should have as many proper copies as it has been used (in our case – two). Fig. 1 depicts graphical schemes of three mentioned PRALU-blocks: Main_process, Auctioneer and $Buyer_n$.

Main_process ()

1: $\rightarrow$2.3.4
2: $\rightarrow *$Auctioneer $\rightarrow$5
3: $\rightarrow *Buyer_1 \rightarrow$6
4: $\rightarrow *Buyer_2 \rightarrow$7
5.6.7: $\rightarrow$.

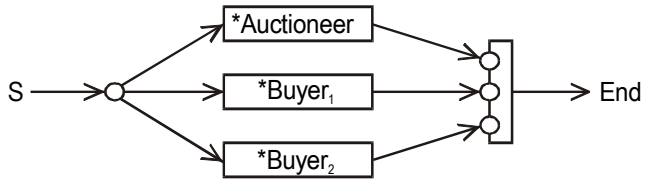
$Buyer_n$ (start_auction, price_proposed, end_auction / $accept\_price_n$, not_understand)

1: –start_auction $\rightarrow$2
2: –price_proposed $\rightarrow *$Decide(/decision_accept,decision_reject) $\rightarrow$3
–end_auction $\rightarrow$.
3: –decision_accept $\rightarrow$ $accept_price_n \rightarrow$4
–decision_reject $\rightarrow$′$accept_price_n \rightarrow$4
–not_understand $\rightarrow$4
4: –timeout $\rightarrow$2

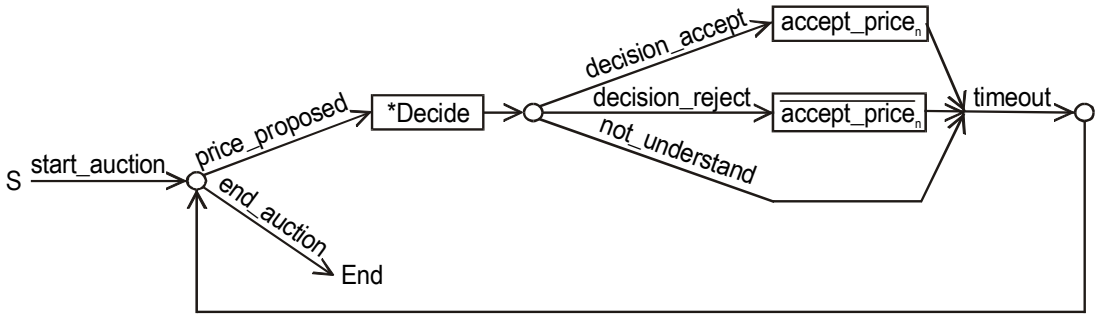
Auctioneer ($accept\_price_1$, $accept\_price_2$, not_understand / start_auction, price_proposed, end_auction)

1: $\rightarrow$start_auction $\rightarrow$2
2: $\rightarrow$′$accept\_price_1$.′$accept\_price_1$.′not_understand $\rightarrow *$Price_propose(/price_proposed) $\rightarrow$3
3: –not_understand $\rightarrow$2
– $accept_price_1 \rightarrow$4

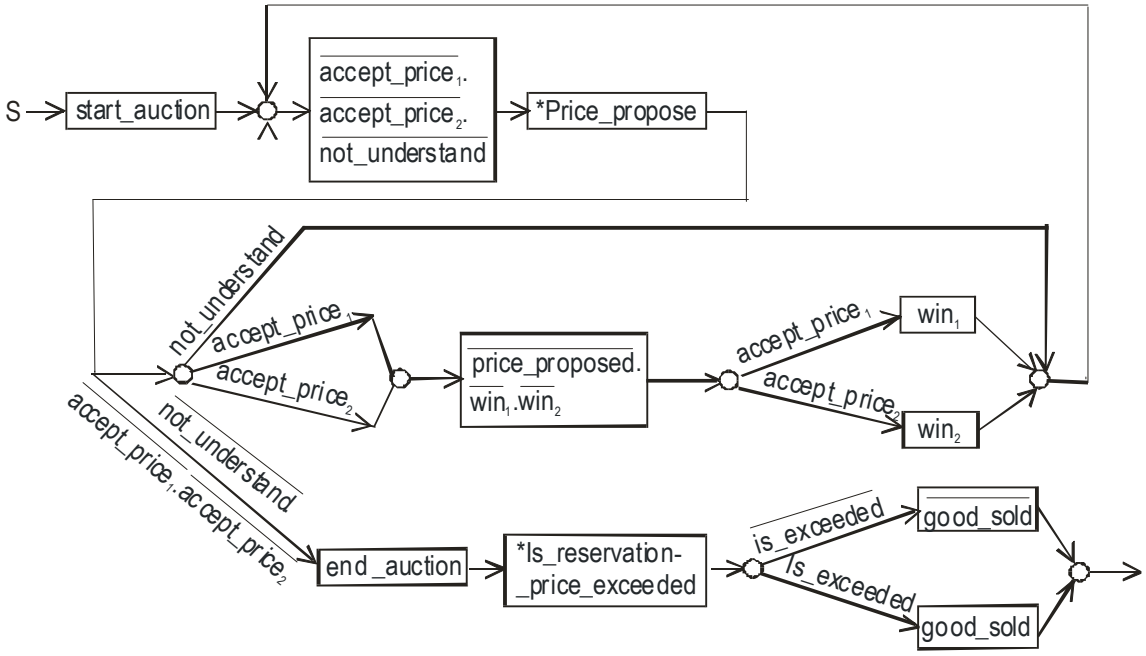
– accept_ price$_2$ →4
–′not_understand.′accept_ price$_1$.′accept_ price$_2$ →end_auction
→∗Is_reservation_price_exceeded(/is_exceeded) →6
4: →′price_proposed.′win$_1$.′win$_2$ →5
5: – accept_ price$_1$ →win1 →2
– accept_ price$_2$ →win2 →2
6: –is_exceeded →good_sold →7
–′ is_exceeded →′good_sold →7
7: →.



Main_process ()



Buyer$_n$ (start_auction, price_proposed, end_auction / accept_price$_n$, not_understand)



Auctioneer (accept_price$_1$, accept_price$_2$, not_understand / start_auction, price_proposed, end_auction)

Fig. 1. English auction interaction protocol in PRALU

It is assumed that all unformalized operations are referred to as acting operations that set values of logical variables concerned with them. For example, Buyer's operation "Decide" decides for accepting or rejecting the announced price. Depending on adopted decision, it outputs true value of logical variable "decision_accept" or "decision_reject". In a similar, Auctioneers operation "Price_propose" proposes an initial price or increments the charged price outputting true value of logical variable "price_proposed"; the operation "Is_reservation_price_exceeded" verifies if the price accepted by a buyer exceeds the auctioneer's reservation price outputting true or false value of logical variable "is_exceeded".

The operation "–timeout" (where timeout is integer number) means waiting for timeout unit times before doing something followed it. The operation "$\rightarrow$." is interpreted as the transition to an end of a process described by the block. When the processes of Auctioneer and all Buyers reach their end in the Main_process the transition to its end is executed.

Conclusion

This paper has addressed the need for formalized and more expressive logical and graphical methodologies for specifying (and then validating) interaction protocols in multi-agent systems. Towards this, it was proposed to use the formal language PRALU intended for the representation of complex interactions involved in concurrent system, being in need of synchronization among these interactions. It was demonstrated as well how PRALU algorithms could be used for the specification of multi-agent interaction protocols by the example of English auction. In favour of using the language PRALU is the existence of a great deal of methods and software developed for simulation and logical design of PRALU algorithms as well as for their hardware and software implementation.

References

1. B. Burmeister and K. Sundermeyer, "Cooperative problem-solving guided by intensions and perception", edited by E. Werner and Y. Demazean, Decentralized A.I. 3, Amsterdam, The Netherlands, North Holland, 1992.
2. V. Lesser, "Cooperative Multiagent Systems: A Personal View of the State of the Art", IEEE Trans. Knowledge and Data Engineering, vol. 11, no 1, pp. 133–142, 1999.
3. T. R. Gruber, "A Translation Approach to Portable Ontologies", Knowledge Acquisition, 1993, vol. 5, no 2, pp. 199-220, 1993
4. Finin, Y. Labrou and J. Mayfield, "KQML as an agent communication language", edited by J.M. Bradshaw, Software Agents, MIT Press, pp. 291–316, 1997.
4. ARPA Knowledge Sharing Initiative. Specification of the KQML agent-communication language. ARPA Knowledge Sharing Initiative, External Interfaces Working Group, July 1993.
5. Foundation for Intelligent Physical Agents (FIPA), http://www.fipa.org.
6. A.D. Zakrevskij, V.K. Vasiljonok, "The formal description of logical control algorithms for designing discrete systems", Electronnoje modelirovanie, vol. 6, no 4, pp. 79–84, 1984 (in Russian).
7. A.D. Zakrevskij, "Parallel algorithms for logical control", Minsk, Institute of Engineering Cybernetics of NAS of Belarus, 202 p., 1999 (in Russian).
8. L.D. Cheremisinova, "Realization of parallel algorithms for logical control", Minsk, Institute of Engineering Cybernetics of NAS of Belarus, 246 p., 2002 (in Russian).
9. M. Greaves, and J. Bradshaw J., "Specifying and Implementing Conversation Policies", Autonomous Agents '99 Workshop, Seattle, WA, May 1999.
10. G. Booch, J. Rumbaugh, and I. Jacobson, "*The Unified Modeling Language User Guide*", Addison Wesley, 1999.
11. B. Bauer, J.P. Müller, J. Odell, "Agent UML: A Formalism for Specifying Multiagent Interaction," *Agent-Oriented Software Engineering* , edited by P. Ciancarini and M. Wooldridge, Springer-Verlag, Berlin, pp. 91–103, 2001.
12. D. Harel and M. Politi, "Modeling reactive systems with statecharts", McGraw-Hill, 1998.
13. K. Jensen, "Colored Petri Nets – Basic Concepts, Analysis Methods and Practical Use", vol. 1: Basic Concepts, Springer-Verlag, Berlin, 1992.

14. Quan Bai, Minjie Zhang and Khin Than Win, “A Colored Petri Net Based Approach for Multi-agent Interactions”, 2nd International Conference on Autonomous Robots and Agents, Palmerston North, New Zealand, December 13-15, pp. 152–157, 2004
15. R. Cost, “Modelling Agent Conversations with Coloured Petri Nets”, Working Notes of the Workshop on Specifying and Implementing Conversation Policies, Seattle,Washington, pp. 59–66, 1999.
16. D. Poutakidis, L. Padgham, and M. Winikoff, “Debugging Multi-Agent Systems Using Design Artefacts: The Case of Interaction Protocols”, Proceedings of the 1st International Joint Conference on Autonomous Agents and Multi Agent Systems, Bologna, Italy, pp. 960–967, 2002.
17. J. Hutchison and M. Winikoff, “Flexibility and Robustness in Agent Interaction Protocols”, Proceedings of the 1st International Workshop on Challenges in Open Agent Systems, Bologna, Italy, 2002.
18. S. Paurobally, J. Cunningham, “Achieving Common Interaction Protocols in Open Agent Environments”, *AAMAS* ’02, Melbourne, Australia, 2002.
19. M. Hack, ”Analysis of production schemata by Petri nets”, Project MAC-94, Cambridge, 1972.
20. A. Zakrevskij, V. Sklyarov, “The Specification and Design of Parallel Logical Control Devices”, Proceedings of PDPTA'2000, June, Las Vegas, USA, pp. 1635–1641, 2000.

Authors' Information

Dmitry Cheremisinov, Liudmila Cheremisinova – The United Institute of Informatics Problems of National Academy of Sciences of Belarus, Surganov str., 6, Minsk, 220012, Belarus, Tel.: (10-375-17) 284-20-76, e-mail: cher@newman.bas-net.by, cld@newman.bas-net.by

WEBCOMPUTING SERVICE FRAMEWORK

Evgenija Popova

*Abstract: Presented is webComputing – a general framework of mathematically oriented services including remote access to hardware and software resources for mathematical computations, and web interface to dynamic interactive computations and visualization in a diversity of contexts: mathematical research and engineering, computer-aided mathematical/technical education and distance learning. webComputing builds on the innovative web*Mathematica *technology connecting technical computing system* Mathematica *to a web server and providing tools for building dynamic and interactive web-interface to* Mathematica*-based functionality. Discussed are the conception and some of the major components of webComputing service: Scientific Visualization, Domain-Specific Computations, Interactive Education, and Authoring of Interactive Pages.*

Keywords: web-access, mathematical user-interfaces, web computations, mathematical active learning.

ACM Classification Keywords: F.1.2 Modes of Computation: interactive computation, online computation; G.4 Mathematical Software: user interfaces; K.3.1 Computer Use in Education: distance learning.

Introduction

Internet and the World-wide Web make many kind of information and services easily accessible. However many Internet/Web technologies, so powerful in many areas, are not well suited to scientific computation; it is simply not their main focus. The importance of technical/mathematical communication on the Internet is underscored by the activities at the W3 consortium. Internet Accessible Mathematical Computations is part of these activities directed

towards making mathematical computation or information accessible on the Web/Internet [3], [11]. The following topics are subject of investigation:

- Remote access to mathematical software over the Internet.
- Encoding of mathematical expressions (including text-based encodings, for E-mail and HTML embedding, and binary-based encodings for efficient communication between scientific applications).
- Interoperability between software that create/transform/display mathematical expressions (e.g. symbolic, numeric, graphics, text-processing packages) via ad hoc communication protocols & software architectures.
- Web-based mathematics education.
- Access and interoperability to mathematical knowledge bases.
- Protocols, APIs, URL schemes, metadata, and other mechanisms for system interoperability and standardization.
- Application of IAMC for practical purposes such as scientific publishing and archiving, distributed problem solving, etc.

Accessing a scientific computing service should be as simple as entering a command, accessing a Web page, or sending email. A good summary of Web tools for interactive computation is given in [2].

We call a session of mathematical computations *accessible through a Web-interface* if it is triggered by few mouse-clicks while browsing and reading related documents. The interactions happening within the session are related tightly to the actions the user is performing. One can use the computations and their visible results to understand, conjecture, or verify mathematical facts. Within learning environments such connections can be used for *active learning*, that is called also “exploration by doing”.

webComputing is a general framework of mathematically oriented services. These include remote access to hardware and software resources for mathematical computations, and web interface to dynamic interactive computations and visualization in a diversity of contexts: mathematical research and engineering, computer-aided mathematical/technical education and distance learning. webComputing was initiated by a Swiss NSF project [10] and further developed within the frames of the Bulgarian National Science Program “Information Society” under the grant IO-03/2003 “Web-Based Computations and Visualization”. webComputing services are supplied by the COSE server (cose.math.bas.bg); COSE comes from COmputational Science and Engineering. Presently, COSE server is a PC with Athlon64 3000+ CPU having 2GB memory but we hope that in a near future COSE will become part of a computational grid.

In this paper the conception and some of the major components of webComputing service are presented. We begin by overviewing the goals of webComputing. Then the underlying technology for developing and deploying dynamic and interactive web accessible computations and visualization is described shortly. Two major components of webComputing service are discussed in some detail. These are: Scientific Visualization – a service for dynamic and interactive generation of functional graphics, and visualization of data; and webComputing Authoring Framework (wCAF) which allows easy and remotely development of web interface to particular mathematical applications.

Goals of webComputing

webComputing aims to supply a wide range of mathematically oriented computing services over the Internet. Such services include:

- *From the end-user perspective*: accessing remote mathematical computations including access to both general purpose computations and highly specialized computing services; easy and intuitive web access to specialized calculators, general purpose and domain specific computations and visualizations; interactive educational web applications (online training, courseware, exercises, quizzes, etc.) that make a good use of dynamic access to mathematical computing. In these interactive web pages users select different input

parameters and submit data to build up a sequence of results. There is no need to buy install and maintain software; users always have the most recent version; training time is considerably reduced.

- *From the developers perspective*: supporting interactive use of a remote compute server (COSE server where there are installed some scientific computing libraries [4], [5], [6] and *Mathematica* [15] as a computer algebra environment.); providing assistance and promoting research and developments on interoperability between different mathematical systems, environments and data bases; providing resources for easy and remote authoring of dynamic and interactive math oriented pages; hosting application sites using web*Mathematica* technology [13], [14].
- *From a general application perspective*: webComputing is intended to serve the areas of
  - mathematical research and engineering
  - computer-aided mathematical/technical education and distance learning.

We aim to provide a maximum of flexibility for ourselves, and others, in the context of continued development of Internet-based mathematical services.

Underlying Technology

Many Internet accessible mathematical systems have been developed so far. For an overview and general characteristics of these systems see [8]. The following are essential parts of such systems:

- a mathematical system to perform computations
- a user-interface to display the results of computations and to receive the input of the user
- a connection mechanism between the two.

webComputing services are based on the computational and visualization power provided by the technical computing system *Mathematica* [15]. *Mathematica* is a suitable development environment for working on code that models some physical process – code that can then be placed into a dynamic interactive site to enable people to run the model and use its results for their regular work. The system allows building technical computing web services, including numerical, symbolic, and graphical applications that solve daily technical computing problems quickly and easily. In addition, *Mathematica* connects readily to external services, which may be provided by languages such as Java, C, Fortran, Perl.

The connection mechanism and technology empowering webComputing is built upon web*Mathematica*, an innovative new product of Wolfram Research that allows *Mathematica* to run on a server to provide the necessary calculations and graphs [13], [14]. web*Mathematica* is an innovative technology that solves the problem of how to create and distribute solutions to technical computing problems quickly in today's networked environment. It is a server-based technology built on top of two standard Java technologies: Java Servlet and JavaServer Pages (JSP) technologies. The minimum technical components for web*Mathematica* are:

- A Servlet container supporting both the Servlet Specification 2.2 (or higher) and JSP Specification 1.2;
- A Java Development Kit (JDK) 1.2 (or higher), Java 2 Version 1.4 (or higher) is recommended.

web*Mathematica* provides a collection of tools that allow *Mathematica* commands to be placed inside HTML pages. When a request is made for one of these pages, the *Mathematica* commands are evaluated and the computed result is inserted into the page. This can be done by two different HTML templating mechanisms.

Mathematica Server Pages (MSP scripts) are the original form of web*Mathematica* interaction [12]. HTML tags that contain *Mathematica* commands in the classic MSPs are called Mathlets and have the form:

<%Mathlet *Mathematica* commands or web*Mathematica* functions %>

MSP scripting technology is superseded in version 2 of web*Mathematica* [13] by the standard Java templating mechanism, JavaServer Pages (JSP) making use of custom tags. JSPs use a special library of tags called MSP Taglib that work with *Mathematica*. JSPs support the embedding of Java into HTML, and are frequently used along with Java Servlets to develop large dynamic web sites. A web*Mathematica* page uses standard HTML tags as well as special web*Mathematica* tags; these have the form <*msp:tag*>. The web*Mathematica* tags are executed from the top of the page to the bottom. The <`msp:allocate`> tag causes a *Mathematica* kernel to be allocated to use for computations. The contents of the <`msp:evaluate`> tags are sent to *Mathematica*

for computation with the result inserted into the final page. The `</msp:allocate>` tag frees the *Mathematica* kernel to be used for another computation.

Below we present the different stages that are involved in that how web*Mathematica* processes a request [12], see Fig. 1.

1. Make Request. The browser sends an HTTP request to the web server. The request references a particular *Mathematica Server Pages* (MSP) or *Java Server Pages* (JSP) script, includes variables and their values.
2. Forward Request. The web server performs any pre-processing steps, such as authentication, and forwards the request to the web*Mathematica* server (MSP server).
3. Acquire Session. The MSP server acquires a *Mathematica* session for the request from a pool of preinitialized sessions. Any variables and values are sent to this session, which is then instructed to load the script.
4. Process Page. The *Mathematica* session loads the MSP/JSP script and processes any msp tags. *Mathematica* kernel is initialized with input parameters, it carries out calculations, builds then returns the result.
5. Receive Response. The MSP server accepts the response and adds all the necessary HTTP headers for return to the browser. It then clears any temporary settings in the *Mathematica* session and releases the session to the pool of available sessions.
6. Return Result. The web server performs any postprocessing steps and returns the response to the browser.
7. Release Session. The browser accepts an HTML response, which may use applets, plug-ins, or other features of dynamic HTML. Alternatively, the response could be some other format such as MathML, TeX, or a *Mathematica* notebook.

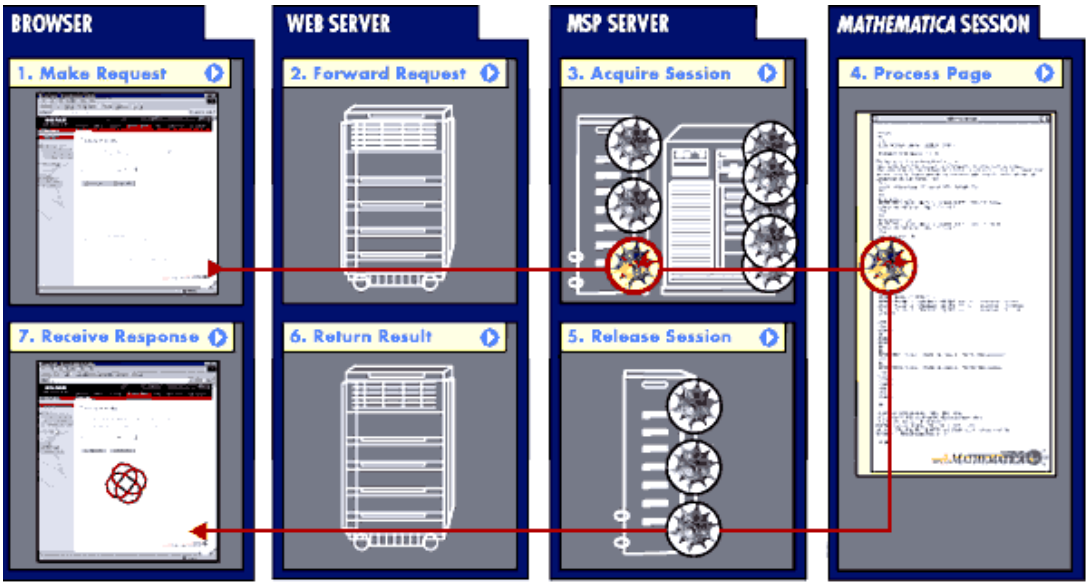


Fig. 1: How web*Mathematica* processes a request.

web*Mathematica* technology uses the request/response standard followed by web servers. Input can come from HTML forms, applets, javascript, and web-enabled applications. It is also possible to send data files to a web*Mathematica* server for processing. Output can be in many different formats such as HTML, images, *Mathematica* notebooks, MathML, SVG, XML, PostScript, and PDF.

An important part of web*Mathematica* is the kernel manager which calls *Mathematica* in a robust, efficient, and secure manner. The manager maintains a pool of one or more *Mathematica* kernels and, in this way, can process more than one request at a time.

webComputing Components

The focus of webComputing framework is on deploying dynamic computations and visualization in diverse contexts. Developed are sites in several domain-specific areas that demonstrate the dynamics in mathematical communication. In these sites users are taken through a sequence of web pages in which they select different input parameters and submit data to build up a sequence of results, Fig. 2. The main webComputing page (http://cose.math.bas.bg/webComputing) contains links to some pages including numerical, symbolic, and graphical applications that solve some technical computing problems quickly and easily.

Step By Step Differentiation page demonstrates how to use active learning strategies in computer-aided education and distance learning. Interactive exercises and activities change the role of the reader from passive to active working subject.

The research work directed towards developing techniques and algorithms for solving problems can now be deployed by live interactive web sites, vastly increasing the number of people who can use and learn from these results. Interval Computations web pages, presently accessible through the main webComputing page, present a typical web application delivering specialized calculations to technical professionals in the following forms: an interval web calculator, dynamic visualization pages, and interval problem solvers.

Teletraffic (http://cose.math.bas.bg/Teletraffic) is a web-based system that performs interactive computations and visualization in the field of teletraffic modeling. It is designed for computation and visualization of parameters of a model which examines the terminal teletraffic in communication systems characterized by (virtual) channels switching, without waiting, with finite number of homogeneous terminals, finite number of switching devices, generalized input flow and repeated calls.

Most of the developed web interfaces are multi-functional – allowing specialized calculations and visualization, and serving for computer-aided distance education involving active elements.

Scientific Visualization

Graphics is an indispensable component in any presentation, research or education material. Scientific Visualization is a web service for interactive graphics generation launched within the frames of webComputing [9]. Scientific Visualization contains a collection of interactive web pages for dynamic generation of 2D and 3D functional plots, visualization of data and some special objects.

Presently, this site provides the following functionality:

- *2D function plots*: two-dimensional graphics of functions of one variable; plots of parametric curves where x, y coordinates are functions of one parameter; plot of one or multiple functions filling the space between each successive pair of curves with a different color; plot of functions with logarithmically scaled x, y, or both coordinates.
- *3D function plots*: three-dimensional graphics of functions in two variables; one or multiple plots of parametric curves where x, y and z coordinates are functions of two parameters; 3D graphics of vector and gradient fields; 3D parametric plots with projections on the coordinate planes.
- *Contour and Density plots* of functions in two variables; plot of curves that are given implicitly as the solutions to equations.
- *2D data plots:* two-dimensional data visualization by points, error bars, labeled points, or in logarithmic scale of x, y, or both coordinates.
- *3D data plots*: visualization of three-dimensional data by density plots or contour lines.
- *Special plots*: 2D graphics of vector fields, bar charts, pie charts, and visualization of zonotopes.

In Scientific Visualization pages users can specify functions or submit data to generate a graphics image. A help-page displays the basic syntactic rules that should be followed in specifying the input. There are no restrictions on the quantity of the input data. The diagram presented on Fig. 2 shows a typical Scientific Visualization interactive page. Every page is equipped with a number of general and specific options that control the image: plot range, axes, axes origin, axes labels, ticks, frame, frame labels, plot label, image size, aspect ratio. The specific options

include e.g., possibilities to control the number and color of the contours, or the size and color of the points in data plots. Each option has assigned a default value so that the user be facilitated as much as possible.

A 3D graphics can be displayed either static or by the LiveGraphics3D applet [7]. The latter gives a real-time rotation of the three-dimensional graphics object.

All the static graphics images can be exported to the client machine in a specified graphics file format. Scientific Visualization supports 12 of the most frequently used graphics file formats: GIF and animated GIF, JPEG, Encapsulated PostScript, *Mathematica* abbreviated PostScript, PDF, Microsoft bitmap format, Encapsulated PostScript with device independent preview, Encapsulated PostScript with TIFF preview, *Mathematica* independent raster graphics format, portable bitmap format, X window system bitmap format, TIFF.

