ITHEA

International Journal

INFORMATION THEORIES & APPLICATIONS

2007 Volume 14 Number 3

International Journal
INFORMATION THEORIES & APPLICATIONS
Volume 14 / 2007, 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.
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: *info@foibg.com* .
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.14, Number 3, 2007

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: **Institute of Information Theories and Applications FOI ITHEA**
Sofia, 1000, P.O.B. 775, Bulgaria. www.foibg.com, e-mail: info@foibg.com

ISSN 1310-0513 (printed) **ISSN 1313-0463 (online)** **ISSN 1313-0498 (CD/DVD)**

ON LOGICAL CORRECTION OF NEURAL NETWORK ALGORITHMS FOR PATTERN RECOGNITION

Levon Aslanyan, Luis de Mingo, Juan Castellanos, Vladimir Ryazanov, Fedor Chelnokov, Alexander Dokukin

Abstract: The paper is devoted to the description of hybrid pattern recognition method developed by research groups from Russia, Armenia and Spain. The method is based upon logical correction over the set of conventional neural networks. Output matrices of neural networks are processed according to the potentiality principle which allows increasing of recognition reliability.

Keywords: Pattern recognition, forecasting, neural networks, logical correction.

ACM Classification Keywords: I.2. Artificial Intelligence, I.5. Pattern Recognition, F.1.1 Models of Computation

Introduction

Mathematical recognition theory has long history and the variety of its reality modeling methods is quite wide. Every research group has its own traditions and usually works in specific area of mathematics. There are two basic approaches which are commonly said to be different. They are functional and algorithmic ones. For example, neural networks approximate output function but their parameters has no appropriate interpretation. Algorithmic models as for example algorithms of estimates calculating provide interpretable parameters though may have high calculation difficulty. Integration of scientific schools and small groups of "particular specialists" in the framework of joint projects provide possibilities for revealing potentials of different methods and their combinations. Developing of one such integrated approach is connected to the execution of series of INTAS projects by research groups from Russia, Spain, Armenia and some other countries.

Algebraic theory of pattern recognition based upon discrete analysis and algebra [1] is the basic approach which has been being used for 35 years in the Computing Centre of RAS under the direction of academician Yu.I. Zhuravlev. Research activities of the Institute for Informatics and Automation Problems of NAS Armenia lie in the same area of discrete recognition models. Their specific is the use of optimization structures of discrete isoperimetric tasks, discrete topology and hierarchical class searching [2]. Neural network models especially ones with polynomial output and linear activation functions [3] are the main area of interest of the Spanish group. In particular, they research temporal signal delays in recognition tasks. Good results have been achieved in forecasting of stock exchange and similar problems.

Some hybrid methods and applications for pattern recognition have been developed by these groups in the framework of INTAS projects 96-952, 00-367, 00-636 and 03-55-1969. One of them is based on assembling of neural networks and logical correction schemes. The main cause of this research was the idea of creating such pattern recognition and forecasting application which requires minimal human intervention or no intervention at all. It should be possible for the operator with no specific knowledge in mathematics to use that software. Such NNLC (Neural Networks with Logical Correction) application has been developed in the framework of INTAS projects 03-56-182 inno and 03-55-1969 YSF. Now we are proud to say that it has justified our expectations in a great extent. The method has shown high and stable results in many practical tasks.

Method description

Further we shall describe general training and recognition scheme for the l-classes task. The notation from [1] will be used. Let the training sample be $S_1, S_2, ..., S_m$ and the testing one $S'_1, S'_2, ..., S'_q$:

$$S_{m_{i-1}+1}, S_{m_{i-1}+2}, ..., S_{m_i} \in K_i, i = 1,2,...,l, m_0 = 1, m_l = m,$$

$$S'_{q_{i-1}+1}, S'_{q_{i-1}+2}, ..., S'_{q_i} \in K_i, i = 1,2,...,l, q_0 = 1, q_l = q.$$

For simplicity sake let us also suppose the task is solved without denials.