Scientific Visualization is designed as a free and easy web service intended for general users, for distance education and active training, as well as for technical professionals. It will expand by more graphics functionality, static demo pages, and facilities for data uploading.

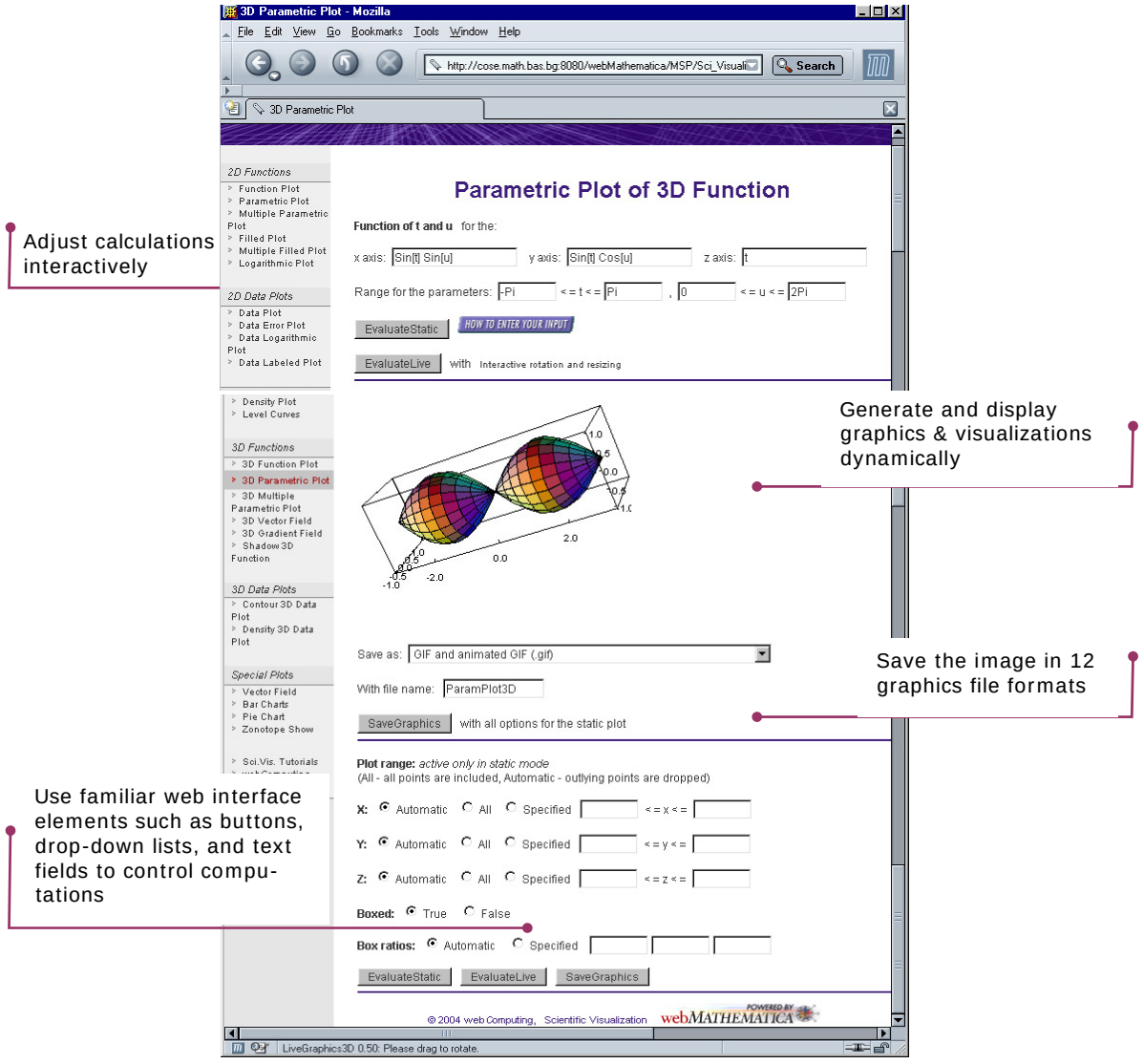


Fig. 2. A typical Scientific Visualization page.

webComputing Authoring Framework

webComputing provides hosting of sites for interactive computing, visualization and education that are built on web*Mathematica* technology. To this end, a webComputing Authoring Framework (wCAF) is designed. The goal of webComputing Authoring Framework is to allow *easy* and *remotely* development of dynamic and interactive web pages based on webMathematica technology. This service is provided for *Mathematica* users who intend to develop interactive courseware or other education materials based on the computational and visualization power of *Mathematica*, and for those who want to make their specific *Mathematica* software web accessible.

wCAF serves two groups of users: developers familiar with web*Mathematica* technology and beginners who need to explore and study the facilities provided by this technology. webComputing allows distance learning of webMathematica technology by a corresponding distance course providing:

- online (web accessible) static pages of web*Mathematica* documentation;
- live examples of interactive pages together with a display of their script code;
- access to script supporting directories of a web*Mathematica* server.

Thus, the content pages, the supporting background *Mathematica* packages, and other scripts can be developed independently but can interoperate from anywhere on the web.

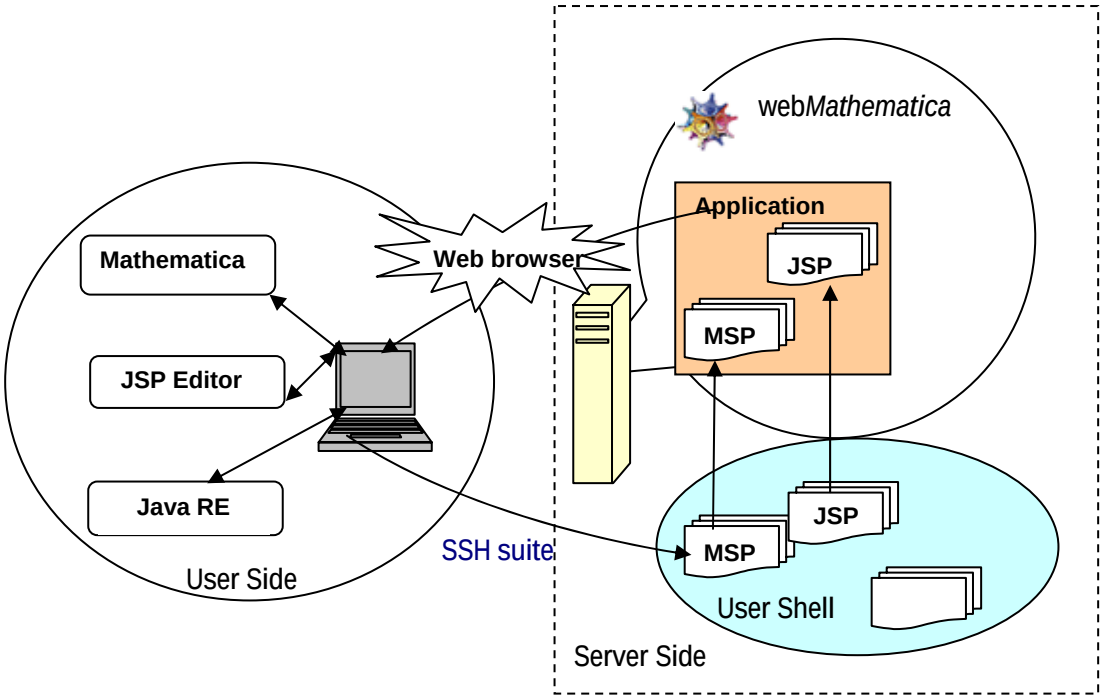


Fig.3. webComputing Authoring Framework concept and the overall architecture.

wCAF Architecture.Clearly, for wCAF to work, it must have front-end and back-end support. More importantly, such support must be delivered within an architectural framework that uses appropriate technologies to integrate components, allowing them to interoperate in a seamless manner on the Web.

- *Back-end support* – On the server-side wCAF architecture consists of three layers: user's home directories, webMathematica architecture connecting *Mathematica* to a web server, and other language, data base and technology support installed on the server. A user home directory has a structure corresponding to the tasks that will be performed and the server resources to be used. For developing interactive web pages e.g., two home subdirectories called MSP (containing web*Mathematica* ver. 1. script pages) and JSP (containing webMathematica ver. 2. script pages) are logically linked to corresponding domains of the webMathematica

architecture. The latter are accessible by URL and can be processed by web*Mathematica* technology. Thus the Mathematica Server Pages and Java Server Pages, developed remotely by the user, can be uploaded on the server and invoked by any web browser to be processed by web*Mathematica*.

- *Front-end support* – The following common tools should be available on the client side: SSH protocol suite of network connectivity tools for secure remote login over the networks [1]. SSH file transfer should be used for downloading matherials from the server and for uploading the developed MSP/JSPs. Any available common Web browsers can be used to view and test the designed MSP/JSPs. Additionally, Java Runtime Environment will be necessary for running interactive 3D graphics. Depending on the Internet technologies that clients want to use and combine in their scripts, other front-end features can be provided through JavaScript, MathML, Java applets, etc. Any JSP editor, or some common type editors, can be used for developing MSP/JSPs. *Mathematica* system should be either installed on the client machine, or run remotely on the server.
- *Content-markup support* – it is provided by web*Mathematica* technology.

The remote MSP developing framework is presented on Fig. 3.

Usage Paradigm of wCAF.

- Users (both developers and students) have accounts, with appropriate structure, on the COSE server. Home directories are accessible via SSH/SSF protocol from anywhere on the Internet, at any time.
- web*Mathematica* documentation, live MSP/JSP examples and their script code are Web accessible from anywhere at any time. Users can study web*Mathematica* technology at their own. They can invoke and try the example scripts. Designing own interactive pages authors can easily include and mix powerful script features developed by others.
- Authors develop interactive pages remotely either on a client machine or on the server-side. They focus on the mathematical/technical content of the pages rather than dealing with the underlying Internet programming. Regardless of the size of the application that is created, wCAF cuts the development time and makes the application more robust as well as easier to use and maintain.
- Authors test/debug their own developed interactive pages by web access to a corresponding URL. In general, debugging anything running inside a server can be difficult. However, web*Mathematica* framework provides some ways for making this process easier. When authoring JSPs one can use messages and print statements to resolve a possible problem. web*Mathematica* provides specific functions that capture of *Mathematica* message and print output and return this output into the corresponding web page. In addition to message and print output, the developers of MSP/JSPs can use a more sophisticated tool called Kernel Monitor. It is a servlet that collects information on the running of a web*Mathematica* site. The kernel monitor is accessible through a web page which shows important status information for the *Mathematica* kernel which allows studying the performance of a site; any logs that have been collected and the input to *Mathematica*. The page provides also a number of controls that cause the kernels to restart, the monitor to reload, the logs to be cleared, reloaded, etc.
- The developed dynamic web pages – containing interactive computations, visualization or education materials – can be used from anywhere on the Web. Each author can provide well-defined set of capabilities within a particular scope.
- Authors deploy and maintain their interactive computations, visualization or courseware simply and easily. They do not have to worry about session management and error recovery. Authors can modify, revise, and change their webComputing pages anytime from anywhere.
- COSE server and web*Mathematica* are upgraded and improved without affecting wCAF user developed pages as long as equivalent contents are returned.

Conclusion

webComputing framework is an effort to deploy dynamic and interactive mathematically oriented services via web interface. The authoring framework, based on *Mathematica* and web*Mathematica*, allows the development of highly interactive mathematical web applications to be done easily and the applications to be more powerful. The overall framework is intended to be open, flexible, and usable as a test bed for research or a platform for application developments. Our prototype implementations represent a first step in this direction. As with any groundbreaking new technology, the capabilities open up an array of new possibilities: deploying calculators and problem solvers, showcasing *Mathematica*-based work in interactive web documents, publishing sophisticated courseware, papers, and book supplements that involve highly interactive web based education tools and materials. Work on an expanded interval web computation site is ongoing. We are also considering specialized education services that foster explorative learning. We hope to collaborate with more people interested in web accessible mathematics. The goal is to contribute to the creation of a rich interactive content and to make dynamic mathematical computations easily accessible in many contexts world-wide.

Bibliography

[1] D.J.Barrett, R.E.Silverman. The Secure Shell, The Definitive Guide. O'Reilly, 2001.

[2] L.Beccerra, O.Sirisaengtaksin, B.Waller. On Categories of Interactive Computational Web Tools. ATCM 2000 Proceedings, Thailand, 2000.

[3] IAMC: Internet Accessible Mathematical Computations, home page (1999-2004). http://icm.mcs.kent.edu/research/iamc/

[4] R.Hammer, M.Hocks, U.Kulisch, D.Ratz. C++ Toolbox for Verified Computing: Basic Numerical Problems. Springer-Verlag, Berlin/Heidelberg/New York, 1995.

[5] W.Hofschuster, W.Krämer. C-XSC 2.0 – A C++ Class Library for Extended Scientific Computing. In: Alt, R. et al. (eds.): Numerical Software with Result Verification. Lecture Notes in Computer Science, Vol. 2991, Springer-Verlag, Berlin/Heidelberg/New York, 2004, 15-35.

[6] R.Klatte, U.Kulisch, K.Lawo, M.Rauch, A.Wiethoff. C-XSC, A C++ Class Library for Extended Scientific Computing. Springer-Verlag, Berlin/Hiedelberg/New York, 1993.

[7] M.Kraus, M.: LiveGraphics3D: a Java applet which can display Mathematica graphics. http://wwwvis.informatik.uni-stuttgart.de/~kraus/LiveGraphics3D/

[8] P.Libbrecht. Mathematical Systems Accessed on the Web. Proceedings of IAMC'02 Workshop, 2002. http://www.symbolicnet.org/conferences/iamc02/

[9] E.Popova, M.Ivanova. Scientific Visualization: A Service for Interactive Graphics Generation. In: Mathematics and Education in Mathematics, Proceedings of 34 Spring Conference of UBM, Sofia, 2005. (in Bulgarian)

[10] SCOPES 7 IP 65642 project: Establishing Computational Science and Engineering in Bulgaria and Macedonia. Supported by Swiss NSF, 2002-2004. http://cose.math.bas.bg/

[11] P.S.Wang. Design and Protocol for Internet Accessible Mathematical Computation. Proceedings ISSAC'99, ACM Press, 1999, 291-298.

[12] T.Wickham-Jones. web*Mathematica*: A User Guide, Version 1.0. Wolfram Research, August 2001.

[13] T.Wickham-Jones. web*Mathematica*: A User Guide, Version 2.0, April 2003, Wolfram Research.

[14] Wolfram Res. Inc.: web*Mathematica*, the way the web calculates. http://www.wolfram.com/products/webmathematica/

[15] S.Wolfram. The *Mathematica* Book, 4th ed. Wolfram Media/Cambridge U. Press, 1999.

Authors' Information

Evgenija D. Popova – Institute of Mathematics and Informatics, Bulgarian Academy of Sciences; Acad. G. Bonchev str. Block 8, Sofia-1113, Bulgaria; e-mail: epopova@bio.bas.bg

THE DEVELOPMENT OF THE GENERALIZATION ALGORITHM BASED ON THE ROUGH SET THEORY

Marina Fomina, Alexey Kulikov, Vadim Vagin

Abstract: This paper considers the problem of concept generalization in decision-making systems where such features of real-world databases as large size, incompleteness and inconsistence of the stored information are taken into account. The methods of the rough set theory (like lower and upper approximations, positive regions and reducts) are used for the solving of this problem. The new discretization algorithm of the continuous attributes is proposed. It essentially increases an overall performance of generalization algorithms and can be applied to processing of real value attributes in large data tables. Also the search algorithm of the significant attributes combined with a stage of discretization is developed. It allows avoiding splitting of continuous domains of insignificant attributes into intervals.

Keywords: knowledge acquisition, knowledge discovery, generalization problem, rough sets, discretization algorithm.

ACM Classification Keywords: H.2.8 Database Applications: data mining; I.2.6 Learning: knowledge acquisition; B.2.4 High-Speed Arithmetic: algorithms.

1. Introduction

Many enterprises in the various areas create and maintain huge databases with information about their activity. However without the productive analysis and generalization such streams of the "raw" data are useless. Due to the application of methods for information generalization in decision making systems, the construction of the generalized data models and processing of large arrays of experimental data are possible. There are sources of such large dataflows in many areas. Application domains of methods for generalization include marketing, medicine, the space researches and many others. Common for these data is that they contain a great many of the hidden regularities, which are important for the strategic solutions making. However, the discovery of these regularities lays outside the human possibilities mainly because of large and permanently increasing size of the data. Therefore the methods for generalization and computer systems implementing these methods are used to derive such regularities.

Concept generalization problem under redundant, incomplete or inconsistent information is very actual. The purpose of this paper is to consider opportunities of the using the rough set theory for solution of a problem of generalization, and to propose the methods improving work of known algorithms. The new discretization algorithm of continuous attributes and the search algorithm of the significant attributes which essentially increase an overall performance of algorithms for generalization will be proposed.

2. Statement of the Generalization Problem

For the description of object we will use features $a_1, a_2, \ldots, a_k$, which are further called attributes. Each object x is characterized by a set of given values of these attributes: $x=\{v_1, v_2, \ldots, v_k\}$, where v_i is value of the *i-th* attribute. Such description of an object is called *feature description*. For example, the attributes may be a color, a weight, a form, etc.

Let we have a training set U of objects. It contains both the positive examples (which are concerning to interesting concept) and the negative examples. The concept generalization problem is the construction of the concept allowing the correct classifying with the help of some recognizing rule (*decision rule*) of all positive and negative objects of training set U. Here the construction of the concept is made on the basis of the analysis of a training set.

Let's introduce the following notions related with set U. Let $U = \{x_1, x_2, \ldots, x_n\}$ is a non-empty finite set of objects. $A = \{a_1, a_2, \ldots, a_k\}$ is a non-empty finite *set of attributes*. For each attribute the set V_a is defined which refers to

the *value set* of attribute *a*. We will denote given value of attribute *a* for object $x \in U$ by $a(x)$. At the decision of a generalization problem often it is necessary to receive the description of the concept, which is specified by value of one of the attributes. We will denote such attribute *d* and call it *decision* or *decision attribute*. The attributes which are included in *A* are called *conditional attributes*. The decision attribute can have some values though quite often it is binary. The number of possible values of a decision attribute *d* is called the rank of the decision and is designated as $r(d)$. We will denote the value set of the decision by $V_d = \{v_1^d, v_2^d, ..., v_{r(d)}^d\}$. The decision attribute *d* defines the partition of *U* into classes $C_i = \{x \in U: d(x) = v_i^d\}$, $1 \le i \le r(d)$.

Generally the concept generated on the basis of training set *U* is an approximation to concept of set *X*, where the closeness degree of these concepts depends on the representativeness of a training set, i.e. how complete the features of set *X* are expressed in it.

3. Basic Notation of the Rough Set Theory

The rough set theory has been proposed in the beginning of 80th years of the last century by the Polish mathematician Z. Pawlak. Later this theory was developed by many researchers and was applied to the decision of various tasks. We will consider how the rough set theory can be used to solve concept generalization problem (also see [1-8]).

In Pawlak's works [1, 9] the concept of an information system has been introduced. An *information system* is understood as pair $S = (U, A)$, where $U = \{x_1, x_2, \ldots, x_n\}$ is a non-empty finite set of objects named *training set* or *universe*, and $A = \{a_1, a_2, \ldots, a_k\}$ is a non-empty finite *set of attributes*. A *decision table* (or *decision system*) is an information system of the form $S = (U, A \cup \{d\})$, where $d \notin A$ is a distinguished attribute called *decision* or *decision attribute*, *A* is a set of *conditional attributes*.

Let us introduce the *indiscernibility* or *equivalence relation* on the training set *U*: $IND(A) \subseteq U \times U$. We will say, that if $(x, y) \in IND(A)$ then *x* and *y* are indiscernible by values of attributes from *A*. A set of equivalence classes of relation *IND(A)* is denoted by $\{X_1^A\ X_2^A, \ldots, X_m^A\}$. Then we can approximately define set *X* using attribute values by the constructing of the lower and upper approximations of *X*, designated by $\underline{A}X$ and $\overline{A}X$ respectively. As a *lower approximation* of set *X* we will understand the union of equivalence classes of an indiscernibility relation which belongs to *X*, i.e. $\underline{A}X = \bigcup \{X_i^A \mid X_i^A \subseteq X\}$. And as an *upper approximation* of set *X* we will understand the union of equivalence classes which part of objects belongs to *X*, i.e. $\overline{A}X = \bigcup \{X_i^A \mid X_i^A \cap X \neq \varnothing\}$. The set $U \setminus \overline{A}X$ will consist of *negative objects* for *X*. A set $POS_A(d) = \underline{A}C_1 \cup \ldots \cup \underline{A}C_{r(d)}$ includes objects, which are guaranteed concerning to one of the decision classes, and this set is called *positive region* of the decision system *S*.

Rough set X is formed by pair $< \underline{A}X, \overline{A}X >$. If upper and lower approximations of *X* are equal then *X* is an ordinary set.

The equivalence relation can be associated not only with the full set of conditional attributes *A* but also with any attribute subset $B \subseteq A$. Further this relation is denoted as $IND(B)$ and is called a *B-indiscernibility relation*. Formally the *B*-indiscernibility relation is defined as follows: $IND(B) = \{(x, y) \in U \times U: \forall a \in B\ (a(x) = a(y))\}$.

Thus two objects belong to same equivalence class, if they cannot be discerned by the given subset of attributes. The concepts of *B*-upper and *B*-lower approximations based on $IND(B)$ are similarly introduced.

Since it is not always possible to find a single-valued decision for all objects of a decision system, we will introduce notion of a generalized decision. We will define function $\partial_B: U \to P(V_d)$ which is called a *generalized decision* of *S* on a set of attributes $B \subseteq A$, as follows: $\partial_B(x) = \{v \in V_d : \exists x' \in U\ (x' IND(B) x \wedge d(x') = v)\}$. The generalized decision ∂_A of a system *S* is simply called the generalized decision of *S*. Instead of ∂_A we also will write ∂_S. The decision table *S* is *consistent*, if $|\partial_A(x)| = 1\ \forall x \in U$, otherwise *S* is *inconsistent*.

Since not all conditional attributes are equally important, some of them can be excluded from a decision table without loss of the information contained in the table. The minimal subset of attributes $B \subseteq A$ which allows to keep

the generalized decision for all objects of a training set, i.e. $\partial_B(x) = \partial_A(x)\ \forall x \in U$, is called a *decision-relative reduction* of a table $S = (U, A \cup \{d\})$. In the sequel, when considering decision tables, instead of a decision-relative reduction we will use a *reduction*.

Now let us consider the methods for concept generalization.

4. Methods of the Rough Set Theory

Generally a work of the algorithm based on a rough set theory consist of the following steps: search of equivalence classes of the indiscernibility relation, search of upper and lower approximations, search of a reduction of the decision system and constructing a set of decision rules. Moreover discretization is applied to processing attributes with a continuous domain. In the case of the incomplete or inconsistent input information the algorithm builds two systems of decision rules, one of them gives the certain classification, the second gives the possible one. Further we will consider the most labour-consuming steps: search of reduction and discretization making.

4.1. The Problem of Search of Reduction

Let's consider the process of search of a reduction that is very important part of any method used the rough set approach. Quite often an information system has more than one reduction. Each of these reductions can be used in procedure of decision-making instead of a full set of attributes of original system without a change of dependence of the decision on conditions that is characteristic for original system. Therefore the problem of a choice of the best reduction is reasonable. The answer depends on an optimality criterion related to attributes. If it is possible to associate with attributes the cost function which expresses complexity of receiving attribute values then the choice will be based on criterion of the minimal total cost. Otherwise as a rule the shortest reduction is chosen. However the complexity of a search of such reduction consists in that the problem for checking whether exist a reduction, which length is less than some integer *s*, is NP-complete. The problem of searching for a reduction with minimal length is NP-hard [10].

Thus the problem of a choice of relevant attributes is one of the important problems of machine learning. There are several approaches based on rough set theory to its decision.

One of the first ideas was to consider as the relevant attributes those attributes which contain in intersection of all reductions of an information system.

Other approach is related to dynamic reductions [2], i.e. conditional attribute sets appearing "sufficiently often" as reductions of sub-samples of an original decision system. The attributes belonging to the "most" of dynamic reductions are considered as relevant. The value thresholds for "sufficiently often" and "most" should be chosen for a given data.

The third approach is based on introduction of the notion of significance of attributes that allows by real values from the closed interval [0,1] to express how important an attribute in a decision table.

4.2. Discretization Making

The stage of discretization is necessary for the most of modern algorithms for generalization. The discretization is called a transformation of continuous domain of attributes in a discrete one. For example, the body temperature of the human being which is usually measured by real numbers can be divided into some intervals, corresponding to the low, normal, high and very high temperature. The choice of suitable intervals and partition of continuous domains of attributes is a problem, whose complexity grows in exponential dependence on the number of attributes to which discretization should be applied.

Let's give formal definition of a considered discretization task. Let $S = (U, A \cup \{d\})$ is a consistent decision system. We will assume that the domain of any attribute $a \in A$ is a real interval, i.e. $V_a = [l_a, r_a) \subset R$. Any pair of the form p^a = (a, c) where $a \in A$ and $c \in R$, we will call *cut* on areas V_a. For each attribute $a \in A$ a set P_a = { $[c_0^a, c_1^a), [c_1^a, c_2^a), ..., [c_{s_a}^a, c_{s_a+1}^a)$ } where s_a is some integer, $l_a = c_0^a < c_1^a < ... < c_{s_a}^a < c_{s_a+1}^a = r_a$ and $V_a = [c_0^a, c_1^a) \cup [c_1^a, c_2^a) \cup ... \cup [c_{s_a}^a, c_{s_a+1}^a)$, we will call *partition* of a domain V_a. It is easy to notice, that the partition P_a is

uniquely defined by $C_a = \{ c_1^a \; c_2^a , \ldots, \; c_{s_a}^a \}$, which is called *set of cuts* of V_a. Therefore in the sequel we often will name P_a by a set of cuts and write down as $P_a = \{a\} \times C_a = \{(a,c_1^a),(a,c_2^a),\ldots,(a,c_{s_a}^a)\}$. Then full set of cuts P can be presented as $P = \bigcup_{a \in A} \{a\} \times C_a$.

Any set of cuts P on the basis of an original decision system $\mathrm{S} = (U, A \cup \{d\})$ determines a new decision system $\mathrm{S}^P = (U, A^P \cup \{d\})$, where $A^P = \{a^P : a \in A\}$ and $a^P(x) = i \Leftrightarrow a(x) \in [c_i^a, c_{i+1}^a)$ for any object $x \in U$ and $i \in \{0, \ldots, s_a\}$. A decision table S^P is called *P-discretization* of the table S. Our purpose is that during discretization to construct such set of cuts P.

It is obvious, that is possible to construct many of sets of cuts. Therefore there is a question how among them to find set with the minimal number of elements. For this purpose we will introduce the following concepts.

Two sets of cuts P and P' we will regard as equivalent, if $\mathrm{S}^P = \mathrm{S}^{P'}$. We will say that set of cuts P is consistent with S, if generalized decisions of systems S and S^P are equal, i.e. $\partial_{\mathrm{S}}(x) = \partial_{\mathrm{S}^P}(x) \;\; \forall x \in U$. The consistent set of cuts P^{irr} is *irreducible* in S if any its own subset is not consistent with S. Finally, the consistent set of cuts P^{opt} we will call *optimal* in S if it has the minimal cardinality among sets of cuts which are consistent with S.

The problem of finding optimal set of cuts P for the given decision system S is NP-complete [11]. This fact clearly speaks about importance of development of effective heuristic algorithms for search of suboptimal set of cuts.

The general approach of the most of discretization algorithms is based that any irreducible set of cuts of a decision table S is a reduction of other decision table $\mathrm{S}^* = (U^*, A^* \cup \{d^*\})$ constructed on a basis of S as follows [11].

Let $\mathrm{S} = (U, A \cup \{d\})$ be an original decision table. An arbitrary attribute $a \in A$ defines sequence $v_1^a < v_2^a < \ldots < v_{n_a}^a$, where $\{v_1^a, v_2^a, \ldots, v_{n_a}^a\} = \{a(x) : x \in U\}$ and $n_a \le n$. Objects of new decision table S^* are all pairs of objects of S with different decisions, and the set of conditional attributes is defined as cuts of attribute domains of an original decision table, i.e. $A^* = \bigcup_{a \in A} \left\{ p_i^a : p_i^a = (a, c_i^a), \text{ where } c_i^a = (v_i^a + v_{i+1}^a)/2, \; 1 \le i \le n_a - 1 \right\}$.

These attributes are binary. Set A^* is named an *initial set of cuts*. We will speak, that the cut $p_i^a = (a, c_i^a)$ *discerns* objects x and y of different decision classes, if $min(a(x), a(y)) < c_i^a < max(a(x), a(y))$. A value of the new attribute corresponding to a cut p_i^a for pair (x, y) is equal to 1 if objects x and y are discerned by this cut, and 0 otherwise. Moreover a new object $\perp$ for which all conditions and the decision d^* are 0 is added to the objects of a new decision table. For all other objects of a new decision table the new decision value is equal to 1. Reductions of a new decision table S^* determine all irreducible sets of cuts of an original decision table S.

On the basis of this general layout the heuristic algorithms finding a suboptimal set of cuts are developed. Often the discretization algorithm based on straightforward implementation of Jonson's strategy [8,12] is used. Computational complexity of this algorithm is equal to $O(|P| \cdot kn^3)$. It does its inapplicable for processing large databases. Thus, the main problem of discretization stage of continuous attributes is its high computational complexity. Now we propose the effective modification that solves this problem.

4.3. The Modification of the Discretization Algorithm

Our algorithm is directed towards the decreasing of time and memory consumption. It is based on the Jonson's strategy and extension of idea of iterative calculation of number of pairs of objects, discerned by a cut. This idea has been offered in [4], however, originally, it is applicable only when some restrictions on the decision table are imposed. This idea is based on assumption that there is a close relation between two consecutive cuts. So, for example, it is possible to notice, that in each row of the table S^* all the cells with value 1 are placed successively within one attribute. Therefore some pairs of objects are discerned by both consecutive cuts, and changes in the number of discernible pairs of objects can be only due to objects which attribute values lay between two these cuts. In [4], the situation, when no more than one object lies in this interval, is considered. We generalize this idea

on a case of the arbitrary number of such objects. Thus, our algorithm extends idea of iterative calculating number of pairs of objects discerned by a cut to an arbitrary decision table.

For some cut $p_t^a = (a, c_t^a) \in A^*$ for the attribute a where $a \in A$ and $1 \le t \le n_a$, and some subset $X \subseteq U$ we introduce the following notation: $W^X(p_t^a)$ is a number of pairs of objects from X discerned by a cut p_t^a; $I^X(p_t^a)$ and $r^X(p_t^a)$ is the number of objects from X, which have a value of the attribute a less (more) than c_t^a; $I_q^X(p_t^a)$ and $r_q^X(p_t^a)$ is the number of objects from X, which have a value of the attribute a less (more) than c_t^a and belong to the q-*th* decision class, where q = 1, …, $r(d)$; $N^X(p_t^a, p_{t+1}^a)$ is the number of objects from X, values of the attribute a which lay in an interval (c_t^a, c_{t+1}^a); $N_q^X(p_t^a, p_{t+1}^a)$ is the number of objects from X, values of attribute a which lay in an interval (c_t^a, c_{t+1}^a) and belonging to the q-*th* decision class, where q = 1, …, $r(d)$.

Now we formulate two our theorems which underlie proposed discretization algorithm. The first theorem will allow us to derive value $W^X(p_{t+1}^a)$ from $W^X(p_t^a)$, where p_t^a and p_{t+1}^a are two consecutive cuts of a domain of the attribute a.

Theorem 1. Let set $X \subseteq U$ consists of $N^X(p_t^a, p_{t+1}^a)$ objects which values of the attribute a belongs to an interval (c_t^a, c_{t+1}^a). Then

(a) $I_q^X(p_{t+1}^a) = I_q^X(p_t^a) + N_q^X(p_t^a, p_{t+1}^a)\ \forall q = 1,...,r(d)$;

(b) $r_q^X(p_{t+1}^a) = r_q^X(p_t^a) - N_q^X(p_t^a, p_{t+1}^a)\ \forall q = 1,...,r(d)$;

(c) $W^X(p_{t+1}^a) = W^X(p_t^a) + N^X(p_t^a, p_{t+1}^a) \cdot \left(r^X(p_t^a) - I^X(p_t^a)\right) -$

$$-\sum_{i=1}^{r(d)} N_i^X(p_t^a, p_{t+1}^a) \cdot \left(r_i^X(p_t^a) - I_i^X(p_t^a)\right) + \sum_{i=1}^{r(d)} \left(N_i^X(p_t^a, p_{t+1}^a)\right)^2 - \left(N^X(p_t^a, p_{t+1}^a)\right)^2.$$

Let's consider a case when during discretization we have a set of cuts $P \subseteq A^*$ that defines equivalence classes X_1, X_2, …, X_m of the indiscernibility relation $IND(A^P)$ of table S^P, and also two consecutive cuts p_t^a and p_{t+1}^a of the attribute a. Then we can calculate value $W_P(p_{t+1}^a)$ from $W_P(p_t^a)$ as follows:

Theorem 2. Let there are K equivalence classes $X_{\alpha_1}, X_{\alpha_2}, ..., X_{\alpha_K}$ to each of which belongs $N^{X_{\alpha_i}}(p_t^a, p_{t+1}^a) \ge 1$ objects which values of attribute a are within an interval (c_t^a, c_{t+1}^a). Then

$$W_P(p_{t+1}^a) = W_P(p_t^a) + \sum_{i=1}^{K} \left[N^{X_{\alpha_i}}(p_t^a, p_{t+1}^a) \cdot \left(r^{X_{\alpha_i}}(p_t^a) - I^{X_{\alpha_i}}(p_t^a)\right) - \sum_{q=1}^{r(d)} N_q^{X_{\alpha_i}}(p_t^a, p_{t+1}^a) \cdot \left(r_q^{X_{\alpha_i}}(p_t^a) - I_q^{X_{\alpha_i}}(p_t^a)\right) + \right.$$

$$\left. \sum_{q=1}^{r(d)} \left(N_q^{X_{\alpha_i}}(p_t^a, p_{t+1}^a)\right)^2 - \left(N^{X_{\alpha_i}}(p_t^a, p_{t+1}^a)\right)^2 \right].$$

Now we present steps of our algorithm. We will name its *GID* (Generalized Iterative algorithm for Discretization).

Algorithm 1. *Algorithm GID*.

Input: The consistent decision table S.

Output: Suboptimal set of cuts P.

Used data structures: P is a suboptimal set of cuts, L = [$IND(A^P)$)] – the set of equivalence classes of an indiscernibility relation of the table S^P; A^* - a set of possible cuts.

1. $P := \varnothing$; $L := \{U\}$; A^*: = initial set of cuts;
2. For each attribute $a \in A$ do:
   begin
   $W_P(p_0^a) := 0$;
   For each $X_i \in L$ do:

$r^{X_i} := |X_i|$; $I^{X_i} := 0$;

for q = 1,..., $r(d)$ assign $r_q^{X_i} := |\{x \in X_i : d(x) = v_q^d\}|$; $I_q^{X_i} := 0$;

For each cut $p_j^a = (a, c_j^a) \in A^*$ do:

For all classes X_{α_i} which objects have a value of attribute a from an interval (c_{j-1}^a, c_j^a) to calculate $N^{X_{\alpha_i}}$ and $N_q^{X_{\alpha_i}}$.

Find $W_P(p_j^a)$ according to the theorem 2.

Count values $r^{X_{\alpha_i}}, I^{X_{\alpha_i}}$ and $r_q^{X_{\alpha_i}}, I_q^{X_{\alpha_i}}$ under the theorem 1.

end;

3. Assume as $p_{\max}$ the cut with maximal value $W_P(p)$ among all cuts p from A^*.
4. Assign $P := P \cup \{p_{\max}\}$; $A^* := A^* \setminus \{p_{\max}\}$;
5. For all $X \in L$ do: if $p_{\max}$ divides the set X into X_1 и X_2 then remove X from L and add to L two sets X_1 and X_2.
6. If all sets from L consist of the objects belonging to same decision class then Step 7 otherwise go to the Step 2.
7. End.

Let's estimate computational complexity of offered algorithm. The most labour-consuming steps of algorithm are the second and the fifth.

On step 2, during calculation of number of pairs of objects discerned by a cut $p_j^a = (a, c_j^a)$ values $r^{X_{\alpha_i}}$, $I^{X_{\alpha_i}}$, $N^{X_{\alpha_i}}$ and $r_q^{X_{\alpha_i}}$, $I_q^{X_{\alpha_i}}$, $N_q^{X_{\alpha_i}}$ are changed, where q = 1, …, $r(d)$. These operations are carried out only for those equivalence classes X_{α_i}, even which one object satisfies to the condition of belonging of value of attribute a to interval (c_{j-1}^a, c_j^a). For one such equivalence class it will be executed $3 \cdot r(d)+3$ described operations. We will designate this number as α. It does not depend on the number of objects n and the number of attributes k. The number of such equivalence classes cannot exceed the number n_j of objects which belong to them and which value of attribute a are in interval (c_{j-1}^a, c_j^a). Hence, during calculation $W_P(p_j^a)$ for one cut p_j^a it is carried out no more than $\alpha \cdot n_j$ operations. Therefore during processing all cuts of one attribute it will be executed $\sum_{j=1}^{n-1} \alpha \cdot n_j \le \alpha \cdot n$ operations. For processing the cuts of all k attributes it is required $\alpha \cdot kn$ operations. The second step repeats $|P|$ times. It means, that its total computational complexity is equal to $O(|P| \cdot kn)$.

On step 5 splitting equivalence classes is carried out. We take the worse case when finally any class consists of exactly one object. Since there are n objects then during work of the algorithm it will be executed n-1 splitting operations. Hence, computational complexity of the fifth step is $O(n)$.

Thus total computational complexity of the proposed discretization algorithm is equal to $O(|P| \cdot kn) + O(n) = O(|P| \cdot kn)$. It is less on two orders than computational complexity of Jonson's algorithm.

Also we estimate the space complexity of our algorithm. It should be noticed that it does not build the auxiliary table S^*. It is required only $k(n$-1) memory cells for a storing set of possible cuts from A^*, n cells for designating an equivalence class to which belongs each of the objects, and no more than $\alpha \cdot n$ cells for storing numbers r^{X_i}, I^{X_i}, N^{X_i} and $r_q^{X_i}$, $I_q^{X_i}$, $N_q^{X_i}$ for all equivalence classes $X_i \in L$ where $i \le n$ and q = 1,..., $r(d)$ and the value α does not depend on k and n. Hence the space complexity of our discretization algorithm is equal to $O(kn)$. It is less on the order than space complexity of Jonson's algorithm. For more details about our algorithm see [5, 6].

4.4. The Modification of Algorithm for Searching the Significant Attributes

In the majority of the algorithms which are based on the rough set theory and carrying out splitting of continuous attribute domains into finite number of intervals, the stage of discretization is considered as preparatory before search of significant attributes. And consequently at a stage of discretization there is a splitting of the domains of all continuous attributes, including insignificant. In this work the combined implementation of discretization with the search of a reduction is offered to make discretization only for those quantitative attributes which appear to be significant during search of a reduction.

Besides, as significant attributes we will consider the attributes which are included in approximate reductions with sufficiently high quality. The concept of an approximate reduction [8] represents generalization of concept of the reduction considered earlier. Any subset B of set A can be considered as an *approximate reduction* of set A, and value

$$\varepsilon_{(A,d)}(B) = \frac{dep(A,d) - dep(B,d)}{dep(B,d)} = 1 - \frac{dep(B,d)}{dep(A,d)}$$

is named a *reduction approximation error*. Here the value $dep(B, d)$ represents a measure of dependence between $B \subseteq A$ and d: $dep(B, d) = |POS_B(d)|/|U|$. The reduction approximation error shows how precisely the set of attributes B approximates whole set of conditional attributes A (relatively d). Application of approximate reductions is useful while processing inconsistent and noisy data.

Thus, the developed algorithm for search of significant attributes is based on two ideas: 1) combination of discretization of quantitative attributes with the search of significant attributes, 2) search for an approximation of a reduction, but no for reduction itself. Let's name it as Generalized Iterative algorithm based on the Rough Set approach, GIRS.

4.5. Results of the Experiments

The realized experiments show that the developed algorithm allows reducing time for search of significant attributes essentially, due to combination with discretization stage and use of proposed algorithm GID.

The results of the experiments executed on 11 data sets from a well known collection UCI Machine Learning Repository [7] of the University of California are given in Table 1.

Data set	Classification accuracy				
	ID3	C4.5	MD	Holte-II	GIRS
Monk-1	81.25	75.70	100	100	100
Monk-2	65.00	69.91	99.70	81.9	83.10
Monk-3	90.28	97.20	93.51	97.2	95.40
Heart	77.78	77.04	77.04	77.2	78.72
Hepatitis	n/a	80.80	n/a	82.7	84.51
Diabetes	66.23	70.84	71.09	n/a	81.00
Australian	78.26	85.36	83.69	82.5	88.71
Glass	62.79	65.89	66.41	37.5	70.10
Iris	94.67	96.67	95.33	94.0	96.24
Mushroom	100	100	100	100	100
Soybean	100	95.56	100	100	100
Average	81.63	83.18	88.67	85.3	88.89

Table 1. Comparison of classification accuracy of the developed algorithm with other known generalization algorithms.

For all data sets taken into the comparison, the developed algorithm has shown classification accuracy that not concedes to other generalization algorithms, and in some cases surpasses it. Average accuracy of classification is approximately 88.9 %. It is necessary to note, that the classification accuracy received by our algorithm is much above that the classification accuracy achieved by methods of an induction of deciding trees (ID3, ID4, ID5R,

C4.5) at the solving the majority of the problems. It is explained by the impossibility of representation of the description of some target concepts as a tree. Moreover it is possible to note that combining of search of significant attributes and discretization procedure is very useful. Most clearly it is visible from the results received at the decision of the Australian credit task. It is possible to explain by the presence in these data the attributes both with continuous and with discrete domains. The modification of search procedure of significant attributes is directed namely to processing of such combination.

Conclusion

We have considered the concept generalization problem and the approach to its decision based on the rough set theory. The means provided by this approach have been shown. They allow solving the problem of processing of real-world data arrays. The heuristic discretization algorithm directed towards the decreasing of time and memory consumption has been proposed. It is based on Jonson's strategy and extension of idea of iterative calculating number of pairs of objects discerned by a cut. Computational and space complexities of the proposed algorithm have linear dependence on the number of objects of decision table. Also the search algorithm of the significant attributes combined with a stage of discretization is developed. It allows avoiding splitting into intervals of continuous domains of insignificant attributes.

Bibliography

[1] Pawlak Z. Rough sets and intelligent data analysis / Information Sciences, Elsevier Science, November 2002, vol. 147, iss. 1, pp. 1-12.

[2] Bazan J. A comparison of dynamic non-dynamic rough set methods for extracting laws from decision tables / Rough Sets in Knowledge Discovery 1: Methodology and Applications // Polkowski L., Skowron A. (Eds.), Physica-Verlag, 1998.

[3] Vagin V.N., Golovina E.U., Zagoryanskaya A.A., Fomina M.V. Dostovernyi i pravdopodobnyi vyvod v intellektual'nyh sistemah (Reliable and plausible inference in intellectual systems) / Pod red. V.N. Vagina, D.A. Pospelova. Moscow, Fizmatlit, 2004. – 704 p.

[4] Nguyen S.H., Nguyen H.S. Some efficient algorithms for rough set methods / Proc. of the Conference of Information Processing and Management of Uncertainty in Knowledge-Based Systems, Spain, 1996, pp. 1451-1456.

[5] Kulikov A., Fomina M. The Development of Concept Generalization Algorithm Using Rough Set Approach / Knowledge-Based Software Engineering: Proceedings of the Sixth Joint Conference on Knowledge-Based Software Engineering (JCKBSE 2004) // V.Stefanuk and K. Kajiri (eds). – IOS Press, 2004. – pp.261–268.

[6] Vagin V.N., Kulikov A.V., Fomina M.V. Methods of Rough Sets Theory in Solving Problem of Concept Generalization / Journal of Computer and System Sciences International, Vol. 43, No. 6, 2004. – pp. 878 - 891.

[7] Merz C.J., Murphy P.M. UCI Repository of Machine Learning Datasets. – Information and Computer Science University of California, Irvine, CA, 1998. http://www.ics.uci.edu/~mlearn/MLRepository.html.

[8] Komorowski J., Pawlak Z., Polkowski L., Skowron A. Rough Sets: A Tutorial. / Rough Fuzzy Hybridization, Springer-Verlag, 1999.

[9] Pawlak Z. Rough Sets / International Journal of Information and Computer Science. 1982, 11(5), pp. 341-356.

[10] Skowron A., Rauszer C. The Discernibility Matrices and Functions in Information Systems / Intelligent Decision Support - Handbook of Applications and Advances of the Rough Sets Theory, Kluwer, 1992, pp. 331-362.

[11] Nguyen H.S., Skowron A. Quantization of real value attributes / Second Annual Joint Conference on Information Sciences (JCIS'95) // Wang P.P. (ed.), North Carolina, USA, 1995, pp. 34-37.

[12] H.S. Nguyen, S.H. Nguyen. Discretization methods in data mining / Rough Sets in Knowledge Discovery 1: Methodology and Applications // Polkowski L. and Skowron A. (Eds.), Heidelberg, Physica-Verlag, 1998. pp. 451-482.

Authors' Information

Marina Fomina – e-mail: fominhome@mtu-net.ru

Alexey Kulikov –e-mail: kulikov@apmsun.mpei.ac.ru

Vadim Vagin – e-mail: vagin@apmsun.mpei.ac.ru

Moscow Power Engineering Institute, Krasnokazarmennaya str, 14, Moscow, 111250, Russia;

THE DEVELOPMENT OF PARALLEL RESOLUTION ALGORITHMS USING THE GRAPH REPRESENTATION

Andrey Averin, Vadim Vagin

Abstract. The parallel resolution procedures based on graph structures method are presented. OR-, AND- and DCDP- parallel inference on connection graph representation is explored and modifications to these algorithms using heuristic estimation are proposed. The principles for designing these heuristic functions are thoroughly discussed. The colored clause graphs resolution principle is presented. The comparison of efficiency (on the Steamroller problem) is carried out and the results are presented. The parallel unification algorithm used in the parallel inference procedure is briefly outlined in the final part of the paper.

Keywords: Automated Reasoning, Logical inference

ACM Classification Keywords: I.2.3 Artificial Intelligence: Deduction and Theorem Proving

1. Introduction

The deductive inference procedures are broadly used in variety of fields, such as expert systems, decision support systems, deductive databases and intelligent information systems. Due to high amount of data in practical problems and the exponential growth of the search space, the efficiency of deductive procedures becomes the key factor in the development of deductive inference systems. One of the ways to improve the efficiency is the parallelization of inference procedures. We present a parallel inference procedure based on resolution principle. The connection graph representation is chosen as the basis for designing the parallel resolution procedures. Using the graph representation simplifies the parallelization of the inference process and allows to apply the different parallelization techniques such as OR, AND and DCDP parallelism. We study and implement OR-, AND- and DCDP- parallel resolution procedures, develop useful heuristics which can be used in parallel resolution procedures on connection graphs and make a comparison of the obtained results with the results of algorithms developed by other researchers. Also we describe and implement the clause graphs inference procedure. As the test task, the Schubert's Steamroller problem is examined [1]. The problem of parallelism on the term level is also investigated. The data structure for the term representation and the parallel unification algorithm using this data structure are presented.

2. Connection Graph

The connection graph method was designed by R. Kowalski [2]. A connection graph is a scheme for representing the proper first-order formulas in disjunctive normal form. Each literal is associated with a node in the connection graph. Literals in a clause are combined into a group. If the literals in two clauses form a contrary pair (P and $\neg$P) then there is an edge between the respective nodes of the connection graph.

Example 1.

The initial set of clauses:

1.Q(c)
2. Q(b)
3.R(x) $\vee \neg$Q(y) $\vee \neg$P(x)
4.P(b)
5.$\neg$R(x) $\vee$S(x,y) $\vee \neg$T(x)
6.T(y) $\vee \neg$B(y)
7.B(a)
8.$\neg$F(y) $\vee \neg$S(y,z) $\vee \neg$B(z)
9.B(x) $\vee \neg$C(x) $\vee \neg$D(y)
10.D(c)
11.F(b)
12.F(c)
13.C(b)

The corresponding connection graph is shown in fig. 1.

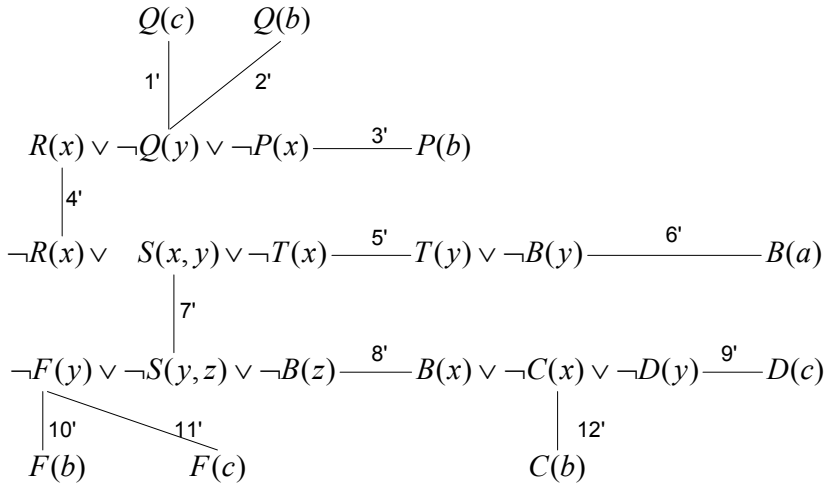


Fig.1

3. Methods of Inference on Connection Graphs

3.1 The Sequential Proof Procedure

To prove the unsatisfiability of a clause set we must generate and resolve the initial connection graph, i.e. derive an empty clause.

The main operation in connection graph refutation is the link resolution, when the resolvent is computed and added to the graph. The corresponding link is deleted and the links of the added resolvent are inserted. A pure clause is a literal group containing a node with no links. Pure clause with all its links can be easily removed from the graph without losing the completeness of the connection graph refutation procedure. Similarly, if we have a tautology clause, it also can be removed from a graph. If a resolvent on some step is a pure clause or a tautology, there is no need to insert this clause into a graph.

The refutation algorithm consists of the following steps:

1. the verification whether there is any clause in a graph or not. If there are no clauses, the algorithm terminates unsuccessfully. If there is the empty clause, then the algorithm is successfully terminated, else go to step 2;
2. if a graph does not contain any link, then the algorithm is unsuccessfully terminated, else go to step 3;
3. a link selection. The link is resolved and the resolvent is generated;
4. if an empty resolvent is obtained, then the algorithm terminates successfully, else the resolvent is inserted into the graph, its links are added, and the algorithm goes to step 2.

The fundamental problem in the connection graph refutation is the choice of suitable links by some criteria at each step of an algorithmic operation. Links are usually selected by using heuristics.

3.2 Parallel Inference on the Kowalski Connection Graph

The Kowalski connection graph can easily be used as the basis for designing parallel resolution algorithms [3]. Since the search space is complete, there is a possibility of using parallel computation strategies for enhancing the inference procedure efficiency. Parallel resolution algorithms differ from the sequential algorithm in step 3 at which a set of links satisfying certain criteria (not a single link as in the sequential procedure) is chosen and parallel resolution of all the links in this set is carried out.

3.2.1. OR-parallel Resolution on a Connection Graph

In case of OR-parallelism, the inference system associates some goal clause with the heads of clauses – possible candidates for resolution. Literals are unified and new clauses are generated. Admissible OR-links sets for the connection graph of Fig.1 are {1', 2'} and {10', 11'}.

3.2.2. DCDP-parallel Resolution on a Connection Graph

One modification of the parallel inference on the Kowalski connection graph is called DCDP parallelism (parallelism for distinct clauses) [4]. The correctness of the DCDP parallel resolution is proved in [5].

Definition 1. Clauses are said to be adjacent if there exist one or several links joining the literals of one clause with the literals of another clause.

Definition 2. A set of links joining pairs of distinct clauses is called a DCDP-link set if the clauses of every pair are not adjacent to any clause of other pairs.

To illustrate these definitions let us study the DCDP-link set for the connection graph of Fig.1. Adjacent pairs of clauses for this set are {(1), (3)}, {(2), (3)}, {(3), (4)}, etc... Thus, one of the DCDP-link sets is {1', 6', 9'}. Other examples of DCDP-link sets are: {2', 6', 12'}, {4', 12'}, {1', 6', 10'}.

3.2.3 AND-parallel Inference

Definition 3. A clause where all its links are resolved in parallel is called a SUN-clause.

Definition 4. Clauses joined with the literals of a SUN-clause are called the satellite clauses.

In AND-parallelism all links of literals of the SUN clause are resolved simultaneously. All resolvents are inserted in the graph along with all inherited links of a satellite clause. A SUN clause with all its links is removed. There is proved the correctness of the AND-parallel resolution [5]. The correct unification of separable variables under AND-parallel resolution is studied in [3].

Let us consider the choice of an AND-link set for the connection graph of fig.1. Admissible AND-link sets are, for example, {5', 4', 7'} (SUN-clause –clause (5)) and {5',6'} (SUN-clause – clause (6)). Detailed description of methods and algorithms can be found in [3,6].

4. Modification of Parallel Inference Procedures

Different heuristics can be used for choosing a link in resolving upon a connection graph. In the parallel resolution we must choose a set of links satisfying certain conditions. Note that the inference procedure becomes unsuitable if links are chosen unsuccessfully. The main principles underlying the design of heuristics are:

1. the number of literals in resolved clauses must be minimal,
2. the number of links in resolved clauses must be minimal,
3. the number of links in a literal for which the clauses are resolved must be minimal,
4. the unifiers of a resolved link must have a substitution of the type {*c*/*x*}, where *c* is a constant or a functional term, and *x* is a variable.

Further we describe the meaning of each principle in more detail.

Principle (1) simplifies a resulting connection graph, because a clause with a small number of literals, usually has a fewer number of links.

Principle (2) also simplifies a resulting connection graph, because the resolution of clauses with a small number of links yields clauses also with a small number of links.

Principle (3) prefers those links, the resolution of which yields "pure" clauses that on removing, can considerably simplify a connection graph.

The same is true for principle (4): as a result of the resolution by links containing a substitution of a variable for a constant, we obtain a clause containing constant terms. Such a clause has, at the first, a small number of links and, at the second, can be effectively used in the resolution. Principles (I)-(4) are taken into consideration in the heuristic function described below.

4.1. The Heuristic Function HI

In the heuristic function *H1* the link estimation is represented as a linear combination of estimations of the objects in the link (unifiers, clauses, and predicate literals in the link).

4.1.1 Computation of the Value of the Heuristic Function

Let *WeightLink* denote the heuristic estimation of a link. Then:

$$WeightLink = k_{clause} \cdot (Clause_{1Heur} + Clause_{2Heur}) + Uni_{Heur} \cdot k_{uni} + k_{pred} \cdot (Pred_{1Heur} + Pred_{2Heur}),$$

where $Clause_{1Heur}$, $Clause_{2Heur}$, Uni_{Heur}, $Pred_{1Heur}$ and $Pred_{2Heur}$ are the heuristic estimations for the first clause, the second clause, the link unifier, the predicate literal in the first clause of a link, and the predicate literal in the second clause of a link, respectively, and k_{uni}, k_{clause} and $k_{pred} \in [1; 100]$ are coefficients. Let us describe these symbols in more detail.

4.1.2 Heuristic Estimation of a Clause

The heuristic estimation must take into account the changes taking place in the characteristics of a connection graph during the inference (for example, changes in the number of links and the number of literals in a clause). Let us examine the heuristic estimation based on principles (1) and (2):

$$Clause_{Heur} = k_1 \cdot ClauseNumberOfLinks + k_2 \cdot ClauseNumberOfClauses,$$

where *ClauseNumberOfLinks* is the number of links in a clause, *ClauseNumberOfClauses* is the number of predicate literals in a clause; k_1, k_2 are arbitrary coefficients, chosen a priori on the basis of graph characteristics such as an average number of literals and links for the clause.

Let for example the average number of literals and links for the clause be 3.2 and 4.8, respectively. Then we can take $k_1 = 4.8/3.2 = 1.5$ and $k_2 = 1$. In this case, both principles (1) and (2) have the same weight. Characteristics may change their values during the inference. In this case, the initially chosen values of coefficients become "obsolete," and one principle gains a greater weight over the other. To avoid such a situation, the values of coefficients must be changed during the inference. Let *AverageLinkCount* and *AverageClauseLength* denote an average number of links and literals in a clause, respectively.

We can take $k_1 = AverageLinkCount/AverageClauseLength$ and $k_2 = 1$. In this case, principles (1) and (2) both gain the same weight in the course of the whole inference process. The heuristic clause estimation takes the final form:

$$Clause_{Heur} = (AverageLinkCount/AverageClauseLength) \times ClauseNumberOfLinks + ClauseNumberOfClauses.$$

4.1.3 Heuristic Estimation of the Unifier

The heuristic unifier estimation must be based on principle 4 (the unifier of a resolved link must have a substitution of the type *c/x*, where *c* is a constant or a functional term with constants and *x* is a variable) and must take into account the changes in values of the graph characteristics. The heuristic estimation of the unifier Uni_{Heur} is computed by the formula:

$$Uni_{heur} = AverageLinkCount/(1 + NumberOfConstantSubst + NumberOfFuncSubst),$$

where *NumberOfConstantsSubst* is the number of substitutions of the type *{c/x}* in the unifier, *c* is a constant term and *x* is a variable.

NumberOfFuncSubst is the number of substitutions of the type {*f/x*} m the unifier, where *f* is a functional term with constants and *x* is a variable.

4.1.4 Heuristic Estimation of a Predicate Literal

The heuristic estimation of a predicate literal must be based on principle (3) (the number of links in a literal, for which clauses are resolved upon must be minimal). The heuristic function must also take into account the changes in the graph characteristics. The heuristic estimation $Pred_{Heur}$ of a predicate literal is computed by the formula:

$$Pred_{Heur}=PredNumberOfLinks \cdot AverageClauseLength,$$

where *PredNumberOf Links* is the number of links in a predicate literal.

4.1.5 Selection of Coefficients

The coefficients k_{uni}, k_{clause} and k_{pred} are chosen experimentally. We have chosen the following ratio $k_{uni}=k_{pred}=(1/10)\cdot k_{clause}$ for coefficients. In this relationship, the greater weight is attached to principles (1) and (2). As a rule, k_{clause} is taken from the interval [10; 100] so that k_{uni} and k_{pred} lie in the interval [1; 10]. Thus, the heuristic function *H1* takes account of all principles (I)-(4). The weight of principles can be changed. The changes in the graph characteristics are also taken into account. The function *H1* enhances the efficiency of parallel inference algorithms for problems of practical complexity.

5. Deductive Inference on Clause Graphs

The deduction algorithm transforms a clause graph via two special operators – a predicate node elimination operator and a predicate node splitting operator [6,7]. They are applied to a predicate vertex depending on whether the node has multiarcs or not. A predicate node I said to be joined to a clause with multiarcs if the clause contains more than one literal with predicate symbol of the predicate node. For example, the node P in Fig.2 is joined to clause 1,2 and 3 (clause 4 is joined with the predicate node P with an ordinary arc) by the multiarcs.

1. $P(x,y) \rightarrow P(f(x),y)$ $(or \quad \neg P(x,y) \vee P(f(x),y))$
2. $P(u,f(x)) \,\&\, P(v,g(w)) \rightarrow$ $(or \quad \neg P(u,f(x)) \vee \neg P(v,g(w)))$
3. $\rightarrow P(g(x),y) \vee P(a,z)$ $(\text{or } P(g(x),y) \vee P(a,z))$
4. $\rightarrow P(a,x)$ $(or \quad P(a,x))$

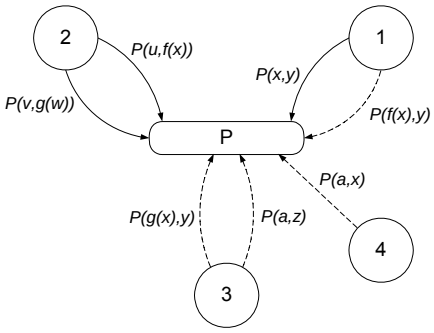


Fig. 2

In this method clauses are expressed in implicative form.

Here 1-4 are the nodes corresponding to the clauses of the logical model. The oval vertex represents the predicate symbol P. Continuous arcs are weighted with conditional literals of clauses, whereas dotted ars are weighted with inference literals of the clause.

An operation similar to colouring of clause graphs is introduced. Clause condition is «network is coloured» by color C1 (the corresponding arcs in figures are shown by a continuous line) and the condition for inference is colored by color C2 (a dotted line in figures). If clause graphs are represented in colored form, then it is easy to

search for information and new assertions can be inferred easily and thus the effectiveness of the inference system is enhanced.

The general inference scheme for an empty clause via transformation of clause graphs consists of the following:

1. if the network contains predicate nodes to which the node elimination operator can be applied, then such nodes are removed by this elimination operator;
2. if there are nodes with multiarcs, then the splitting operator is applied to generate nodes without multiarcs. Then the node elimination operator is applied, etc. until a contradiction is obtained in network.

If the node elimination operator consists of a set of usual operations of resolution of specially chosen pairs of clauses, then the splitting operator contains a distinct feature of this algorithm, i.e., a feature that has no analog in other logical inference mechanisms.

The predicate node elimination operator is applied to nodes having no multiarcs. It resolves in all possible ways those clauses that contain contrary pairs of literals. Resolution is implemented by the predicate of the chosen predicate node. Upon completion of all resolutions, parent clauses and the predicate node are removed from the network and the new clauses found via resolution are included in the network. Figures 3a and 3b show a clause graph before and after the application of the elimination operator to the vertex M for the following disjunct set:

$$1.\, Q \,\&\, M \rightarrow$$
$$2.\, M \,\&\, H \rightarrow Q$$
$$3.\, F \,\&\, M \rightarrow H$$
$$4. \quad \rightarrow M$$
$$5. \quad \rightarrow F$$

The node M and clauses (1)-(4) were removed and the clauses

$$6.\, Q \rightarrow . \qquad (1,4)$$
$$7.\, F \rightarrow . H \qquad (3,4)$$
$$8.\, H \rightarrow Q \qquad (2,4)$$

were added to the network.

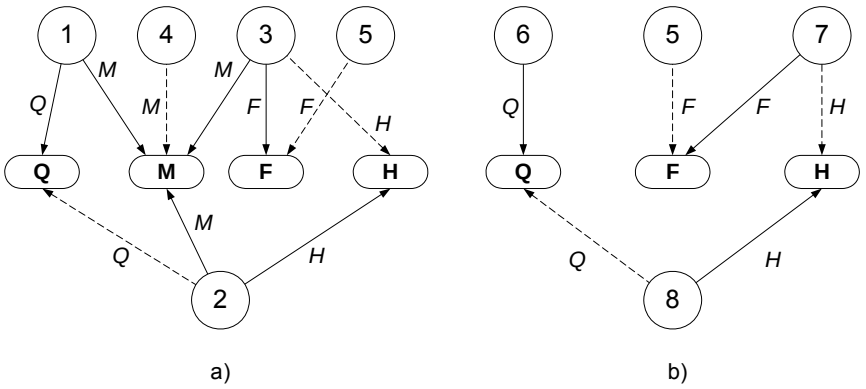


Fig. 3

The predicate splitting operator is introduced in order to remove multiarcs from nodes and thereby create conditions for the application of the elimination operator. The splitting operator generates copies of clauses and a few new vertices having the same predicate symbol as the split vertex, but with new indexes 1,2,3, … . The clause copies and new nodes are interrelated with each other. First, the elimination operator removes those nodes that have higher indexes.

Figure 4 shows the action of a splitting operator for the case of three multiarcs starting from clauses 1,2 and 3 (Fig. 2).

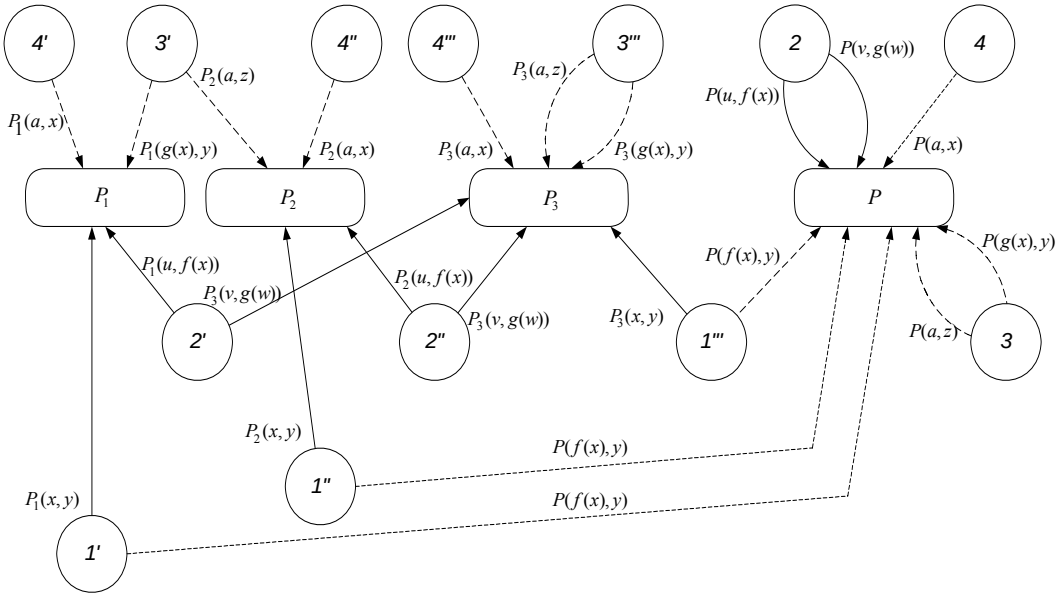


Fig.4

Splitting the three multiarcs ("petals") of the node P by this operator 1we obtain four nodes P_1, P_2, P_3 and P (their amount is one more than the amount of «petals»), four copies of clause 4, three copies of the «split» clause 1, two copies of the «split» clause 2 with a duplicate at vertex P, and a «split» clause 3 with a duplicate at nodes P and P_3. Splitting occurs in the following order: first the multiarc (1 – P), then the multiarc (2 – P), and finally the multiarc (3 – P) are split. The priority elimination strategies for removing nodes having no multiarcs and nodes with minimal number of arcs are reasonable for forming a sequence of processed nodes in a clause graph. They yield a sequence of eliminations and generate a lesser number of intermediate clauses than the variant in which these strategies are not used. Both these operations are correct i.e. unsatisfiable clauses remain unsatisfiable after the application of these operators[1]. A parallel algorithm for deductive inference on colored clause graphs is described in [7,8].

5. Efficiency

As the test problem, we have researched the "steamroller" problem formulated by L. Schubert in 1978 to test automatic proof systems [1]. This problem requires the generation of an exponential number of intermediate clauses and unifications during the inference process. Below we describe the results obtained by our and other algorithms of deductive inference for the «steamroller» problem.

procedures have also shown their efficiency for deductive inference on connection graphs.

CG is the inference strategy with a connection graph. SOS is the inference on a connection graph with a goal statement as a support set. TR is the inference within Theory Links - the extension of the standard resolution method. UR is inference with Unit Resolution - a modification of the resolution method. LUR is the inference by Linked Unit Resolution - a modification of the UR resolution method.

According to Figs. 5 and 6, the best results are obtained by the McCune/LUR procedures. Parallel inference

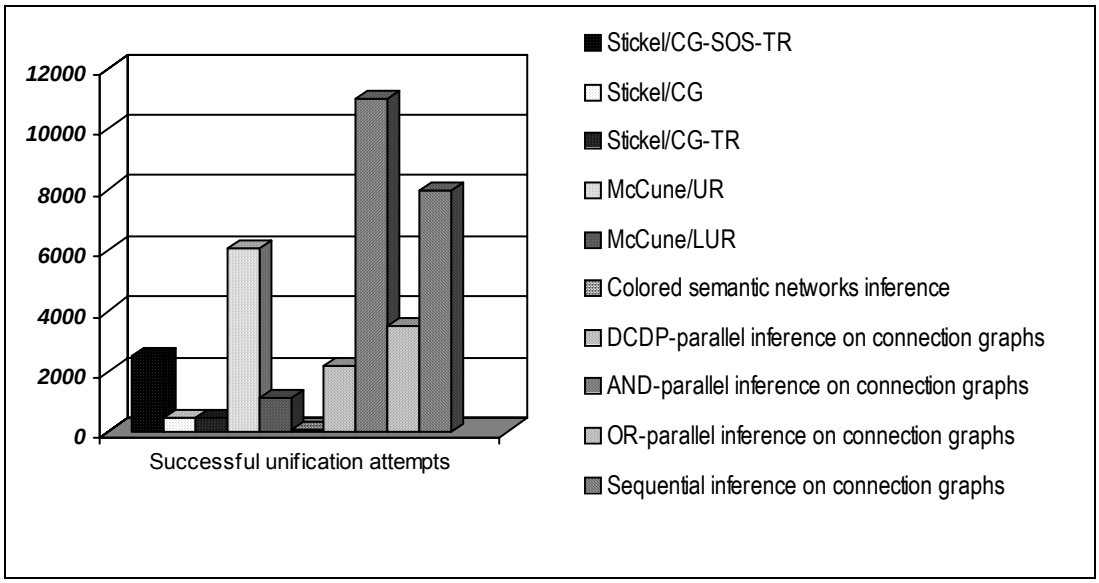


Fig. 5

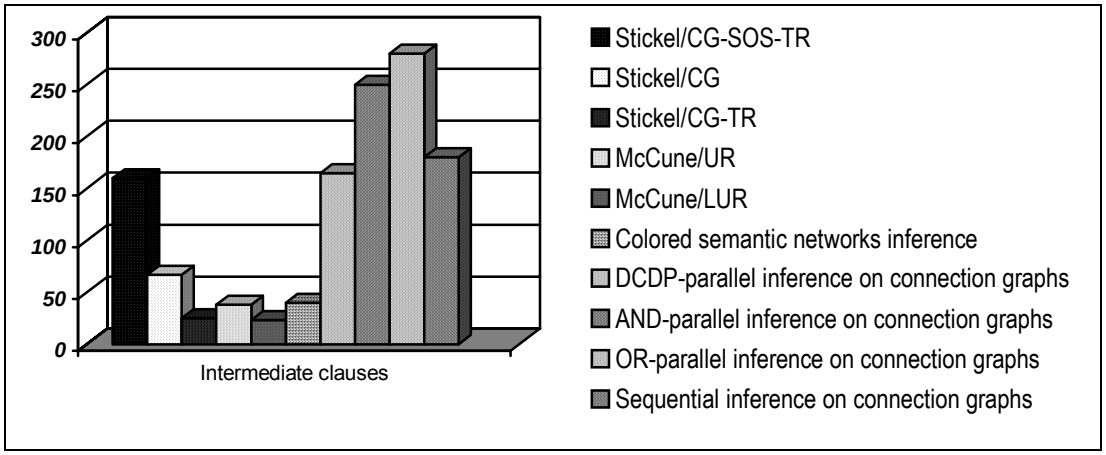


Fig. 6.

6. Parallel Unification in Connection Graph Inference

Due to a high amount of the unification tasks in deductive inference, the efficiency of the implementation of the unification procedure is one of the main factors in designing deductive inference procedures. There is a variety of effective unification procedures, but all of them have their own disadvantages. The main disadvantage is the use of complex, tightly coupled data structures, which are very difficult to parallelize.

We develop rather simple data structure for the term representation based on the notion of path strings. Terms (and clauses) are stored in tables, which are connected with links. Some tables and parts of the tables can be processed in parallel (with having links in mind). As we have no possibility to describe the algorithm and the data structure for term representation in detail, we just briefly outline the main principles used in this approach.

The term is stored as the sequence of strings, where every string presents one symbol in the term. Also such information as the type of a symbol (a variable or a constant symbol), the index number of an argument in a function symbol and the depth of a symbol (i.e. the number of function symbols which this symbol belongs to) is stored. The terms that can be unified are connected with the links of two types.

The first type corresponds to the terms (and strings) of different depth. The second type corresponds to the terms of the same depth. Let us illustrate these notions by the simple example.

Consider the unification task $\{t_1=t_2\}$, where t_1= *f(a,g(h(x)))* and t_2=*f(x,g(y))*. The representation of the term *f(a,g(h(x)))* is *f1.a.1.constant*(1) and *f2.g1.h1.x.3.variable* (2).

The representation of the term *f(x,g(y))* is *f1.x.1.variable*(3) and *f2.g1.x.2.variable*(4) (the index number is shown in brackets). The links are created between the strings (1) and (3) and between the strings (2) and (4). The first link has the second type and the second link has the first type.

The task of link establishing is one of the main tasks in the proposed approach. Two techniques can be used. The simplest one is the special sorting procedure on tables containing strings from the unification task. The main drawback of this approach is the deficit of efficiency caused by the nature of the sorting problem (the complexity of the sorting problem is *nlog(n)*). The second approach is based on automata representation of strings and is similar to discrimination trees. Using automata increases the efficiency of the procedure, but requires higher memory consumption.

The main idea lying in the parallelization of the unification procedure is the use of dependencies graph, where strings connected with arcs cannot be proceeded in parallel. The complexity of the task of determining maximum sets for parallel proceeding is equal to the complexity of the graph coloring task, though heuristics can be used to find non-maximum but satisfactory sets. This unification procedure has been combined with the parallel inference procedure on connection graphs. The structure for term representation and unification are thoroughly described in [9].

7. Conclusions

The Kowalski connection graph and the sequential inference procedure on connection graphs are investigated. The structure of the OR-, AND- and DCDP-parallel inference methods is described. Methods of modifying parallel inference procedures are analyzed and the main principles underlying the design of heuristic link estimations are stated. The heuristic function for link selection is designed. A procedure of inference on colored clause graphs is also described. The developed deductive inference procedures are compared with other procedures on the "steamroller" problem. The procedures of parallel inference on connection graphs and clause graphs are the effective ones for the deductive inference. They can be applied in expert and decision making systems.

Bibliography

[1] M.F. Stickel, Schubert's Steamroller Problem: Formulations and Solutions // *J. Autom. Reasoning,* 1986, vol. 2, pp. 89-100.

[2] R. Kowalski, A Proof Procedure using Connection Graphs // J. ACM, 1975, 22(4), pp. 595-599.

[3] V.N. Vagin and N.O. Salapina, A System of Parallel Inference with the Use of a Connection Graph // *Journal of Computer and System Sciences International,* Vol. 37, No. 5, 1998, pp. 699-708.

[4] G. Hornung, A. Knapp and U. Knapp, A Parallel Connection Graph Proof Procedure // *German Workshop on Artificial Intelligence. Lecture Notes in Computer Science,* Berlin: Springer-Verlag, 1981. pp. 160-167.

[5] R. Loganantharaj, Theoretical and Implemetational Aspects of Parallel Link Resolution in Connection Graphs, *Ph.D. Thesis,* Dept. of Computer Science, Colorado State Univ., 1985.

[6] Vagin V.N. Deduktsiya i obobshcenie v sistemakh prinyatia reshenii (Deduction and Generalization in Decision Making Systems), Moscow: Nauka, 1988.

[7] Vagin V.N., Parallel Inference on Logical Networks, IFIP/WG 12.3 International Workshop on Automated Reasoning, Beijing, China, 1992, pp. 305-310.

[8] A.I. Averin, V.N. Vagin and M.K. Khamidulov, The Investigation of Algorithms of Parallel Inference by the Steamroller Problem // *Journal of Computer and System Sciences International,* Vol. 39, No. 5, 2000, pp. 758-766.

[9] A.I. Averin, V.N. Vagin, Using Parallelism in Deductive Inference, // Journal of Computer and System Sciences International. -2004. - vol. 45, №4. – pp.603-615.

[10] Vagin V.N., Golovina E.Y., Zagoryanzkaya A.A., Fomina M.V.. «Dostoverniy i pravdopodobnyi vivod v intellektualnykh systemakh». (Relaible and Plausible Inference in Intellectual Systems) – Moscow: Fizmatlit. - 2004.

Authors' Information

Andrey Averin – Moscow Power Engineering Institute, Krasnokasarmennaya str., 14, Moscow, Russia; e-mail: averin@rbcmail.ru

Vadim Vagin – Moscow Power Engineering Institute, Krasnokasarmennaya str., 14, Moscow, Russia; e-mail: vagin@apmsun.mpei.ac.ru

APPROACHES TO SEQUENCE SIMILARITY REPRESENTATION

Artem Sokolov, Dmitri Rachkovskij

Abstract: We discuss several approaches to similarity preserving coding of symbol sequences and possible connections of their distributed versions to metric embeddings. Interpreting sequence representation methods with embeddings can help develop an approach to their analysis and may lead to discovering useful properties.

Keywords: sequence similarity, metric embeddings, distributed representations, neural networks

ACM Classification Keywords: I.2.6 Connectionism and neural nets, E.m Miscellaneous, G.2.3 Applications

Introduction and Background

In various applications, it is necessary to search for similar sequences of data. Examples include (but are not limited to) gene similarity search in biology, speech recognition, document database or Internet search, comparison of network traffic flows in computer security systems or detecting dangerous deviations from normal behavior of users by observing sequences of their actions.

There are many ways to formalize an intuitive notion of similarity ("looking the same") between strings[1], e.g., by edit distance. Given a set of edit operations, edit distance $L(s,t)$ between string s and t is the minimum number of edit operations needed to transform s into t. This definition benefits from being intuitive clear, simple to understand, and is often motivated by applications (e.g. evolutionary arguments in biology).

The simplest edit distance is the Hamming distance that counts the number of positions where strings differ – or, alternatively, the number of character changes needed to transform one string to another. Being simple (however, very useful in some applications, e.g. error correction), it is not sufficient for many real-world symbol sequences.

A more general definition is the Levenshtein distance [Levenshtein, 1965], where operations are symbol changes, insertions and deletions. Its exact value can be computed using dynamic programming algorithm in $O(n^2)$ time. More flexible definitions extend the set of possible operations with block copies, moves, indels etc.

In applications, sequence length can be very large, especially in Internet and networking applications and/or applications working with streaming data. Estimating their similarity requires fast algorithms, making time consumption of exact algorithms prohibitive. For example, finding Levenshtein distance in quadratic time is not fast enough. Finding minimum sequence of block edit operations is substantially harder – some versions of it were proved to be NP-hard [Lopresti, Tomkins, 1997].

As representations allowing for a simple (element-wise) definition of similarity or distance, consider vector representations of symbolic sequences. Let each element of a vector represent some item - e.g. some substring of the input string. Depending on the chosen sets of substrings, we obtain representations known by different names in literature. If items are all substrings of nearby symbols of length q and the vector contains their occurrence numbers/frequencies, they are known as "q-grams" [Ukkonen, 1992]. If the sequence is a text and items are single words (i.e., q=1), we get a common “bag-of-words” representation often used in Vector Space Models (VSM) for informational retrieval [Salton, 1989]. The latter case (q=1) does not take order of items into account, but this can be regulated by setting q>1.

Such vector representations can be linked with edit distance through an observation that when two strings s and t are within a small edit distance of each other, they share a large number of items. Vector representations of strings can be used for (weakly) approximating edit distances.

A way to solve the problem of fast finding similar sequences is to look for approximate solutions. One research area where such approximate solutions are sought is embedding techniques [Indyk, 2004]. They deal with

[1] We consider sequences composed of symbols from finite alphabet Σ (e.g., letters, commands), i.e. strings, as opposed to other types of sequences (e.g. speech, movements), where components are not so evident.

mapping complex data to some “easier” space, which allows finding or approximating distances faster. “Easy” spaces are usually vectors spaces; particularly interesting are Euclidean and Hamming ones. In those spaces, efficient algorithms are often available for a specific task (like nearest neighbor search) and/or computing similarity or distances between vectors can be done faster than in the original “complex” space.

The idea behind embeddings is that smaller parts of objects are often sufficient to approximate distances or similarity, provided objects are partitioned in sufficiently random manner. In a number of interesting cases, this happens because of the phenomenon known as concentration of measure. An excellent state-of-the-art review of embeddings of general metrics as well as of special metrics such as edit distances is given in [Indyk, 2004].

Another research area where similar problems are considered is distributed representations that try to capture brain’s way of representing complex objects [Thorpe, 2003; Arbib, 2003]. In distributed representations objects of various complexity – from elementary to structured ones – are sought to be represented by patterns of activity over pools of "neurons", which can be thought of as "codevectors" (e.g., [Rachkovskij, Kussul, 2001]). It is believed that brain uses similar representations for an efficient recall and comparison of complex internal representations of real-world objects.

It was commonly thought that distributed representations are suitable only for "bag-of" representations [Feldman, 1989; Malsburg, 1986], however, the introduction of the so-called binding operations changed the situation [Plate, 2003]. In contrast to the non-distributed representations described above, here similarity of items is put into correspondence with the degree of correlation of their codevectors. Another property is the possibility of composing "reduced representations" of complex objects from subsets of codevectors of their parts as well as reconstructing full representations of parts from a reduced representation of the whole. Recursive construction of reduced representation results in a codevector representing the whole complex object. Different possibilities of combining components and reducing their representations give potential for constructing representations that reflect some necessary application-specific notion of similarity. One can use e.g. dot products or whatever for measuring similarity of codevectors – which are usually made of the same dimensionality even for items of different complexity.

For sequences, goals for embeddings and distributed representations are similar at least in the following: both try to find such a vector representation of sequences that preserves necessary application-specific similarity and provides fast calculation of similarity or difference in the target space. In this paper, we describe some ways of introducing order information into distributed representations, including those of binary sparse type; and their connection to traditional sequence vector representations and embeddings. We show that some non-distributed and distributed sequence-processing methods can be related through random embeddings (particularly, random projections) and can be viewed in a coherent way. We think that connecting sequence representation methods with embedding theory can help develop approach to their analysis and discover useful properties.

Vector Representation of Strings by q-grams

In this section, we mention some of non-distributed representations that can be characterized as bag-of representation approaches and that allow some approximation of edit distance.

Consider a string s and a sliding window of q symbols on it. For each possible window content (q-gram) the number of its occurrences in the string is recorded in a corresponding coordinate of the vector $v_q(s)$ (q-gram vector). In case of q=1 it is just a vector with elements equal to the number of times a respective letter was met in a sequence, equivalent to frequency vector in VSM. As it was noted in [Ukkonen, 1992], each edit operation changes at most q q-grams, so if the edit distance is at most L, then

$$||v_q(s)-v_q(t)|| \le qL \qquad (2)$$

This method discards order of seen q-grams. Nevertheless, this gives a way for approximating edit distance from below: $L \ge ||v_q(s)-v_q(t)||/q$ and thus can be used for filtering. Given a query string for which it is necessary to find the nearest one from a set of strings, too distant strings can be filtered out using Manhattan distance between q-gram vectors. Developed further, the idea of q-grams can be used to define such notions of similarity as "resemblance" and "containment" of strings [Broder, 1997].

Another interesting application of q-grams is solving gapped edit distance problem [Bar-Yossef et al, 2004]. To solve a k vs. m gapped edit distance problem is to be able to answer, given strings s and t, whether the edit

distance between them is less than k or it is greater than m. In the algorithm, each string is divided into non-intersecting regions of some length D. Then each q-gram is accompanied with a fingerprint that is equal to the number of region it starts within, constituting a set of pairs (γ, i), where $\gamma \in \Sigma^q$. Pairs are considered equal only if their q-grams are equal and they appear in the same region. So, identical q-grams, but starting far enough from each other, receive different fingerprints and are different. Then Hamming distance is measured between vectors corresponding to the sets of such pairs (γ, i). It turns out that it is possible to solve k vs. $(kn)^{2/3}$ gapped edit distance problem by measuring Hamming distance between characteristic vectors of sets of fingerprinted q-grams. Second step in [Bar-Yossef et al, 2004] is to use dimensionality-reducing technique in Hamming space from [Kushilevitz et al., 1998].

Distributed Sequence Coding

Let us consider some ideas of representing strings with distributed representations. Since symbols are considered non-similar, they are assigned independent random binary codevectors. Here we consider non-hierarchical version of order representation, that are designed for representation of short sequences – to represent long strings it may be necessary to build representations in a hierarchical manner.

Distributed representation of strings by unordered substrings. We will consider a distributed version of q-gram representation from the previous section. Let us allocate a random vector for each of the sequence substrings and sum up (or take disjunction of) them to form a codevector of the sequence. Sequences containing many identical elements are likely to receive close codes. This method takes information about order in the sequence into account only to an extent captured by chosen substrings.

This method is somewhat similar to the Random Indexing (RI) and Random Labeling (RL) methods used in semantic processing of texts, which are versions of vector space methodologies for producing distributed words' representations using co-occurrence data [Kanerva et al., 2000; Karlgren, Sahlgren, 2001]. There a word can be abstractly viewed as a set of contexts it is used in. Each context is a bag of words (either a text where the word was met (RI) or a word's neighborhood (RL)). Each element of a context (either a text or a word) is assigned a random vector (with small number of +1 and -1). Context representation is a sum of those random vectors corresponding to constituent words. Target word representation forms by adding those context vectors each time this word appears in a corpus. As we will see in the following sections, this method can be interpreted straightforwardly with embeddings.

The following methods are trying to merge information about position of an item within a sequence with the representation of the item itself by modifying item's representation in a position-dependent manner. In each of following schemes vectors can be made binary (sums could be substituted by disjunctions), making dot product (a common similarity measure) computation efficient.

Positional binding with permutations. Consider the so-called shift coding which exemplifies an attempt to preserve information about ordering in a string. Vectors are modified by circularly shifting (or otherwise permuting) item's vector by the number of bits that depends on the position of the item. Code of the whole sequence is formed by disjunction of codes of all constituent items.

Consider e.g. strings '*abc*', '*abd*', '*cba*'. Each symbol is represented by a random vector: $v(a)$, $v(b)$, $v(c)$, etc. Then sequence codevectors are formed in this way:

$$v(abc) = (v(a) >> 1) \vee (v(b) >> 2) \vee (v(b) >> 3)$$

$$v(bca) = (v(b) >> 1) \vee (v(c) >> 2) \vee (v(a) >> 3),$$

where $\vee$ is bit disjunction, $X>>y$ means codevector X shifted by y bits.

The intersection between obtained vectors for strings with no identical items in the same positions will be (in expectation) negligible, provided the random vectors are sufficiently sparse and strings of short enough length. Tthis coding scheme discards information about identical symbols in different positions. However, partial permutation or shift may provide a way to fix that.

Positional binding with codevector. Here codevectors for positions are generated additionally to those for substrings. To code a substring in a particular position, the codevector of the substring and the codevector of its position are bound. In the binary case, binding is done by bit-wise conjunction and is called

thinning [Rachkovskij, Kussul, 2001]. Other types of binding are also possible, both for binary codevectors [Rachkovskij, Kussul, 2001; Kanerva, 1996] and codevectors with continuous elements [Plate, 2003].

If we encode substring as continuous codevectors and positions as binary codevectors, binding by element-wise product is possible (see also Gayler's multiplicative binding for real-valued codevectors in [Plate, 2003]), which in this case can be also considered as thinning. Thinning does not remove similarity of identical elements in different positions, as the shift method does. Representations of positions may be correlated for nearby ordinal numbers. Codevectors of symbol-position bindings are then combined by bit-wise disjunction or addition.

Binding an element with its context. Another option is to bind item's vector with vectors of item's from its context. This may give way to build position-independent representation of items, where item's contribution depends only on its context and does not depend directly on the position in a sequence. As edit metrics is usually also position-independent in the sense of counting edit operations independently of the place they were applied; this may help to approximate them. Note an analogy with taking into account contexts by q-grams.

Embeddings and Representational Economy

In this section we discuss some results for embedding vector and sequence distances. Target spaces for embedding are usually vectors spaces like l_p (where $||x||_p=(\sum_{i=1,\dots,d} x_i^p)^{1/p}$), particularly interesting are Euclidean spaces (p=2), l_1 with Manhattan metrics ($||x||_1=\sum_{i=1,\dots,d}|x_i|$) or the Hamming space ($\rho(x,y)=\sum_{i=1,\dots,d}[x_i \neq y_i]$).

The seminal result that we will use is the Johnson-Lindenstrauss reduction lemma [Johnson, Lindenstrauss, 1984]. Let the elements of matrix $R_{d' \times d}$ be from $N(0,1)$ distribution. Let vectors $v'=Rv$. Then for every $\varepsilon>0$ and any two vectors $v_1, v_2 \in l^d_2$:

$$(1-\varepsilon)||v_1-v_2||_2 \le ||v'_1-v'_2||_2 \le (1+\varepsilon)||v_1-v_2||_2 \qquad (3)$$

holds with probability $\exp(\Omega(-d'/\varepsilon^2))$. In case of normal distribution of the vectors embedding into normed space is due to 2-stability of the normal distribution. There exist embeddings of norms for l_p using other p-stable distributions for $0<p\le 2$ (see [Cormode et al, 2002] for a brief overview). Thus it is possible to logarithmically reduce the input dimension while distorting mutual distance not too much. Note that instead of vector elements distributed according to normal distribution we can use either binary {-1,1} or ternary {-1,0,1} elements (with proper distribution) as proven in [Achlioptas, 2001].

Despite the progress in embedding "usual" metrics, embedding Levenshtein distance is a long-standing problem [Indyk, 2001], and a negative result was proved recently that any such algorithm would not have distortion smaller than 3/2 [Andoni et al., 2003]. However, it is possible to embed a relaxed version of edit distance (with block moves) to l_1 with approximately logarithmic distortion $\tilde{O}(log(n))$ by carefully selecting substrings, which add to the characteristic vector of the sequence [Cormode et al, 2000]. This result is particularly interesting because the exact calculation of this distance is NP-hard.

Connection between Distributed and Non-distributed Sequence Processing

In this subsection, we show how to interpret some distributed sequence coding methods with the help of embedding theory. We consider continuous case, so elements of the used vectors are from R and operations are usual summation and multiplication; as well as binary case, where operations are Boolean OR and AND.

Distributed representation of strings by unordered substrings According to it, a codevector $v(s)$ for a string $s=s_1,\dots,s_n$ is defined as $v(s)=\Sigma_{i=1,\dots,n} r(s_i)$, where $r(s)$ is a random codevector corresponding to an item s. The expression for $v(s)$ can be rewritten as

$$v(s) = \sum_{i=1,\dots,n} r(s_i) = \sum_{\sigma \in \Sigma} \sum_{i=1,\dots,n} I[s_i=\sigma] = \sum_{\sigma \in \Sigma} r(\sigma)\, n_s(\sigma), \qquad (4)$$

where $n_s(\sigma)$ is the number of times σ occurs in s.

Expression (3) is the same as projecting a bag-of-items vector $n_s^T(\sigma)=(n_1(\sigma),n_2(\sigma),\dots,n_{|\Sigma|}(\sigma))$ by multiplying it by a random matrix ($d\times|\Sigma|$) with columns $r(\sigma)$. Thus, this coding can be viewed as mapping from (e.g., VSM) space of dimension $|\Sigma|$ to R^d, where d can be made considerably lower then the number of all possible items $|\Sigma|$.

If both spaces are Euclidean, then, applying JL-lemma (1) and taking $r(s)$ from normal distribution (or from certain binary or ternary distributions, see [Achlioptas, 2001]) we obtain, that for a desired distortion $0<\varepsilon<1$, it is possible

to reduce dimension, while keeping $||v(s)-v(t)||$ within the range $(1\pm\varepsilon)||v_q(s)-v_q(t)||$ with high probability. This is what is done in RI (however, with sparse codevectors) appealing to "near orthogonality" of random vectors in high dimensions (but, in the same time, dimension can be considerably lower than the number of all contexts). And so using inequality (2), for the two strings within edit distance less then L the Euclidean distance between corresponding codevectors is no more then $(1+\varepsilon)qL$. This provides an upper bound for distance between vectors, and can be used for filtering purposes using representation vectors of low dimension instead of large (however, sparse) q-gram vectors.

There are experimental evidences that taking into account only information about presence of words in texts, it is possible to preserve similarity of texts to an extent sufficient for some applications [Grossman, Frieder, 1998].

Connection of the thinning coding to random sampling In this subsection we try to establish an analogy between thinning coding and some of the edit distance approximation approaches. Consider the i's element of output vector $v(s)$:

$$v_i(s) = \sum_{k=1,\dots,n} c_{ki}\, r_{i\,s[k]} = \sum_{k=1,\dots,n} c_{ki} \sum_{\sigma\in\Sigma} r_{i\sigma}\, I[s_k=\sigma] = \sum_{\sigma\in\Sigma}\sum_{k=1,\dots,n} r_{i\sigma}\, I_{\sigma k}\, c_{ki}. \quad (4)$$

Then we can define $V(s)=R\cdot L(s)\cdot C$, where columns of matrix R are random codevectors r_i assigned to the i'-th symbol of Σ and rows of C are position codevectors C_i.. Elements of the indicator matrix L are: $l_{\sigma k}=1$ if the k'-th substring (or symbol if $q=1$) of string s is σ and $l_{\sigma k}=0$ otherwise. That is, the element of the matrix shows where symbols from the alphabet (symbols or substrings) in the string are situated.

Consider the product $X=L(s)\cdot C$. If matrix C had contained only 1's in each of its cells, then this product would have just given columns of vector representation (e.g., q-gram) for the string. But, as C does contain zeros, substring in not all of the positions is counted into the vector. So, columns of the resulting matrix X would become an "incomplete" vector representations (q-gram representations, if Σ is all q-grams) of the input string s. Columns of matrix C act as binary masks marking positions in s, from where symbols sum up to a particular q-gram vector, e.g., the j's column of C masks s to obtain a vector with the i's element equal to $\sum_{k=1,\dots,n} l_{\sigma k} c_{ki}$. So, while earlier q-gram representations discarded order information in a sequence, here we introduced it with the help of position vectors.

Above, it was noted that position codevectors (rows of C) can be made so that nearby ones have small Hamming distance of each other and this distance grows as the distance between positions grows. Possible ways of constructing such codevectors from for ordinal numbers are considered in [Rachkovskij et al, 2005] and can also be implemented with a 2-state Markov chain with proper transition probabilities.

Different columns act as independent samplers. We note that similar approach has already led to even sublinear approximation of edit distance in [Batu et al, 2004]. Their approach is similar to ours in that the approximation is also achieved by randomly sampling input string and using the mentioned observation that strings within small edit distance have many substrings in nearby positions.

Consider now the effect of multiplying X by matrix R. Each i-th element of the output vector $v(s)$ is a dot product of the random codevector r_i by the corresponding "thinned" q-gram vector. We already saw such an operation (projection on a random direction) when established analogy between unordered distributed encoding of strings by assigning random codevectors to their substrings (q-grams). The difference is that there each of the q-grams was projected to all random directions, but here different thinned q-gram vectors are projected along different directions. However, if we look closer, we note that because of intersections between columns of C "common parts" of masked (thinned) q-gram vector may undergo projection to different directions. From expression (4):

$$v_i(s) = \sum_{\sigma\in\Sigma}\sum_{k=1,\dots,n} (c_{ki}\, r_{i\sigma}) l_{\sigma k} = \sum_{k=1,\dots,n} R(C(k))\cdot l_k. \quad (4)$$

$R(C(k))$ is matrix R with only those rows left that correspond to those position codevectors that have 1s in position k. So, we see that each vector l_k undergoes projection to a respective random subspace, defined by those C_j that cover position k. The farther are identical symbols (or q-grams) from each other, the less common random directions they have, and so, the more distant are their projections.

Conclusions

We believe that enriching distributed representations with ideas and methods of analysis from embeddings can provide a more formal interpretation of distributed methods usually introduced in an ad-hoc manner. This may help infer the similarity type they approximate or find modifications that will allow them to approximate some, and help obtain bounds on the distortion of the proposed coding schemes, their complexity and limitations.

Here we have described approach to analysis of only few distributed schemes for coding sequences. We hope it could be extended to other schemes for encoding sequences mentioned above, as well as to schemes for distributed encoding of other types of data like numerical [Rachkovskij et al, 2005] or complex relational structures [Rachkovskij, 2004]. Those schemes include binding by context-dependent thinning and hierarchical representations [Rachkovskij, Kussul 2001].

Bibliography

[Achlioptas, 2001] D. Achlioptas, Database-friendly random projections. In Proceedings of PODS-01, pp. 274-281, 2001.

[Andoni et al., 2003] A. Andoni, M. Deza, A. Gupta, P. Indyk, S. Raskhodnikova, Lower Bounds for Embedding of Edit Distance into Normed Spaces. In Proc. of the 14th Symposium on Discrete Algorithms, 2003

[Arbib, 2003] M. Arbib, The Handbook of Brain Theory and Neural Networks. – Cambridge, MA: The MIT Press, 2003

[Bar-Yossef et al, 2004] Z. Bar-Yossef, T.S. Jayram, R. Krauthgamer, R. Kumar: Approximating Edit Distance Efficiently. FOCS, pp. 550-559, 2004

[Batu et al, 2004] T. Batu, F. Ergun, J. Kilian, A. Magen, S. Raskhodnikova, R. Rubinfeld, R. Sami. A sublinear algorithm for weakly approximating edit distance, In Proc. 36th STOC, 2004

[Broder, 1997] A. Z. Broder. On the resemblance and containment of texts. In Proceedings of Compression and Complexity of SEQUENCES, 1997

[Cormode et al, 2000] G. Cormode, M. Paterson, S. C. Sahinalp, U. Vishkin. Communication complexity of text exchange. In Proc. of the 11th ACM-SIAM Annual Symposium on Discrete Algorithms, pp. 197--206, San Francisco, CA, 2000

[Cormode et al, 2002] G. Cormode, P. Indyk, N. Koudas, S. Muthukrishnan. Fast mining of tabular data via approximate distance computations. In Proc. of the International Conference on Data Engineering, pp. 605–616, 2002

[Feldman, 1989] J. Feldman, Neural Representation of Conceptual Knowledge. In L. Nadel, L. Cooper, P. Culicover et al. Neural Connections, mental computation - London, England: The MIT Press, pp. 68-103, 1989

[Grossman, Frieder, 1998] D.A. Grossman, O. Frieder, Information Retrieval: Algorithms and Heuristics, Kluwer, 1998

[Indyk, 2001] P. Indyk, Algorithmic Aspects of Geometric Embeddings, FOCS, 2001

[Indyk, 2004] P. Indyk, Embedded Stringology, talk at the Symposium on Combinatorial Pattern Matching 2004. Available at http://theory.lcs.mit.edu/~indyk/cpm.ps

[Johnson, Lindenstrauss, 1984] W. Johnson, J. Lindenstrauss, Extensions of Lipschitz maps into a Hilbert space, Contemp. Math., 26 , pp. 189-206, 1984

[Kanerva et al, 2000] P. Kanerva, J. Kristofersson, A. Holst. Random indexing of text samples for latent semantic analysis. In Proc. of the 22nd Annual Conference of the Cognitive Science Society, p. 1036. Erlbaum, 2000

[Kushilevitz et al., 1998] E. Kushilevitz, R. Ostrovsky, Y. Rabani. Efficient Search for Approximate Nearest Neighbor in High Dimensional Spaces. STOC, pp. 614-623, 1998

[Levenshtein, 1965] V.I. Levenshtein. Binary codes capable of correcting deletions, insertions and reversals, Doklady Akademii Nauk SSSR, 163(4):845-848, 1965 (Russian)

[Lopresti, Tomkins, 1997] D. P. Lopresti, A. Tomkins, Block Edit Models for Approximate String Matching. Theoretical Computer Science, 181(1): 159-179, 1997

[Malsburg, 1986] C. von der Malsburg. Am I Thinking Assemblies? In Proc. of the Trieste Meeting on Brain Theory, pp.161-176, 1986

[Plate, 2003] T. Plate, Holographic Reduced Representation: Distributed Representation for Cognitive Structures. - Chicago: Center for the Study of Language and Information, 2003

[Rachkovskij et al, 2005] D.A. Rachkovskij, S.V. Slipchenko, E.M. Kussul , T.N. Baidyk, Sparse binary distributed encoding of scalar values, (to be published) 2005

[Rachkovskij, 2004] D.A. Rachkovskij, Some approaches to analogical mapping with structure sensitive distributed representations. Journal of Experimental and Theoretical Artificial Intelligence,16, №3, pp.125-145, 2004

[Rachkovskij, Kussul, 2001] D.A. Rachkovskij, E.M. Kussul, Binding and Normalization of Binary Sparse Distributed Representations by Context-Dependent Thinning. Neural Computation, 2, №13, pp.411-452, 2001

[Salton, 1989] G. Salton, G. Automatic Text Processing: The Transformation, Analysis, and Retrieval of Information by Computer, Addison-Wesley, Reading, MA., 1989

[Thorpe, 2003] S. Thorpe, Localized Versus Distributed Representations. In Arbib M. The Handbook of Brain Theory and Neural Networks - Cambridge, MA: MIT Press, pp. 643-646, 2003

[Ukkonen, 1992] E. Ukkonen. Approximate string-matching with q-grams and maximal matches. Theoretical Computer Science, 92:191-211, 1992

Authors' Information

Artem M. Sokolov, Dmitri A. Rachkovskij – International Research and Training Center of Information Technologies and Systems; Pr. Acad. Glushkova, 40, Kiev, 03680, Ukraine; e-mails: sokolov@ukr.net, dar@infrm.kiev.ua

APPLICATION OF THE MULTIVARIATE PREDICTION METHOD TO TIME SERIES [1]

Tatyana Stupina, Gennady Lbov

Abstract: An approach to solving the problem of heterogeneous multivariate time series analysis with respect to the sample size is considered in this paper. The criterion of prediction multivariate heterogeneous variable is used in this approach. For the fixed complexities of probability distribution and logical decision function class the properties of this criterion are presented.

Keywords: the prediction of multivariate heterogeneous variable, multivariate time series, the complexity of distribution.

ACM Classification Keywords: G.3 Probability and Statistics: Time series analysis

Introduction

Let certain object (process) is described by the set of random features $X = X_1,...,X_n$, changing on time. On the base of analysis information, that presents features measurements in the consequent moments time series (prehistory), it is necessary to predict a values of features set $Y = Y_1,...,Y_m$ at certain future time moment (in particular, $Y \subseteq X$. Distinguishing feature of considered below prediction problems is the measured features heterogeneity: the variable set be able consist of binary, nominal and quantitative variables simultaneously. In this case, multivariate time series presents itself a set of binary, symbol and numeric random sequences. Classical methods are directed to the analysis of numeric sequences basically. Many methods allow analyse univariate binary or symbol sequences. However the most of important applied problems number are concerned with need to heterogeneous time series analyse. There is reason to suppose in some problems that time series is the realization of random processes, in which probabilistic characteristics (distribution) are saved on a time. At other times such suggestions to do it is impossible under the matter of problem (probabilistic characteristics of process are changed on time). There is possible to offer a different depending on specified suggestions targets setting and the different methods of their decision accordingly. The methods of heterogeneous time series analysis for different targets setting, including the logical deciding functions class for heterogeneous variable are considered in work [Lbov G.S., 1994].

[1] This work was financially supported by RFBR-04-01-00858

The Target Setting

One is considered the n – measured heterogeneity random process $G = \{X_1(t),...,X_j(t),...,X_n(t)\}$. Let it set of predictable characteristic is $Y_j = X_j$, $j = 1,...,n$. Fix some consequent moments of the time, $1 \le R \le N$. Denote the value random variable X_j at a moment of the time t_d, $x_j^d \in D_{X_j}$, as this x_j^d, and x^d is the value random variable of X, $x^d \in D_X$, $D_X = \prod_{j=1}^{n} D_{X_j}$. The problem consist of that, it is necessary to predict the values set $y = (y_1,...,y_j,...,y_n)$ at certain future moment of the time t_{R+1}, where $y_j = x_j^{R+1}$ using the data, characterizing prehistory, $b = \{x_j^d\}$, $j = 1,..,n$, $d = 1,...,R$. It is necessary to build decision function, allowing predict a set of values $y = (y_1,...,y_j,...,y_n)$ on prehistory b.

The set of every possible all prehistory, that have line measure R denote as B, and the set of every possible all sets y denote as D_Y, $b \in B$, $y \in D_Y$, $D_Y = \prod_{j=1}^{n} D_{Y_j}$. Let us understand a prediction decision function as a f mapping of the B set on the D_Y set, i.e. $f : B \to D_Y$. At the building decision functions f is used following hypothesis: It is supposed that conditional distribution $P(y/b)$ does not depend on the shift on the time, i.e. distribution is specified for moments of the time $t_1,...,t_R,t_{R+1}$ is contemporized with distribution for moments of the time $t_1 \pm \Delta T,...,t_R \pm \Delta T,t_{R+1} \pm \Delta T$. If the conditional distribution $P(y/b)$ is known, then it is possible to find optimum prediction decision function $f_\circ$. Since specified distribution is unknown, decision function shall be constructed on the base of multivariate time series analysis.

Let the features $X_1,...,X_j,...,X_n$ are measured at consequent moments of the time with the gap $\Delta t = t_d - t_{d-1}$ for the random process G. Denote this set of moments as $T = \{t_1,...,t_k,...,t_N\}$. Thus, the empirical information is presented by n – measured heterogeneity time series $q = \{x_j^k\}, j = 1,...,n, k = 1,...,N$. The set of values $x^{k-d} = (x_1^{k-d},...,x_j^{k-d},...,x_n^{k-d})$ will is called prehistory with the number d, correlated with a moment of the time t_k, $k = R+1,...,N$. The prehistory with line measure R for a specified moment of the time t_k is denoted as a table $b^k = \{x^{k-d}\}$, $d = 1,...,R$. Note that univariate symbol sequence for $R = 1$ is the realization of simple Markoff process with the transfer probability matrix $P(y/x)$, $x \in A$, $y \in A$, A –an alphabet of symbols. Decision function $\bar{f}$, constructed on the base of set prehistory analysis with line measure R, is named sample decision function of prediction.

It is necessary to construct the sample decision function on the small sample in the multivariate heterogeneous space, so the most proper class is a class of logical decision functions [Lbov G.S., Starceva N.G, 1999]. Methods of time series analysis propose to decision of problem in two stages: It is constructing decision function for fixed prehistory with the number d ($d = 1,...,R$) it is constructing the generalise logical decision function (mapping $f : B \to D_Y$). The first stage is consist of decision the prediction multivariate variable problem Y on other multivariate variable X, i. e. for each prehistory d we have two data tables $\{x^{k-d}\}$, $\{y^k\}$, $k = R+1,...,N$, on base which necessary to construct the sample decision function (mapping $D_X \to D_Y$). Below it is considered a decision of this problem, in which is used criterion, introduced in work [Lbov G.S., Stupina T.A., 2002].

The Performance Criterion of Prediction

In the probabilistic statement of the problem, the value *(x,y)* is a realization of a multidimensional random variable *(X,Y)* on a probability space $< \Omega, \mathrm{B}, \mathrm{P} >$, where $\Omega = D_X \times D_Y$ is μ-measurable set (by Lebeg), B is the borel σ-algebra of subsets of Ω, P is the probability measure (probability distribution) on B, D_X is heterogeneous domain of under review variable, $\dim D_x = n$, D_y is heterogeneous domain of objective variable, $\dim D_y = m$.

Definition 1. The strategy of nature is *c={p(x,y)=p(x)p(y/x))}*, where a conditional probability *p(y/x)* is specified for any elements on B.

Let us put $\Phi_\circ$ is a given class of decision functions. Class $\Phi_\circ$ is μ-measurable functions that puts some subset of the objective variable $E_y \subseteq D_Y$ to each value of the under review variable $x \in D_X$, i.e. $\Phi_\circ = \{f : D_x \to 2^{D_y}\}$. This class of decision function is more total than class of logical decision functions [Lbov G.S., Starceva N.G, 1999]. In this paper, we will consider criterion for decision function from total class $\Phi_\circ$. So criterion was considered for logical decision functions in work [Lbov G.S., Stupina T.A., 2002]. But here we will achieve that class of logical decision functions is a universal class about relative to criterion.

The quality $F(c,f)$ of a decision function $f \in \Phi_\circ$ under a fixed strategy of nature *c* is determined as follows.

$$F(c,f) = \int_{D_x} (P(E_y(x)/x) - \mu(E_y(x)))dP(x),$$

where $E_y(x) = f(x)$ is a value of decision functions in *x*, $P(y \in E_y(x)/x)$ is a conditional probability of event $\{y \in E_y\}$ under a fixed *x*, $\mu(E_y(x))$ is measurable of subset E_y. Note that if $\mu(E_y(x))$ is probability measure, than criterion $F(c,f)$ is distance apart distributions. If the specified probability coincides with equal distribution than such prediction does not give no information on predicted variable (entropy is maximum). The measure $\mu(E_y(x)) = \dfrac{\mu(E_y)}{\mu(D_Y)} = \prod_{j=1}^{m} \dfrac{\mu(E_{y_j})}{\mu(D_{y_j})}$ is the normalized measure of the subset E_y and it is introduced with taking into account the type of the variable. The measure $\mu(E_y(x))$ is measure of interval, if we have a variable with ordered set of values and it is quantum of set, if we have a nominal variable (it is variable with finite non-ordering set of values). Clearly, the prediction quality is higher for those E_y whose measure is smaller (accuracy is higher) and the conditional probability $P(y \in E_y(x)/x)$ (certainty) is larger.

For a fixed strategy of nature *c*, we define an optimal decision function $f_\circ(x)$ as function for which $F(c,f_\circ) = \sup_{f \in \Phi_\circ} F(c,f)$, where $\Phi_\circ$ is represented above class of decision functions.

As a rule, the strategy of nature is unknown; for this reason, a decision function is constructed from a training sample $v = (x^i, y^i)_{i=1,\ldots,N}$ by sampling criterion $F(\bar{f})$ with the use of some algorithm $Q(v) = \bar{f}$, where $\bar{f}(x)$ is a sampling decision function and *N* is the size of the training sample. The sampling criterion $F(\bar{f})$ is empirical risk of the criterion $F(c,f)$.

When we solve this problem in practice the size of sample is very smaller and type of variables different. In this case is used class of logical decision function. The logical decision function *f* is assigned the pair $<\alpha,\beta>$, where $\alpha \in \Psi_M$ and $\beta \in R_M$. The class Ψ_M is the set of partitions $\alpha = \{E_x^1,\ldots,E_x^t,\ldots,E_x^M\}$ of the space D_X into disjoint subsets for which $E_x^t = \prod_{i=1}^{n} E_{x_i}^t$, $E_{x_i}^t \subseteq D_{X_i}$, $E_{x_i}^t \neq \varnothing$ and $E_{x_i}^t \in W_{X_i}$, where W_{X_i} is the set of all possible intervals if X_i is a variable with ordered set of values and W_{X_i} is the set of arbitrary subsets of D_{X_i} if X_i is a nominal variable, i.e. a variable with a finite unordered set of values; we have $E_x^t \in W_X$, where $W_X = \prod_{i=1}^{n} W_{X_i}$.

The class R_M is the set of decisions (arbitrary subset of the space D_Y) $\beta = \{E_y^1,\ldots,E_y^t,\ldots,E_y^M\}$ for which $E_y^t = \prod_{i=1}^{m} E_{y_i}^t$, $E_{y_i}^t \subseteq D_{Y_i}$, $E_{y_i}^t \neq \varnothing$ and $E_{y_i}^t \neq \varnothing$, where W_{Y_i} is defined so as W_{X_i}. The decision function is presented in simple form for understanding: *if* $x \in E_x^t$ *than* $y \in E_y^t$. The subsets E_x^t and E_y^t represented as above can be described in terms of conjunctions of simple predicates. Such a coarsening of the decision function

is caused by the necessity to construct solutions from small samples. The class of logical decision function Φ_M can be represented as $\Psi_M \times R_M$.

Under the assumptions made, the complexity of the class Φ_M is only determined by the *M* parameter: $v(\Phi_M) = M$. Thus, the larger the number *M*, the more complex the class Φ_M. We achieve important property of this class by theorem.

Theorem. For a fixed type of the predicate, the class Φ_M of logic decision functions is a universal class in the problem of prediction multivariate heterogeneous value by criterion *F(c,f)*, i.e. for any strategy of nature *c* and any $\varepsilon > 0$ there exists a number *M (M=1,2,3,…)* and for some logical decision function $f \in \Phi_M$ (it is represented in the form of decision tree on *M* vertices) such that $|F(c,f) - F(c,f_\circ)| \le \varepsilon$, where $f_\circ$ is optimal function in class $\Phi_\circ$.

The proof of this theorem readily follows from the property of μ-measurability and P-measurability of space *D* and its projections on the space D_X, D_Y correspondingly.

The proof for the case where *Y* is a discrete variable is given in [Lbov G.S., Starceva N.G, 1994]. The proof for the case where *Y* is a continuous variable is given in [Berikov V.,1995].

We can introduce a complexity of distribution (strategy of nature *c*) using the class logical decision function. It is necessary for solving statistical stability problem of decision function.

Statement 1. For any nature strategy *c* the quality criterion $F(c,f)$ (risk function) of logical decision function *f* belonging to Φ_M is presented by following expression:

$$F(c,f) = \int_{D_X}\int_{D_Y}(1 - L(y,f(x)))p(x,y)dxdy = \sum_{t=1}^{M} p_x^t (p_{y/x}^t - \mu^t),$$

where the loss function $L(y,f)$ such as $L(y,f) = \begin{cases} p_o & y \in \beta \\ 1 + p_o, & y \notin \beta \end{cases}$, $p_o = \mu(E_Y^t)$, $\beta = f(\alpha), \alpha \in \Psi_M$.

Proof. $F(c,f) = \int_{D_x}(P(E_y(x)/x) - \mu(E_y(x)))dP(x) = \sum_{t=1}^{M}\left[\int_{E_X^t}\int_{E_Y^t} p(x,y)dxdy - p_o \int_{E_X^t} p(x)dx\right] =$

$$\sum_{t=1}^{M}\left[\int_{E_X^t}\int_{E_Y^t} p(x,y)dxdy + \int_{E_X^t}\int_{D_Y}(-p_o)p(x,y)dxdy\right] =$$

$$\sum_{t=1}^{M}\left[\int_{E_X^t}\int_{E_Y^t}(1-p_o)p(x,y)dxdy + \int_{E_X^t}\int_{D_Y}(-p_o)p(x,y)dxdy - \int_{E_X^t}\int_{E_Y^t}(-p_o)p(x,y)dxdy\right] =$$

$$\sum_{t=1}^{M}\int_{E_X^t}\left[\int_{E_Y^t}(1-p_o)p(x,y)dxdy + \int_{\bar{E}_Y^t}(-p_o)p(x,y)dxdy\right] = \int_{D_X}\int_{D_Y}(1 - L(y,f(x)))p(x,y)dxdy.$$

Definition 2. To each subclass Φ_M we put in correspondence the subset $L_\varepsilon(M) = \{c : \exists f \in \Phi_M, |F(c,f) - F(c,f_\circ)| \le \varepsilon\}$ of nature strategies; ε is an arbitrarily small number determining an admissible error level of this subset of strategies, where $f_\circ$ is optimal function in class $\Phi_\circ$.

The complexity measure of each subset $L_\varepsilon(M)$ is defined as the complexity measure of the corresponding subclass of decision functions: $v(L_\varepsilon(M)) = v(\Phi_M) = M$. Accordingly, the nature strategy *c* belonging to $L_\varepsilon(M)$ has complexity measure *M*. The important statement follows from this theorem and definition.

Statement 2. The set of all possible strategies can be ordered according to complexity, i.e. $L_\varepsilon(1) \subset L_\varepsilon(2) \subset \ldots \subset L_\varepsilon(M) \subset \ldots \subset L_\circ$, and $\varepsilon^{M+1} \le \varepsilon^M$, where $v(L_\varepsilon(M)) = M$ is the complexity and ε^M is the admissible error level of the strategy class $v(L_\varepsilon(M))$.

Proof. For an arbitrary *M*, let us prove the embedding $L_\varepsilon(M) \subset L_\varepsilon(M+1)$ i.e. show that $\forall c \in L_\varepsilon(M)$, $\exists f \in \Phi_{M+1}$ such that $\left|F(c,f) - F(c,f_\circ)\right| \le \varepsilon$. The definition of the class $L_\varepsilon(M)$ implies that $\exists g \in \Phi_M$ such that $\left|F(c,g) - F(c,f_\circ)\right| \le \varepsilon^M$. Since $\Phi_M \subset \Phi_{M+1}$, we can obtain *f* from *g* by partitioning some subset E_X^t into two subsets: if $g \sim <\alpha,\beta>$, $\alpha = \{E_X^t\}_{t=1,...,M}$, $\beta = \{E_Y^t\}_{t=1,...,M}$ than $f \sim <\alpha',\beta'>$, $\alpha' = \{E_X^1,...,E_X^{t_1},E_X^{t_2},...,E_X^M / E_X^t = E_X^{t_1} \cup E_X^{t_2}\}$, $\beta' = \{E_Y^1,...,E_Y^{t_1},E_Y^{t_2},...,E_Y^M / E_Y^t = E_Y^{t_1} \cup E_Y^{t_2}\}$, where $\mu(E_X^t) = \mu(E_X^{t_1}) + \mu(E_X^{t_2})$ and $\mu(E_Y^t) \ge \mu(E_Y^{t_1}) + \mu(E_Y^{t_2})$. Therefore, $\left|F(c,f) - F(c,f_\circ)\right| \le \varepsilon = \varepsilon^{M+1} \le \varepsilon^M$, it is followed from the definition $F(c,f)$.

We can suppose that the true (optimal) decision function belongs to Φ_M it is followed from this statement 1.

Definition 3. Define a nature strategy c_M (generated by logical decision function $f \in \Phi_M$) such as set of parameters satisfying the following conditions:

1) $\sum_{t=1}^{M} p_x^t = 1$,

2) $P(E_Y^t / E_X^t) = p_{y/x}^t$ (conditional distribution is same for any $x \in E_X^t$ and $y \in E_Y^t$),

3) $P(\overline{E}_Y^t / E_X^t) = 1 - p_{y/x}^t$,

where $E_X^t \in \alpha$, $E_Y^t \in \beta$, $<\alpha,\beta> \sim f \in \Phi_M$. The complexity of this strategy is *M*, i.e. $v(c_M) = M$. Note that c_M generated by logical decision function belongs to class $L_\varepsilon(M)$. Clearly, the decision function that generated this strategy is optimal function in class Φ_M.

Statement 3. For a fixed nature strategy $c_M \in L_\varepsilon(M)$ of complexity *M* the quality criterion $F(c_M,\widetilde{f})$ (risk function) of logical decision function $\widetilde{f} \in \Phi_{M'}$ of complexity M' is presented in following form:

$$F(c_M,\widetilde{f}) = F(\widetilde{\alpha}) = \sum_{t'=1}^{M'} \widetilde{p}_x^{t'} \rho^{t'} = \sum_{t'=1}^{M'} \widetilde{p}_x^{t'} (\widetilde{p}_{y/x}^{t'} - \mu_y^{t'}),$$

where $\widetilde{p}_x^{t'} = P(x \in \widetilde{E}_X^{t'}) = \sum_{t=1}^{M} p_x^t \dfrac{\mu(\widetilde{E}_X^{t'} \cap E_X^t)}{\mu(E_X^t)}$,

$$\widetilde{p}_{y/x}^{t'} = \frac{1}{\widetilde{p}_x^{t'}} \sum_{t=1}^{M} p_x^t \frac{\mu(\widetilde{E}_X^{t'} \cap E_X^t)}{\mu(E_X^t)} \left(p_{y/x}^t \frac{\mu(\widetilde{E}_Y^{t'} \cap E_Y^t)}{\mu(E_Y^t)} + (1 - p_{y/x}^t) \frac{\mu(\widetilde{E}_Y^{t'}) - \mu(\widetilde{E}_Y^{t'} \cap E_Y^t)}{1 - \mu(E_Y^t)} \right).$$

Proof. Since the decision function $\widetilde{f}$ belongs to class $\Phi_{M'}$ than there exists partition $\widetilde{\alpha} = \{\widetilde{E}_X^1,...,\widetilde{E}_X^{t'},...,\widetilde{E}_X^{M'}\}$ of space D_X and according to it the set of subsets $\widetilde{\beta} = \{\widetilde{E}_Y^{t'},...,\widetilde{E}_Y^{t'},...,\widetilde{E}_Y^{M'}\}$ of space D_Y. The expression of the criterion $F(c,\widetilde{f}) = \sum_{t'=1}^{M'} \widetilde{p}_x^{t'} (\widetilde{p}_{y/x}^{t'} - \mu_y^{t'})$ follows from statement 1, where $\widetilde{p}_x^{t'} = P(x \in \widetilde{E}_X^{t'})$, $\widetilde{p}_{y/x}^{t'} = P(y \in \widetilde{E}_Y^{t'} / x \in \widetilde{E}_X^{t'})$. Since the strategy $c = c_M$, $c_M \in L_\varepsilon(M)$ is generated by logical decision function $f \sim <\alpha,\beta> \in \Phi_M$, there is a partition $\alpha = \{E_X^1,...,E_X^t,...,E_X^M\}$ of space D_Y and according to it the set of subsets $\beta = \{E_Y^1,...,E_Y^t,...,E_Y^M\}$ of space D_Y, the sets of parameters $p_x^t = P(E_X^t)$, $p_{y/x}^t = P(E_Y^t / E_X^t)$ as provided by definition 3. Late for simplicity we will not write the mark '$\in$' and '$\cap$' in view of the events. Express the $\widetilde{p}_x^{t'}$ and $\widetilde{p}_{y/x}^{t'}$ by way of p_x^t and $p_{y/x}^t$ take account of the event distribution is inside of subsets E_X^t, E_Y^t:

$$\widetilde{p}_x^{t'} = P(\widetilde{E}_X^{t'}) = P(\cup E_X^t \widetilde{E}_X^{t'}) = \sum_{t=1}^{M} P(E_X^t) P(\widetilde{E}_X^{t'} / E_X^t) = \sum_{t=1}^{M} p_x^t \frac{\mu(\widetilde{E}_X^{t'} E_X^t)}{\mu(E_X^t)};$$

$$\widetilde{p}_{y/x}^{t'} = P(\widetilde{E}_Y^{t'} / \widetilde{E}_X^{t'}) = \frac{P(\widetilde{E}_Y^{t'} \widetilde{E}_X^{t'})}{P(\widetilde{E}_X^{t'})} = \frac{1}{\widetilde{p}_x^{t'}} P(\widetilde{E}_Y^{t'} \widetilde{E}_X^{t'}),$$

$$P(\widetilde{E}_Y^{t'}\widetilde{E}_X^{t'}) = P(D\widetilde{E}_Y^{t'}\widetilde{E}_X^{t'}) = P(\cup E_X^t D_Y \widetilde{E}_Y^{t'}\widetilde{E}_X^{t'}) = \sum_{t=1}^{M} P(E_X^t D_Y \widetilde{E}_Y^{t'}\widetilde{E}_X^{t'}) = \sum_{t=1}^{M} (P(E_X^t E_Y^t \widetilde{E}_Y^{t'}\widetilde{E}_X^{t'}) + P(E_X^t \overline{E}_Y^t \widetilde{E}_Y^{t'}\widetilde{E}_X^{t'})),$$

$$P(E_X^t E_Y^t \widetilde{E}_Y^{t'}\widetilde{E}_X^{t'}) = P(E_X^t E_Y^t) P(\widetilde{E}_X^{t'}\widetilde{E}_Y^{t'} / E_X^t E_Y^t) = p_{xy}^t \frac{\mu((E_X^t E_Y^t) \cap (\widetilde{E}_X^{t'}\widetilde{E}_Y^{t'}))}{\mu(E_X^t E_Y^t)} =$$

$$= p_x^t \frac{\mu(E_X^t \widetilde{E}_X^{t'})}{\mu(E_X^t)} p_{y/x}^t \frac{\mu(E_Y^t \widetilde{E}_Y^{t'})}{\mu(E_Y^t)},$$

$$P(E_X^t \overline{E}_Y^t \widetilde{E}_Y^{t'}\widetilde{E}_X^{t'}) = P(E_X^t \overline{E}_Y^t) P(\widetilde{E}_X^{t'}\widetilde{E}_Y^{t'} / E_X^t \overline{E}_Y^t) =$$

$$= p_x^t (1 - p_{y/x}^t) \frac{\mu((E_X^t \overline{E}_Y^t) \cap (\widetilde{E}_X^{t'}\widetilde{E}_Y^{t'}))}{\mu(E_X^t \overline{E}_Y^t)} = p_x^t \frac{\mu(E_X^t \widetilde{E}_X^{t'})}{\mu(E_X^t)} (1 - p_{y/x}^t) \frac{\mu(\overline{E}_Y^t \widetilde{E}_Y^{t'})}{\mu(\overline{E}_Y^t)},$$

where $\frac{\mu(\overline{E}_Y^t \widetilde{E}_Y^{t'})}{\mu(\overline{E}_Y^t)} = \frac{\mu(\widetilde{E}_Y^{t'}) - \mu(E_Y^t \widetilde{E}_Y^{t'})}{1 - \mu(E_Y^t)}$ and $\overline{E}_Y^t = D_Y \setminus E_Y^t$.

Remark. If the nature strategy c_M such that some subset E_Y^t coincides with the space D_Y, than

$$\widetilde{p}_{y/x}^{t'} = \frac{1}{\widetilde{p}_x^{t'}} \sum_{t=1}^{M} p_x^t \frac{\mu(\widetilde{E}_X^{t'} \cap E_X^t)}{\mu(E_X^t)} p_{y/x}^t \frac{\mu(\widetilde{E}_Y^{t'} \cap E_Y^t)}{\mu(E_Y^t)}.$$

It is followed from that $p_{y/x}^t = P(D_Y / E_X^t) = 1$, $\mu(D_Y) = 1$.

Consequence 1. If the decision function $\widetilde{f}$ belonging to Φ_M coincides with the function f belonging to Φ_M, than $F(c, \widetilde{f}) = F(c, f)$.

Consequence 2. For the decision function $\widetilde{f}$ belonging to $\Phi_{M'}$ we have the expression $P(\overline{\widetilde{E}}_Y^{t'} / \widetilde{E}_X^{t'}) = 1 - \widetilde{p}_{y/x}^{t'}$.

Really, it is follows from the statement 3, where $\frac{\mu(\overline{\widetilde{E}}_Y^{t'} E_Y^t)}{\mu(E_Y^t)} = \frac{\mu(E_Y^t) - \mu(E_Y^t \widetilde{E}_Y^{t'})}{\mu(E_Y^t)}$,

$$\frac{\mu(\overline{\widetilde{E}}_Y^{t'} \overline{E}_Y^t)}{\mu(\overline{E}_Y^t)} = \frac{1 - \mu(E_Y^t) - \mu(\widetilde{E}_Y^{t'}) + \mu(E_Y^t \widetilde{E}_Y^{t'})}{1 - \mu(E_Y^t)}.$$

Consequence 3. If we have M=1 and the optimal function f generating c_1 such that $E_Y^1 = D_Y$, than $F(c_1, f) = 0$.

Really, for the express of criterion we have $F(c, f) = \sum_{t=1}^{M} \left(P(E_X^t E_Y^t) - P_o(E_Y^t) \right) = P_o(D_X D_Y) - P_o(D_Y) = 0$.

It means that we have the event distribution in D for the nature strategy of the complexity M=1. It is case when the entropy is maximum.

Consequence 4. If we have M=1 and the optimal function f generating c_1 such that $E_Y^1 = D_Y$, than for any decision function $\widetilde{f} \in \Phi_{M'}$ the criterion $F(c_1, \widetilde{f}) = 0$.

Really, $\widetilde{p}_{y/x}^{t'} = \frac{\mu(\widetilde{E}_Y^{t'} D_Y)}{\mu(D_Y)} P_o(D_Y / D_X) = \mu(\widetilde{E}_Y^{t'})$, $\widetilde{p}_x^{t'} = \frac{\mu(\widetilde{E}_Y^{t'} D_X)}{\mu(D_X)} P_o(D_X) = \mu(\widetilde{E}_X^{t'})$,

$$F(c_1, \widetilde{f}) = \sum_{t'=1}^{M'} \mu(\widetilde{E}_X^{t'})(\mu(\widetilde{E}_Y^{t'}) - \mu(\widetilde{E}_Y^{t'})) = 0.$$

Consequence 5. If the decision function $\widetilde{f}$ belongs to Φ_1 and $\widetilde{E}_Y^1 = D_Y$, than we have $F(c_M, \widetilde{f}) = 0$ for any complexity $M \geq 1$.

Really, we have $\widetilde{p}_x = \sum_{t=1}^{M} p_x^t \frac{\mu(D_X E_X^t)}{\mu(E_X^t)} = 1$, $\widetilde{p}_{y/x} = \sum_{t=1}^{M} p_x^t \left(p_{y/x}^t \frac{\mu(D_Y E_Y^t)}{\mu(E_Y^t)} + (1 - p_{y/x}^t) \frac{1 - \mu(D_Y E_Y^t)}{1 - \mu(E_Y^t)} \right) = 1$.

As stated above when the nature strategy is unknown the problem of statistical stability of sample decision functions is appeared. The quality $F(c,\bar{f})$ of sample decision function depends on the size N of the sample, the complexity M of the distributions, and the complexity M' of the class of functions $\Phi_{M'}$ used by the algorithm $Q(v)$ and empirical criterion $F(\bar{f})$ for constructing sample decision functions $\bar{f}$. The empirical criterion $F(\bar{f})$ (empirical risk function) is presented by expression: $F(\bar{f}) = \frac{1}{N}\sum_{i=1}^{N}(1 - L(x^i, y^i)) = \sum_{t=1}^{M'} \frac{N(\widetilde{E}_X^t)}{N}\left(\frac{N(\widetilde{E}_Y^t \widetilde{E}_X^t)}{N(\widetilde{E}_X^t)} - \mu^t \right) = \sum_{t=1}^{M} \hat{p}_x^t (\hat{p}_{y/x}^t - \hat{\mu}^t)$, where $N(*)$ is a number of sample spots belonging to the corresponding subset *, $\hat{\mu}^t = \mu(\widetilde{E}_Y^t)$, $\bar{f} \sim < \widetilde{\alpha}, \widetilde{\beta} > \in \Phi_{M'}$.

On the one hand, if the constraints on the class of decision functions are too strong, then this class may be inadequate to the true distribution, and the higher the degree of inadequacy, then poorer the quality of the decision function. On the other hand, using a complex class of functions on small samples also lowers the quality for the decision function.

At present time there are two well-known approaches solving this problem. The Vapnik -Chervonenkis approach uses the principle of uniform convergence [Vapnik V.N., Chervonenkis A.Ya, 1970]: the quality criterion $F(c,\bar{f})$ depends on VC-complexity of the decision function class Φ and the level of empirical risk $F(\bar{f})$. In the case of one discrete variable prediction was provided results [Nedelko V.M., 2004]. When the nature strategy c belongs to even probability distribution class such problem was decided by the method of statistical modeling for the case of several heterogeneous variable prediction [Lbov G.S., Stupina T.A., 2003]. It is the particular case of our problem. Really, we can provide the biased estimator of criterion (risk function) $E\varepsilon_N = E_{v_N}\left|F(c,\bar{f}) - F(\bar{f})\right|$ by the statistical modeling method for any nature strategy c belonging to the class $L(M)$. It is follows from the consequence 1-4 that we have the expression $E\varepsilon_N = E_{v_N}F(\bar{f})$ for $c \in L(1)$.

Another (Bayesian) approach to solving this problem consists in the construction of the evaluation $EF(c,\bar{f})$ that is obtained by averaging over all samples of N-size. Raudys in [Raudis Sh.Yu.,1976] used that (Bayesian) approach to solving pattern recognition problem that is admitted small samples, but is imposed a fairly strong constraint on the form of the distribution.

When the nature strategy is unknown, the quality of decision function is assigned by the expectation $E_c EF(c,\bar{f})$ of criterion $EF(c,\bar{f})$, which is obtained by averaging over all distributions. This problem was solved for pattern recognition problem in the case of one discrete variable prediction [Startseva N.G.,1995], [Berikov V.B., 2002] and for regression analysis in the case of one real variable prediction [Lbov G.S., Stupina T.A., 1999].

The problem concerned at this paper generalizes the problem of pattern recognition and the problem regression analysis. From the presented above properties of the quality criterion is followed that we can use both approaches solving statistical stability problem.

Conclusion

An approach to solving the problem of heterogeneous multivariate time series analysis with respect to the sample size was considered in this paper. The solution of this problem was assigned by means of presented criterion. The universality of the logical decision function class with respect to presented criterion makes the possible to introduce a measure of distribution complexity and order all possible distributions (nature strategies) according to this measure. The logical decision function class allows us to introduce such orderings in the space of heterogeneous multivariate variables. For the fixed complexities of probability distribution and logical decision function class, the properties of this criterion are presented by means of theorem, statements and consequences. The approaches to the solution of the statistical stability sampling decision function problem were considered.

Bibliography

[Lbov G.S., 1994] Lbov G.S. Method of multivariate heterogeneous time series analysis in the class of logical decision function. Proc. RBS, 339, Vol. 6, pp.750-753.

[Lbov G.S., Starceva N.G, 1999] Lbov G.S., Starceva N.G. Logical Decision Functions and Questions of Statistical Stability. Inst. Of Mathematics, Novosibirsk.

[Lbov G.S., Stupina T.A., 2002] Lbov G.S., Stupina T.A. Performance criterion of prediction multivariate decision function. Proc. of international conference "Artificial Intelligence", Alushta, pp.172-179.

[Lbov G.S., Starceva N.G, 1994] Lbov G.S., Starceva N.G. Complexity of Distributions in Classification Problems. Proc. RAS, Vol 338, No 5,pp 592-594.

[Berikov V.,1995] Berikov V. On the convergence of logical decision functions to optimal decision functions. Pattern Recognition and Image Analysis. Vol 5, No 1, pp.1-6.

[Vapnik V.N., Chervonenkis A.Ya, 1970] Vapnik V.N., Chervonenkis A.Ya .Theory of Pattern Recognition, Moscow: Nauka.

[Nedelko V.M.,2004] Nedelko V.M. Misclassification probability estimations for linear decision functions. Proceedings of the seventh International Conference "Computer Data Analysis and Modeling". BSU. Minsk. 2004. Vol 1. pp. 171–174.

[Lbov G.S., Stupina T.A., 2003] Lbov G.S., Stupina T.A. To statistical stability question of sampling decision function of prediction multivariate variable. Proc. of the seven international conference PRIP'2003, Minsk, Vol. 2, pp. 303-307.

[Raudis Sh.Yu.,1976] Raudis Sh.Yu. Limited Samples in Classification Problems, Statistical Problems of Control, Vilnus: Inst. Of Mathematics and Computer Science, 1976, vol. 18, pp. 1-185.

[Startseva N.G.,1995] Startseva N.G. Estimation of Convergence of the Expectation of the Classification Error Probability for Averaged Strategy, Proc. Ross. RAS, vol. 341, no. 5, pp. 606-609.

[Berikov V.B., 2002] Berikov V.B. An approach to the evaluation of the performance of a discrete classifier. Pattern Recognition Letters. Vol. 23 (1-3), 227-233

[Lbov G.S., Stupina T.A., 1999] Lbov G.S., Stupina T.A.. Some Questions of Stability of Sampling Decision Functions, Pattern Recognition and Image Analysis, Vol 9, 1999, pp.408-415.

Author's Information

Gennady Lbov – Institute of Mathematics SBRAS, 4 Koptuga St, Novosibirsk, 630090, Russia; e-mail: mailto:lbov@math.nsc.ru

Tatyana Stupina – Institute of Mathematics SBRAS, 4 Koptuga St, Novosibirsk, 630090, Russia; e-mail: mailto:stupina@math.nsc.ru

RECOGNITION ON FINITE SET OF EVENTS: BAYESIAN ANALYSIS OF GENERALIZATION ABILITY AND CLASSIFICATION TREE PRUNING

Vladimir Berikov

Abstract: The problem of recognition on finite set of events is considered. The generalization ability of classifiers for this problem is studied within the Bayesian approach. The method for non-uniform prior distribution specification on recognition tasks is suggested. It takes into account the assumed degree of intersection between classes. The results of the analysis are applied for pruning of classification trees.

Keywords: classifier generalization ability, Bayesian learning, classification tree pruning.

ACM Classification Keywords: I.5.2 Pattern recognition: classifier design and evaluation

Introduction

An important problem is the analysis of generalization ability of pattern classifiers. This problem arises from the need to find a decision function having good predictive power provided that the probability distribution is unknown, and learning sample has limited size.

A number of different approaches to the solution of the problem can be formulated: experimental approach (based on one-hold-out procedure and its modifications), probabilistic approach (making preliminary evaluation of distribution law), multivariate statistical analysis, statistical learning theory, algorithmic approach, Bayesian learning theory.

Experimental approach is extremely labor-consuming; within the framework of the probabilistic approach asymptotic quality evaluations are received as a rule. For the next approaches, the finiteness of sample is taken into account; however multivariate analysis requires rather bounded classes of distributions and types of decision functions.

Statistical and algorithmic approaches are oriented basically on worst-case analysis. So the received performance estimates are powerfully lowered. Within the Bayesian approach, the average-case estimates are received. As it was shown in [1], these estimates are more fit to volumes of samples available in practical tasks.

Regrettably, the expressions obtained within the Bayesian approach, as a rule, have unclosed form, are cumbersome and labor-consuming in calculating. Thus, a problem of finding more effectively calculated evaluations (possibly, approximate) is actual. These evaluations are to be applied as quality criteria in a learning step (for designing decision functions from the sample).

The study of generalization ability undertaken in given work has the following particularities. Firstly, the Bayesian approach is applied. Secondly, the narrower class of recognition problems – the problems of recognition on finite set of events is considered. This type of problems is most suitable for analytical studies. On the other hand, the results can be extended on broadly used classes of decision functions – logical decision functions and decision trees.

Main Definitions

Let us consider a pattern recognition problem with $K \geq 2$ classes, input features $X_1, X_{2,\ldots}, X_n$ and output feature Y with domain $D_Y=\{1,\ldots,K\}$. Denote D_i as a set of values of feature X_i, $i=1,\ldots,n$. Suppose that the examples from general sample are extracted by chance, therefore the features Y, X_i are casual. A function $f : \prod_{i=1}^{n} D_i \to D_Y$ is called the *decision function*. A special kind of the decision function is a *decision tree* T. Consider binary trees: each node $t \in T$ of the tree can be branched out into two branches. Each internal node is labeled with a feature and each branch corresponds to a subdomain of definition of that feature. To each leaf we assign the majority class of all examples of this leaf.

Decision function is built by the random sample of observations of X and Y (learning sample). Let learning sample be divided into two parts. The first part is used to design decision tree T, and the second part to prune it. Let T_{pr} be a *pruned decision tree*. During the pruning process, one or more nodes of T can be pruned. By numbering the leaves of a tree, we can reduce the problem to one feature X. The values of this feature are coded by numbers $1,\ldots,j,\ldots, M$, where M is number of leaves ("events", "cells"). Let p_j^i be the probability of joint event *"X=j, Y=i"*. Denote a priory probability of the i-th class as p^i. It is evident that $\Sigma_i p^i=1$, $\Sigma_j p^i_j=p^i$. Let N be sample size, n_j^i be a frequency of falling the observations of i-th class into the j-th cell. Denote $s=(n_1^1, n_1^2, \ldots, n_1^K, n_2^1, \ldots, n_M^K)$. $j=1\ldots M$, $i=1\ldots K$. Let $\widetilde{N}$ be a number of errors on learning sample for the given decision function.

Let us consider the family of models of multinomial distributions with a set of parameters $\mathbf{\Theta}=\{\theta\}$, where $\theta=(p_1^1, p_1^2, \ldots, p_1^K, p_2^1, \ldots, p_M^K)$, $p_j^i \geq 0, \sum_{i,j} p_j^i = 1,$ In applied problems of recognition, vector θ (defining the distribution law of a recognition task) is usually unknown. We use the Bayesian approach: suppose that random vector $\Theta=(P_1^1, \ldots, P_1^K, P_2^1, \ldots, P_M^K)$ with known priory distribution $p(\theta)$ is defined on the set of parameters. We

shall suppose that Θ is subject to the Dirichlet distribution (conjugate with the multinomial distribution): $p(\theta)=\frac{1}{Z}\prod_{l,j}(p_j^l)^{d_j^l-1}$, where $d_j^l > 0$ are some given real numbers expressing a priori knowledge about distribution of Θ, l=1,... ,K, j=1,..., M, Z is normalizing constant. For instance, under $d_j^l \equiv 1$ we shall have uniform a priori distribution ($p(\theta)\equiv const$) that can be used in case of a priori uncertainty in the specification of a class of recognition tasks.

Defining a Priori Distribution with Respect to Intersection between Classes

In the given paragraph, for the simplicity, we consider the case of two classes: K=2.

In practical problems of recognition, it is always possible to expect that variables describing the observed objects are not assigned accidentally, but possess certain information. So one may believe that the misclassification probability of optimum Bayesian classifier is not too great (for instance, not more than 0,1 – 0,15). This probability expresses a degree of "intersection" between classes. Let us show how the choice of Dirichlet parameters d_j^l allows taking such a priori information into account.

Let $d_j^l \equiv d$ for all l,j, where d>0 is a parameter. Thus we assume that there is no a priori information on the preferences between cells, however a priori distribution is not uniform ($d \neq 1$). For the fixed vector of parameters θ, the probability of error for the Bayesian classifier f_B is: $P_{f_B}(\theta)=\sum_j \min\{p_j^1, p_j^2\}$. Let us find the expected probability of error $EP_{f_B}(\Theta)$, where the averaging is done over all random vectors Θ with distribution density $p(\theta)$.

Theorem. $EP_{f_B}(\Theta)=I_{0,5}(d+1,d)$, where $I_x(p,q)$ is beta distribution function: $I_x(p,q)=\frac{B_x(p,q)}{B_1(p,q)}$, $B_x(p,q)$ is incomplete beta function: $B_x(p,q)=\int_0^x v^{p-1}(1-v)^{q-1}dv$.

Proof: Consider an auxiliary lemma (see [5]):

Lemma. Let p,q,r are real numbers and $\chi(p,q,r)=\int\limits_{\substack{x+y\le 1,\\ y<x, x\ge 0, y\ge 0}} x^{p-1}y^{q-1}(1-x-y)^{r-1}dxdy$.

Then $\chi(p,q,r)=B(p+q,r)B_{0,5}(q,p)$.

Proof of Lemma. Consider the following substitution: $x=u(1-v)$, $y=uv$. We have:

$$\chi(p,q,r)=\int_0^1 du\int_0^{0.5} u^{p-1}(1-v)^{p-1}u^{q-1}v^{q-1}(1-u)^{r-1}udv=\int_0^1 u^{p+q-1}(1-u)^{r-1}du\int_0^{0.5}(1-v)^{p-1}v^{q-1}dv=$$

$=B(p+q,r)B_{0,5}(q,p)$. Lemma is proved.

Let us calculate the expected probability of mistake:

$$EP_{f_B}(\Theta)=\frac{1}{Z}\sum_j\int_\theta \min\{p_j^1,p_j^2\}\prod_{l,r}(p_r^l)^{d-1}d\theta=\frac{M}{Z}\int\limits_{\substack{p_1^1,p_1^2:\\ p_1^1+p_1^2\le 1}}\min\{p_1^1,p_1^2\}(p_1^1)^{d-1}(p_1^2)^{d-1}\times$$

$$\times\int\limits_{\substack{p_j^l:j\neq 1\\ \sum_{l,j}p_j^l=1-p_1^1-p_1^2}}\prod_{j\neq 1}(p_j^l)^{d-1}d\theta=\frac{M}{Z}\int\limits_{\substack{p_1^1,p_1^2:\\ p_1^1+p_1^2\le 1}}\min\{p_1^1,p_1^2\}(p_1^1)^{d-1}(p_1^2)^{d-1}\frac{(\Gamma(d))^{2M-2}}{\Gamma(2Md-2d)}\times$$

$$\times(1-p_1^1-p_1^2)^{2Md-2d-1}dp_1^1dp_1^2.$$

Since in the considered case a constant $Z=\frac{\Gamma(d)^{2M}}{\Gamma(2Md)}$, and from Lemma, it follows that

$$EP_{f_B}(\Theta)=\frac{2M\Gamma(2Md)}{(\Gamma(d))^2\Gamma(2Md-2d)}\int\limits_{\substack{p_1^1,p_1^2:p_1^1<p_1^2\\ p_1^1+p_1^2\le 1}}(p_1^1)^d(p_1^2)^{d-1}(1-p_1^1-p_1^2)^{2Md-2d-1}dp_1^1dp_1^2=$$

$$=\frac{2M\Gamma(2Md)}{(\Gamma(d))^2\Gamma(2Md-2d)}B(2d+1,2Md-2d)B_{0,5}(d+1,d).$$

After transformations, we obtain: $EP_{f_B}(\Theta)=\frac{\Gamma(2d+1)}{\Gamma(d)\Gamma(d+1)}B_{0,5}(d+1,d)=I_{0,5}(d+1,d)$.

The Theorem is proved.

Figure 1 shows the dependency of $EP_{f_B}(\Theta)$ from the value *d*.

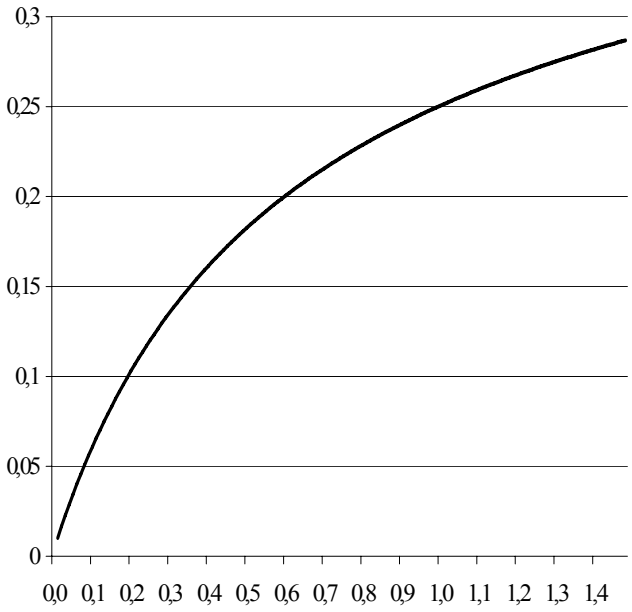


Figure 1.

Parameter *d* can be used for the definition of a priori distribution on recognition tasks: when this parameter reduces, the density of a priori distribution is changed so that classes are less intersected in average. For example, if it is assumed that the expected probability of error for optimal Bayesian classifier does not exceed 0,15, then the parameter *d* must not exceed value 0,38.

Bayesian Estimate of Decision Function Performance and Decision Tree Pruning

Hereinafter in the given work the case of uniform density $p(\theta)=const$ is considered. This assumption is defensible if a priory vagueness in choice of model is present. Let *Y=f(X)* be a decision function which has been found from sample *s* with the help of some deterministic algorithm. The probability of misclassification for this function equals to $P_f(\Theta)=1-\sum_j P_j^{f(j)}$.

The mean misclassification probability for decision function *f* is denoted as $P_{f,s}=EP_f(\Theta|s)$.

Proposition 1: $P_{f,s} = \frac{\widetilde{N} + (K-1)M}{N + KM}$.

The value $P_{f,s}$ will be called the Bayes estimate of misclassification probability for decision function *f* and sample *s*.

Proposition 2. The variance of misclassification probability equals: $VP_{f,s} = \frac{P_{f,s}(1 - P_{f,s})}{N + KM + 1}$.

The proofs are given in [2]. The mean and variance, $P_{f,s}$ and $VP_{F,s}$, can be used for calculation of tolerance interval for the value of misclassification probability [2].

Let us suppose we have an algorithm which can grow classification tree from the first part of the sample. The parameters of the algorithm should be chosen in such a way to get a large number of leaves. Next, we classify the examples from the second part of the sample to define how many examples of each class are assigned to each node. Consider arbitrary subtree *T* of the initial tree (*T* and initial tree have the same root). The set of leaves of *T* can be considered as a set of values of a discrete feature *X*. The vector of the observed frequencies for all leaves can be considered as a vector of frequencies *s*. Note that subtree *T* does not depend on the vector *s*, because the observations from pruning set are not participated in the tree building.

For subtree *T*, we can compute the Bayesian estimate of misclassification probability $P_{T,s}$. This value can be used as criterion of quality for subtree. An optimization of the criterion gives the best complexity of the tree (i.e., the number of leaves).

Now let us suppose that vector θ is fixed, but unknown parameter vector. In this case, the Bayesian estimate of misclassification probability for decision function is an approximation of the true unknown generalization error. It is possible to show that the Bayesian estimate is asymptotically unbiased. In the same time the empirical error estimate ($\widetilde{N}/N$) is unbiased, however the variance of the Bayesian estimate is less than the variance of the empirical estimate. In this sense, the Bayesian estimate is more stable.

Numeric Simulation

For numeric experiments the breast cancer database [3] was used. For decision tree growing was used algorithm C4.5 [4]. The algorithm grows a large tree from learning sample. Then this tree is pruned by second part of learning sample. The "greedy" algorithm of optimal pruning variant search is applied. After the pruning, obtained decision tree is evaluated by test data set.

Three different strategies of experiments were considered.

1. The data set is divided into three parts: for decision tree growing (50%), pruning (30%) and testing (20%). Standard reduced error pruning method (REP) [4] was used for pruning.
2. The data set divided in the same way as in first strategy. We used the Bayesian estimate of error probability for pruning.
3. The data set is divided into two parts. The first one (80%) is used for tree growing and then for pruning and the second one (20%) for testing. The Bayesian estimate is used for pruning. It is known that if growing and pruning sets coincide, the effect of overtraining arises. The purpose of this experiment is to study the behavior of the decisions in this situation.

All experiments were repeated 200 times. Before each experiment, the observations in data set were randomly mixed.

The following results of computer modeling were obtained. For first and second strategy, the errors on test sample coincide (0.022 at average). For third strategy, REP could not prune the tree; the average error on test samples for the Bayesian pruning algorithm was 0,067.

For the next experiment, artificially generated data table was used. This table was unbalanced: the frequencies of classes differ in a large degree (fist class represents 5% and second 95% of sample size of 1000 examples). The 10-fold cross-validation technique was applied for quality estimation. It turned out that the Bayesian method accuracy was 7% better than the accuracy of REP.

Conclusion

Within the framework of the Bayesian learning theory, we analyzed a classifier generalization ability for the recognition on finite set of events. It was shown that the obtained results can be applied for classification tree pruning. Numeric experiments showed that the Bayesian pruning has at least the same efficiency or better than standard reduced error pruning, and at the same time is more resistant to overtraining.

Acknowledgements

This work was supported by the Russian Foundation of Basic Research, grant 04-01-00858a

Bibliography

[1] Lbov, G.S., Startseva, N.G., *About statistical robustness of decision functions in pattern recognition problems.* Pattern Recognition and Image Analysis, 1994. Vol 4. No.3. pp.97-106.

[2] Berikov V.B., Litvinenko A.G. *The influence of prior knowledge on the expected performance of a classifier.* Pattern Recognition Letters, Vol. 24/15, 2003, pp. 2537-2548.

[3] UCI Machine Learning Database Repository. http://www.ics.uci.edu/~mlearn/ MLRepository.html

[4] Quinlan, J.R. *C4.5: Programs for Machine Learning.* Morgen Kaufmann, San Mateo, CA, 1989.

[5] Berikov V.B. *A priori estimates of recognition quality for discrete* features. Pattern Recognition and Image Analysis. V. 12, N 3, 2002. pp. 235-242.

Author's Information

Vladimir Berikov – Sobolev Institute of Mathematics SD RAS, Koptyug pr.4, Novosibirsk, Russia, 630090; e–mail: berikov@math.nsc.ru

EXTREME SITUATIONS PREDICTION BY MULTIDIMENSIONAL HETEROGENEOUS TIME SERIES USING LOGICAL DECISION FUNCTIONS[1]

Svetlana Nedel'ko

Abstract: A method for prediction of multidimensional heterogeneous time series using logical decision functions is suggested. The method implements simultaneous prediction of several goal variables. It uses deciding function construction algorithm that performs directed search of some variable space partitioning in class of logical deciding functions. To estimate a deciding function quality the realization of informativity criterion for conditional distribution in goal variables' space is offered. As an indicator of extreme states, an occurrence a transition with small probability is suggested.

Keywords: multidimensional heterogeneous time series analysis, data mining, pattern recognition, classification, statistical robustness, deciding functions.

ACM Classification Keywords: G.3 Probability and Statistics: Time series analysis; H.2.8 Database Applications: Data mining; I.5.1 Pattern Recognition: Statistical Models

[1] The work is supported by RFBR, grant 04-01-00858-a

Introduction

The specifics of multidimensional heterogeneous time series analysis consists in simultaneous prediction of several goal variables. But the most of known algorithms construct decision function for each goal variable separately. Such approach looses some information about features interdependencies [Mirenkova, 2002].

The next problem is strong increasing of dimensionality when analysing window length increases. So one has to either simplify decision functions class or make the window shorter.

The problem of insufficient sample appears much more essential [Raudys, 2001] when rare events are to be predicted.

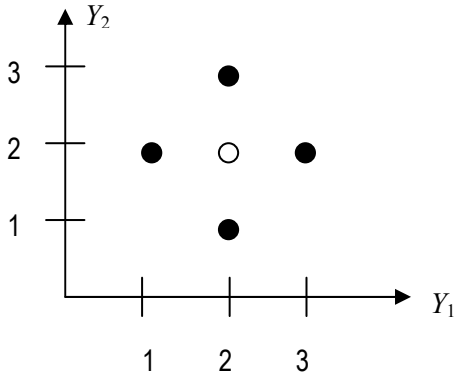


Fig. 1.

In this work an algorithm of prediction multidimensional heterogeneous time series based on finding certain partitioning that maximizes informativity criterion [Lbov, Nedel'ko, 2001] for matrix of transitions between partitioning areas. This allows to avoid increasing complexity when a window get longer, but prediction looses accuracy.

Extreme situations are characterised by low number of precedents in a period under observation. Therefore, one need statistically robust methods of multidimensional heterogeneous time series forecast.

It might be interesting also to predict events having only a few precedents or may be no precedents at all. In this case it seems to be impossible to forecast extreme situations themselves, but one could catch changing a probabilistic model of time series and consider this as an indicator of abnormal process behaviour.

Problem Definition

Let a random n-dimensional process $Z(t)=(Z_1(t), \ldots, Z_n(t))$ with discrete time be given. Features may include both continuous and discrete (with ordered or unordered values) ones. Suppose that for a time moment t values of n variables depend on its values in previous l time moments, i. e. on a window of length l.

The most algorithms for prediction multidimensional time series use replacement of time series sample by a sample in form of data table. This is made via new notation: goal values are designated as $Y_j(t)=Z_j(t)$, and previous values (prehistory) as $X_j(t)=Z_j(t\text{-}1), X_{j+n}(t)=Z_j(t\text{-}2), \ldots, X_{j+n(l\text{-}1)}(t)=Z_j(t\text{-}l)\ j = 1, \ldots, n$.

Now any time series realization $Z(t),\ t=\overline{1,T}$, may be represented like a sample $v=\left\{\left(x^i, y^i\right)\middle|\ i=\overline{1,N}\right\}$, where $N = T - l$ — the sample size. Here $y^i=\left(y_1^i,\ldots,y_j^i,\ldots,y_n^i\right),\ y_j^i = Y_j(i),\ x^i=\left(x_1^i,\ldots,x_j^i,\ldots,x_m^i\right),\ x_j^i = X_j\ (i+l)$, $m = nl$ — predictor space dimensionality. Note that the first l time moments have no prehistory of length l.

Such notation allows using a data mining methods to predict each feature separately. They may be for example classification or regression analysis methods in logical decision functions [Lbov, Startseva, 1999]. But this approach neglects features interdependence, so it is possible to construct an examples where separate decision functions give incompatible forecast [Mirenkova, 2002].

Let's consider an example that shows the weakness of separate feature forecast. Suppose two discrete features are given and probabilistic measure on them is like shown on figure 1. Each of black points has probability 0,25; another points have probability zero. Methods those make decision for every feature separately give predicted value marked by white circle. But such value combination will never occur.

This example shows necessity in methods constructing a decision rule for all features together because interdependencies are important. One need also to use decision in form of an area (in the example such area contains four black points), but not a single point.

In this work, we suggest not to separate features onto X and Y but to build partitioning in space Z directly.

Quality Criterion

Let's introduce quality criterion for decision in form of areas if goal features space. Such type criteria were proposed in [Rostovtsev, 1978].

It's suitable now to consider again separately D_X – space of predictors and D_Y – a goal features space. Let $P(E_Y)$ and $P(E_Y|x)$ be unconditional and conditional measures for $E_Y \subseteq D_Y$. Suppose a set $B_Y = \{E_Y^d \subseteq D_Y \mid d = 1,...,k\}$ of non-intersected areas to be given. Then quality criterion will be $K(B_Y) = \sum_{d=1}^{k} \left(P(E_Y^d|x) - P(E_Y^d) \right)$. Optimal decision in x will be $B_Y^* = \arg\max K(B_Y)$.

Quality criterion for conditional probabilistic measure may be defined as $K(P[D_Y|x]) = \max_{B_Y} K(B_Y)$.

This criterion is some kind of distance between conditional by given x and unconditional measures on goal features space. There are known modifications those use uniform distribution instead of unconditional one.

If B_Y is a partitioning of D_Y one needs to use modified criterion:

$$K'(B_Y) = \sum_{d=1}^{k} \left| P(E_Y^d \mid x) - P(E_Y^d) \right|. \qquad (1)$$

It differs in taking absolute values.

When the distribution is unknown and we have a sample only we can't estimate criterion for each $x \in D_X$, so need to build some partitioning λ of D_X.

Then $K(\lambda) = \sum_{E_X \in \lambda} K(P[D_Y|E_X]) \cdot P(E_X)$ will be integral decision quality criterion.

All probabilities in expression may be estimated on sample.

Algorithm

Suggested algorithm makes partitioning directly in space $D_Z = \prod_{j=1}^{n} D_j$, where D_j – a set of feature Z_j all values.

Since partitioning $\lambda = \{E^i \in D_Z \mid i = \overline{1,k}\}$ was fixed initial time series $Z(t)$ may be represented by one symbolic sequence $\beta(t) \in \{\beta^i \mid i = \overline{1,k}\}$, where β_i – a symbol correspondent to area E^i, and $\beta(t) = \beta_i$ when $Z(t) \in E^i$.

Criterion (1) may be applied to transition matrix of process $\beta(t)$:

$$K'(\lambda) = \sum_{i_0=1}^{k} \ldots \sum_{i_l=1}^{k} \left| p_{i_0 \ldots i_l} - \left(\sum_{j_0=1}^{k} p_{j_0 i_1 \ldots i_l} \right) \left(\sum_{j_1=1}^{k} \ldots \sum_{j_l=1}^{k} p_{i_0 j_1 \ldots j_l} \right) \right| , \qquad (2)$$

where $p_{i_0 \ldots i_l} = \mathrm{P}\left(\bigwedge_{\tau=0}^{l} \left(\beta(t-\tau) = \beta^{i_\tau} \right) \right) = \mathrm{P}\left(\bigwedge_{\tau=0}^{l} \left(Z(t-\tau) \in E^{i_\tau} \right) \right)$ — the probability of given prehistory of length l.

To obtain sample estimation of the criterion need to replace $p_{i_0 \ldots i_l}$ by $N_{i_0 \ldots i_l} / N$ – a rate of prehistory appearance in the sample.

Transition probabilities for partitioning areas are a kind of multi-variant decision functions [Lbov, Nedel'ko, 2001].

Note that a partitioning λ may be constructed in any appropriate class, e. g. by linear discriminating functions or by logical deciding functions (decision trees).

Logical Decision Functions

For constructing a partitioning λ we shall use algorithm LRP [Lbov, Startseva, 1999] that builds a decision tree. This algorithm was designed first for classification task and applied then for various tasks of data analysis by using special quality criteria.

The algorithm builds a partitioning onto multidimensional intervals. Here an interval is a set of neighbour values when order is defined or any subset of values if feature values are unordered. Multidimensional interval is a Cartesian product of intervals.

Algorithm LRP makes sequential partitioning the space D onto given number of areas.

Since partitioning $\left\{E^1,...,E^i,...,E^s\right\}$, $E^i \subseteq D$, was constructed on step $s-1$, on step s the algorithm goes over the all areas and selects one that being split by all possible ways onto two sub-areas provides criterion maximum. Then these sub-areas replace initial area and the process is repeated until k areas been produced.

The partitioning may be represented by decision tree. Each non-terminal node ω is correspondent to some predicate $P^{\omega} \equiv \left(z_j \in E_j^{\omega}\right)$, $E_j^{\omega} \subseteq D_j$. Each terminal node corresponds to an area of the partitioning λ.

Rare Events Prediction

Extreme situations are characterised by low number of precedents in a sample. Therefore, statistical robustness of the methods used is especially actual. Proposed method of multidimensional heterogeneous time series prediction provide high robustness.

Nevertheless, it may be not enough if there are only several precedents. Moreover, it might be interesting to predict events having no precedents.

Obviously, in this case reliable prediction is impossible, but one could try to mark time moments where extreme situation is probable. One of indicators for such time moments may be changing a probabilistic model of time series.

Since we represent initial time series by correspondent Markov chain, all related mathematical results are available. So, a moment of changing a probabilistic model can be revealed.

Another indicator of process abnormality might be occurring in correspondent symbolic chain a transition with small probability.

Conclusion

Methods of simultaneous prediction the all variables of multidimensional heterogeneous time series allows using features interdependence information in comparison with method of separate constructing a decision function for each feature. It's possible also to build decision based on partitioning initial features space that decreases algorithm complexity. As quality criterion the method uses transition matrix informativity that was introduced.

The method proposed represents initial time series by correspondent Markov chain that allows avoiding great increasing complexity when considered prehistory length increases. This is especially important for predicting rare events. Such representation also allows applying all mathematical results related to Markov chains.

To predict time moments when extreme situations have higher probability here was suggested using changes in probabilistic model of time series.

Bibliography

[Lbov, Startseva, 1999] Lbov G.S., Startseva N.G. Logical deciding functions and questions of statistical stability of decisions. Novosibirsk: Institute of mathematics, 1999. 211 p. (in Russian).

[Rostovtsev, 1978] P. S. Rostovtsev. Typology constructing algorithm for big sets of social-economy information. // Models for aggregating a social-economy information. Proceedings, publ. IE and SPP SB AS USSR, 1978. (in Russian).

[Lbov, Nedel'ko, 2001] G.S. Lbov, V.M. Nedel'ko. A Maximum informativity criterion for the forecasting several variables of different types. // Computer data analysis and modeling. Robustness and computer intensive methods. Minsk, 2001, vol 2, p. 43–48.

[Raudys, 2001] Raudys S., Statistical and neural classifiers, Springer, 2001.

[Mirenkova, 2002] S. V. Mirenkova (Nedel'ko). A method for prediction multidimensional heterogeneous time series in class of logical decision functions // Artificial Intelligence, No 2, 2002, p. 197–201. (in Russian).

Author's Information

Svetlana Valeryevna Nedel'ko – Institute of Mathematics SB RAS, Laboratory of Data Analysis, 630090, pr. Koptyuga, 4, Novosibirsk, Russia, e-mail: nedelko@math.nsc.ru

EVALUATING MISCLASSIFICATION PROBABILITY USING EMPIRICAL RISK[1]

Victor Nedel'ko

Abstract: The goal of the paper is to estimate misclassification probability for decision function by training sample. Here are presented results of investigation an empirical risk bias for nearest neighbours, linear and decision tree classifier in comparison with exact bias estimations for a discrete (multinomial) case. This allows to find out how far Vapnik–Chervonenkis risk estimations are off for considered decision function classes and to choose optimal complexity parameters for constructed decision functions. Comparison of linear classifier and decision trees capacities is also performed.

Keywords: pattern recognition, classification, statistical robustness, deciding functions, complexity, capacity, overtraining problem.

ACM Classification Keywords:I.5.1 Pattern Recognition: Statistical Models

Introduction

One of the most important problems in classification is estimating a quality of decision built. As a quality measure, a misclassification probability is usually used. The last value is also known as a risk. There are many methods for estimating a risk: validation set, leave-one-out method etc. But these methods have some disadvantages, for example, the first one decreases a volume of sample available for building a decision function, the second one takes extra computational resources and is unable to estimate risk deviation. So, the most attractive way is to evaluate a decision function quality by the training sample immediately.

But an empirical risk or a rate of misclassified objects from the training sample appears to be a biased risk estimate, because a decision function quality being evaluated by the training sample usually appears much better than its real quality. This fact is known as an overtraining problem.

To solve this problem in [Vapnik, Chervonenkis, 1974] there was introduced a concept of capacity (complexity measure) of a decision rules set. The authors obtained universal decision quality estimations, but these VC–estimations are not accurate and suggest pessimistic risk expectations.

For a case of discrete feature in [Nedel'ko, 2003] there were obtained exact estimations of empirical risk bias. This allows finding out how far VC–estimations are off.

The goal of this paper is to extrapolate the result on continuous case including linear and decision tree classifiers.

[1] The work is supported by RFBR, grant 04-01-00858-a

Formal Problem Definition

A classification task consists in constructing a deciding function that is a correspondence $f: X \to Y$, where X – a features values space and $Y = \{1, k\}$ – a forecasting values space. For simplicity let's assume a number of classes $k = 2$.

For the determination of deciding functions quality one need to assign a loss function: $L: Y^2 \to [0,\infty)$ that for classification task will be $L(y,y') = \begin{cases} 0, & y = y' \\ 1, & y \neq y' \end{cases}$, where $y \in Y,\ y' \in Y$.

By a risk we shall understand an average loss:

$$R(c,f) = \int L(y, f(x)) dP_c[D],$$

where C is a set of probabilistic measures on $D = X \times Y$ and $c \in C$ is a measure $P_c[D]$. The set C contains all the measures for those a conditional measure $P_c[Y/x]$ exists $\forall x \in X$.

Hereinafter we shall use square parentheses to indicate that the measure is defined on some σ-algebra of subsets of the set held, i. e. $P_c[D]: \mathrm{A} \to [0,1]$, where $\mathrm{A} \subseteq 2^D$ – a σ-algebra.

For building a deciding function there is a random independent sample $v_c = \left\{ \left(x^i, y^i\right) \in D \,\middle|\, i = \overline{1,N} \right\}$ from distribution $P_c[D]$ used.

An empirical risk will be sample risk estimation: $\tilde{R}(v,f) = \frac{1}{N} \sum_{i=1}^{N} L\left(y^i, f\left(x^i\right)\right)$.

For the all practically used classification algorithms an empirical risk appears biased risk estimation, being always lowered, as far as the algorithms minimize an empirical risk. So, estimating this bias is actual.

Let $$F(c,Q) = ER\left(c, f_{Q,v}\right), \quad \tilde{F}(c,Q) = E\tilde{R}\left(c, f_{Q,v}\right).$$

Here $Q: \{v\} \to \{f\}$ is an algorithm building deciding functions, and $f_{Q,v}$ – a deciding function built on the sample v by the algorithm Q.

An expectation is calculated over the all samples of volume N.

Introduce an extreme bias function:

$$S_Q\left(\tilde{F}_0\right) = \hat{F}_Q\left(\tilde{F}_0\right) - \tilde{F}_0, \qquad (1)$$

where $\hat{F}_Q\left(\tilde{F}_0\right) = \sup_{c:\tilde{F}(c,Q)=\tilde{F}_0} F(c,Q)$.

We use a supremum because a distribution c is unknown and we assume the "worst" case.

Multinomial Case

In [Nedel'ko, 2003] there is reported the dependency $S_Q\left(\tilde{F}_0\right)$ for the multinomial case when X is discrete, i. e. $X = \{1,\ldots,n\}$, and Q minimizes an empirical risk in each $x \in X$.

For the further comparison let's remember a dependency $S_Q\left(\tilde{F}_0\right)$ in asymptotic case: $\frac{N}{n} = M = \mathrm{const},\ N \to \infty$, $n \to \infty$. Though this is an asymptotic case, the results are applicable to real tasks because the asymptotic bias dependency is close to one for finite samples.

This asymptotic approximation is wholly acceptable already by $n = 10$, herewith it has only one input parameter M.

First, consider "deterministic" case when $\tilde{F}_0 = 0$. In this case $S_Q(0) = \begin{cases} e^{-M/2}, & M \leq 1 \\ \frac{1}{2Me}, & M \geq 1 \end{cases}$.

In general case of $\tilde{F}_0 > 0$ there is no simple analytical formula for $S_Q\left(\tilde{F}_0\right)$ and this dependence is given by plot.

Estimates by Vapnik and Chervonenkis

Now we can calculate an accuracy of Vapnik–Chervonenkis evaluations for the considered case of discrete *X*, as far as we know an exact dependency of average risk on the empirical risk for the "worst" probabilistic measure.

For $S(\widetilde{F}_0)$ in [Vapnik, Chervonenkis, 1974] there is reported an estimate $S'_V(\widetilde{F}_0)=\tau$, as well as an improved estimate: $S'_V(\widetilde{F}_0)=\tau^2\left(1+\sqrt{1+\frac{2\widetilde{F}_0}{\tau^2}}\right)$, where τ asymptotically tends to $\sqrt{\frac{\ln 2}{2M'}}$, $M'=M/\left(1-e^{-M}\right)$.

By substitution $\widetilde{F}_0=0$ there is resulted $S'_V(0)=\frac{\ln 2}{M'}$.

Let's perform a simple inference of the last formula.

Consider a difference between risk and empirical risk:

$$P\left(\left|R-\widetilde{R}\right|>\varepsilon\right)=P\left(\widetilde{R}=0/R=\varepsilon\right)=(1-\varepsilon)^N .$$

Since the algorithm minimizes an empirical risk, it maximizes the distance between risks:

$$P\left(\sup_{f\in\Phi}\left|R-\widetilde{R}\right|>\varepsilon\right)<|\Phi|(1-\varepsilon)^N ,$$

where Φ is a set of all decision functions. This step implies a replacement of a probability of a sum by the sum of probabilities that makes the main contribution to VC-estimates inaccuracy. Assume right term to be equal to 1 (all probabilistic levels are asymptotically equivalent) and take logarithms:

$$\ln|\Phi|+N\ln(1-\varepsilon)=\ln 1 .$$

Since $|\Phi|=2^{n\left(1-e^{-M}\right)}$ and $\ln(1-\varepsilon)\approx-\varepsilon$ obtain:

$$S'_V(0)=\varepsilon=\frac{\ln 2}{M'} .$$

Factor $1-e^{-M}$ is a non-zero numbers probability from Poisson distribution and it appears because only "non-empty" values x contribute to capacity.

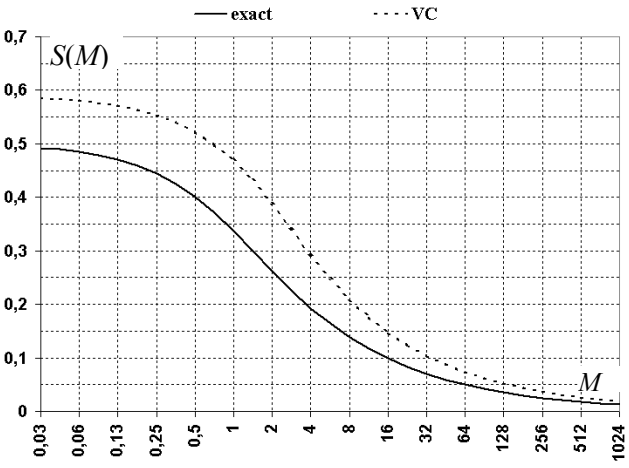


Fig. 1. Risk bias and VC–estimation. Multinomial case, $ER=0{,}5$.

A rate: $\frac{S'_V(0)}{S_Q(0)}=\frac{2Me\ln 2}{M'}\xrightarrow[M\to\infty]{}2e\ln 2\approx 3{,}77$

shows how far VC-estimates are off.

It is known that VC-estimates may be improved by using entropy as a complexity measure. Then the estimate inaccuracy will be:

$$\frac{S''_V(0)}{S_Q(0)}=2(e-1)\ln 2\approx 2{,}38 .$$

But in real tasks, entropy can't be evaluated and the last improvement has no use in practice.

On figure 1 there are drawn the dependency $S(M)=\max_{\widetilde{F}_0}S(\widetilde{F}_0)_M$ and its estimation

$$S_V(M)=\max_{\widetilde{F}_0}S_V(\widetilde{F}_0)_M = \sqrt{\frac{\ln 2}{2M'}} .$$

Plots demonstrate significant greatness of the last. Note that the accuracy of Vapnik–Chervonenkis estimation falls since $\widetilde{F}_0$ decreases.

By $M\le 1$ the "worst" distribution (that provides maximal bias) is uniform on X and the results obtained is consistent with results for multinomial case reported in [Raudys, 2001]. By $M>1$ and restricted $\widetilde{F}_0$ the "worst" distribution is not uniform on X.

Nearest Neighbors Method

This method assigns to each x a class that the most of nearest sample neighbours belongs to.

The number of neighbour objects taken into account is a parameter m that affects a statistical robustness.

Assume a measure on D to be uniform. Then misclassification probability for any decision function is 0,5 and empirical risk is: $\widetilde{F}(m) = \frac{1}{2} - \mathbf{C}_{m-1}^{\left[\frac{m-1}{2}\right]} \frac{1}{2^m}$.

Here square parentheses denote an integer part of a value.

Figure 2 shows $S(M')$ for multinomial case (solid line) and $S(m) = 0{,}5 - \widetilde{F}(m)$ for nearest neighbours classifier, where $m = M'$.

Note that though there is no capacity concept defined for nearest neighbours method the number of neighbours m plays a role of M'.

So the case $m = 1$ corresponds to unbounded capacity (when a sample can be split via decision functions by all the ways). If capacity is unbounded, we can say nothing about expected risk using empirical risk only. But it does not mean that unbounded capacity methods can not be used, it means that they must use other risk estimators.

The fact that a risk bias for multinomial case is close to bias for nearest neighbours classifier is not accidental, because analytic expression for the first one appears to be some kind of averaging the bias for the second case.

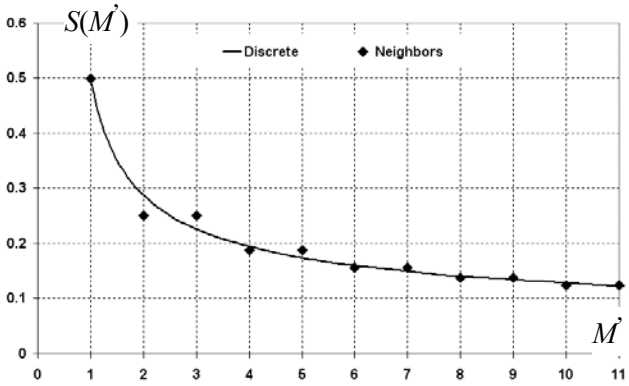


Fig. 2. Risk biases for multinomial and nearest neighbours classifiers.

Linear Decision Functions

Let us compare risk bias values for discrete case with bias for linear decision functions.

For simplifying, there was considered uniform distribution on features for both classes. For such c misclassification probability equals to 0.5 for every decision function, but empirical risk appears to be much lower.

To find a dependence $S(M)$ for linear deciding functions in $X = [0,1]^d$ a statistical modelling was used. By the modelling there was for each combination of parameters a hundred of samples drawn from uniform distribution on D, for each sample the best linear classifier built by exhaustive search. Note that the uniform distribution on D provides maximum of empirical risk bias since we put no restrictions on $\widetilde{F}_0$.

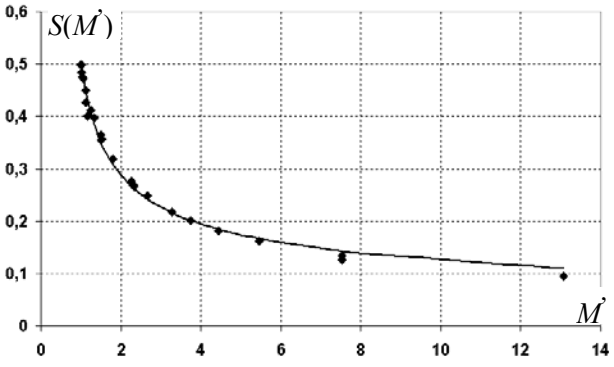


Fig. 3. Risk biases for multinomial and linear classifiers.

A table 1 shows the result of modelling. Here d – features space X dimensionality, N – sample size, $M' = \frac{N}{\log_2 \mathbf{C}}$ – sample size divided by VC-capacity of linear functions class ($\mathbf{C} = 2\sum_{m=0}^{d} \mathbf{C}_{N-1}^{m}$ is a total number of possible decision assignments to sample points by using linear decision functions), S – risk bias.

The same results are shown (by markers) on fig. 3 in comparison with $S(M')$ for discrete case (solid line).

Obtained results show that bias dependence on M' for linear functions is close to dependence for discrete (multinomial) case.

If an algorithm does not perform exhaustive search then a risk bias appears to be lower. This fact is illustrated in table 1 by value S_F that is a risk bias for the Fisher's discriminator.

Decision Tree Classifier

The goal now is to evaluate a risk bias for decision functions in form of binary decision trees [Lbov, Startseva, 1999].

Tab. 1. Risk bias for linear decision functions

d	*N*	*M'*	*S*	S_F	*d*	*N*	*M'*	*S*
1	3	1.16	0.4	0.4	1	10	2.31	0.27
1	20	3.75	0.2	0.2	1	50	7.53	0.13
1	100	13.1	0.1	0.1	2	4	1.05	0.47
2	10	1.53	0.36	0.27	2	20	2.33	0.27
2	50	4.44	0.18	0.13	2	100	7.53	0.13
3	5	1.02	0.48	0.35	3	10	1.25	0.41
3	20	1.79	0.32	0.2	3	50	3.28	0.22
3	100	5.46	0.16	0.09	4	10	1.11	0.45
4	20	1.5	0.36	0.19	4	50	2.66	0.25
5	10	1.04	0.48	0.27	5	50	2.27	0.28

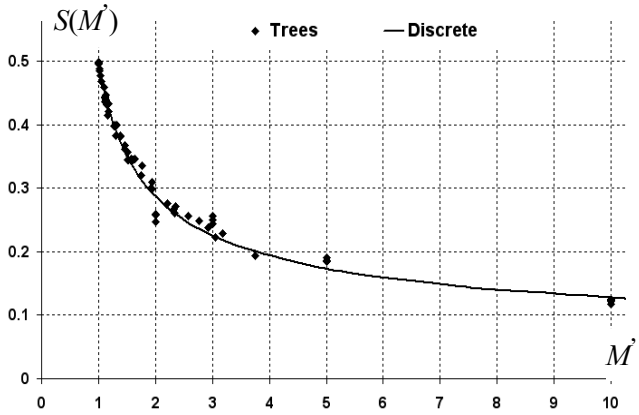


Fig. 4. Risk biases for multinomial and tree classifiers.

Decision tree is a binary tree with terminal nodes marked by goal class (certain value *y*) and non-terminal nodes marked by predicates in form: $X_j < \alpha$, where α is a value. Two arcs starting from each non-terminal node correspond to true and false predicate values.

Each decision tree forms certain sequential partitioning in X.

There was the exhaustive search algorithm implemented. The search is performed over the all decision trees with L terminal nodes and the best tree minimizing an empirical risk is founded.

While searching, the algorithm counts $\mathbf{C}$ – the number of different assignments *y* to sample objects.

Since $\mathbf{C}$ essentially differs on different samples one need to evaluate entropy $H = \mathrm{E}\log_2 \mathbf{C}$.

Then $M' = \frac{N}{H}$.

Table 2 shows statistical robustness of decision trees by different parameters while uniform distribution on D assumed. The same result is shown on figure 4 in comparison with multinomial case.

Tab. 2. Risk bias for tree decision functions

d	*N*	*L*	*M'*	*S*	*d*	*N*	*L*	*M'*	*S*
1	2	1	2	0.26	1	2	2	1	0.5
1	5	2	1.51	0.36	1	5	3	1.12	0.44
1	10	2	2.31	0.27	1	10	3	1.53	0.34
1	20	2	3.76	0.19	1	20	3	2.33	0.26
1	20	5	1.50	0.34	2	5	2	1.26	0.40
2	5	3	1.02	0.49	2	10	2	1.92	0.30
2	10	3	1.28	0.40	2	20	2	3.19	0.23
2	20	3	1.94	0.31	2	20	4	1.46	0.37
3	5	2	1.17	0.42	3	20	2	2.92	0.24
3	20	3	1.77	0.34	3	20	5	1.12	0.45
4	20	2	2.76	0.25	5	10	2	1.57	0.35

One can see again that risk bias is caused and determined by M' (sample size per complexity) rather than any other factor.

Let's compare complexities (capacities) of decision trees and linear classifier.

Table 3 shows linear classifier dimensionality d' that provides the same entropy (average number of different assignments y to sample objects) like decision trees with L terminal nodes in d-dimensional space.

Though decision trees seem to be simple, they have essential capacity. For example if $L = d$ decision trees capacity exceeds capacity of linear classifier.

But, the most of algorithms do not perform exhaustive search in whole class of decisions and their capacities are expected to be lower.

Note that if an algorithm implements good heuristic search and always founds the best decision function, then its capacity will be nevertheless equal to the capacity of exhaustive search algorithm. So, there is no use to count a number of decisions being really tested by an algorithm, because this number is irrelevant to actual capacity.

Hence, calculation of effective capacity requires different approach. Effective algorithm capacity may be estimated by the following way.

First one need to perform statistical modelling using uniform distribution on D. In this case misclassification probability (risk) equals to 0,5 for any decision function. Expectation of empirical risk is estimated by modelling, so risk bias is estimated too.

Then via comparing the bias obtained by modelling with the bias for exhaustive search algorithm, the effective capacity of the algorithm under investigation is easily revealed.

Tab. 3. Correspondent dimensionality for tree and linear decision functions. Non-integer values of d' appears because of interpolation performed.

d	*N*	*L*	*d'*	*d*	*N*	*L*	*d'*
1	5	2	1	2	5	2	1.56
2	10	2	1.4	2	20	2	1.3
3	2	2	1	3	5	2	1.83
3	10	2	1.64	3	20	2	1.47
4	5	2	2.09	4	20	2	1.59
5	10	2	1.93	10	10	2	2.45
1	5	3	2	2	5	3	2.95
2	10	3	2.86	2	20	3	2.66
3	5	3	3.76	3	10	3	3.48
3	20	3	3.07	4	5	3	3.99
4	10	3	3.94	2	5	4	3.99
2	20	4	4.26	3	5	4	4
3	10	4	5.82	3	20	4	5.1
4	10	4	6.77	1	10	5	4
2	10	5	6.45	3	15	5	7.77

Conclusion

Risk estimates by Vapnik and Chervonenkis are known to be excessively pessimistic. But the approach based on complexity measure is very attractive because of universality. The work presented shows that the reason for such pessimistic estimates is an inaccurate inference technique, but not the worst case orientation. So, it is possible to obtain estimates assuming the “worst” distribution and the ‘worst’ sample but these estimates will be appropriate in practice.

For the multinomial case (a discrete feature) there was found how far Vapnik–Chervonenkis risk estimations are off. For continuous features the dependence of risk bias on complexity in considered cases is close to multinomial one that ensures a possibility to apply obtained scaling of VC-estimates to real tasks, e.g. linear decision functions and decision trees. The results obtained for multinomial case may be propagated on continuous one by using VC-capacity of decision function class instead of n.

Comparison of linear classifier and decision trees capacities is also performed.

There was also described a method for estimation an effective capacity of an algorithm that does not perform exhaustive search in the class of decision functions.

Bibliography

[Vapnik, Chervonenkis, 1974] Vapnik V.N., Chervonenkis A. Ja. Theory of pattern recognition. Moscow “Nauka”, 1974. 415p. (in Russian).

[Raudys, 2001] Raudys S., Statistical and neural classifiers, Springer, 2001.

[Lbov, Startseva, 1999] Lbov G.S., Startseva N.G. Logical deciding functions and questions of statistical stability of decisions. Novosibirsk: Institute of mathematics, 1999. 211 p. (in Russian).

[Nedel’ko, 2003] Nedel’ko V.M. Estimating a Quality of Decision Function by Empirical Risk // LNAI 2734. Machine Learning and Data Mining, MLDM 2003, Leipzig. Proceedings. Springer-Verlag. 2003. pp. 182–187.

Author's Information

Victor Mikhailovich Nedel’ko – Institute of Mathematics SB RAS, Laboratory of Data Analysis, 660090, pr. Koptyuga, 4, Novosibirsk, Russia, e-mail: nedelko@math.nsc.ru

TABLE OF CONTENTS OF VOLUME 13, NUMBER 3