Finally, let us have N neural networks $A_j(S) = (\alpha_1^j(S), \alpha_2^j(S), ..., \alpha_l^j(S))$ trained for this task. It will give us the following matrix of recognition results:

$$A_j(S'_t) = (\alpha_1^j(S'_t), \alpha_2^j(S'_t), ..., \alpha_l^j(S'_t)),\ \alpha_i^j(S'_t) \in \{0,1\}, i = 1,2,...,l, \cdot j = 1,2,...,N, \cdot t = 1,2,...,q.$$

Algorithm of recognition by the group of neural networks will be designed according to the principle of potential correction [4]. New object will be assigned to the class of maximum estimation which is calculated according to the following formula:

$$\Gamma_i(S) = \frac{1}{q_j - q_{j-1}} \sum_{t=q_{j-1}+1}^{q_j} \Phi_i(S'_t, S), \cdot i = 1,2,...,l\ .$$

The variable $\Phi_i(S'_t, S)$ is called the potential between S'_t и S and is calculated as follows:

a) $\Phi_i(S'_t, S) = \begin{cases} 1, & \left|\{\alpha_i^j(S) \geq \alpha_i^j(S'_t),\ j = 1,2,...,N,\}\right| / N \geq \delta, \\ 0, & otherwise. \end{cases}$

b) $\Phi_i(S'_t, S) = \{the\ number\ of\ \ correct\ inequalities\ \cdot \alpha_i^j(S) \geq \alpha_i^j(S'_t), \quad j = 1,2,...,N\}.$

A-type potential we will call monotonous, b-type one will be called weekly monotonous with monotony parameter $\delta, \quad 0 < \delta \leq 1$.

Thus, training phase consists of training of N neural networks (with no denials) and consequent calculation of binary matrix $\left\|\alpha_i^j(S'_t)\right\|_{l \times N \times q}$. New object S is classified by calculating its binary matrix $\left\|\alpha_i^j(S)\right\|_{l \times N}$ and its estimates for each class according to either a-type or b-type potential. As we have already mentioned software realization of the method has been made by means of NNLC application. By the grant system of INTAS organization the NNLC application has been qualified as innovation software.

Practical testing

In the current section results of practical experiments will be shown. It represents improvement of NNLC compared to simple neural networks.

The scheme of the first series is very simple. Four practical tasks have been chosen from open UCI repository (http://www.isc.uci.edu/~mlearn/MLRepository.html) and divided into training and testing samples. After that NNLC has been trained and tested. The following tasks have been chosen:

Breast - The task of breast cancer diagnostics was taken from [5]. The training sample consisted of 344 etalons, 218 from class “benign” and 126 from class “malignant”. Nine features, which could take integer values from 1 to 10, were used.

Housing - housing estimation in Boston suburbs [6]. The problem of automatic housing estimation is solved as price interval recognition (very low, low, average, above average, high). As features 13 ecological, social and technical indicators were used: number of rooms, rate of black population in the district, average distance from main supermarkets, air quality, etc. The sample of 242 objects was used for training and 264 objects for testing.

Ionosphere - The following task from radiophysics was considered [7]. There is a system of 16 high-frequency antennas which is used for investigating the properties of ionosphere. The problem is to separate 2 types of signals – “positive” which are reflected by free electrons in ionosphere and carry useful information about ionosphere structure, and “negative” which passed through ionosphere without reflection. The electromagnetic signals are characterized by a set of 17 pulsations each having two attributes. Hence the total number of features is 34.

Credit – credit card confirmation. Credit cards were described by 15 real or k-valued (2/3 of total amount of features) features. 342 objects were used for training.

In the table below there are represented recognition qualities of NNLC and each of basic NNs.

As the table shows results of NNLC are close to best ones of its basic NNs and sometimes are even better.

Algorithm	*Breast*	*Housing*	*Ionosphere*	*Credit*
NN1	*94.6*	*68.9*	*84.6*	*80.2*
NN2	*94.6*	*68.9*	*84.6*	*80.2*
NN3	*94.6*	*68.9*	*84.6*	*80.2*
NN4	*94.1*	*55.7*	*84.6*	*75.6*
NN5	*92.1*	*72.7*	*84.8*	*83.6*
NN6	*92.7*	*67.8*	*89.6*	*76.7*
NN7	*95.2*	*70.8*	*85.7*	*83.6*
NN8	*95.5*	*68.6*	*85.7*	*84.8*
NN9	*91.8*	*57.2*	*85.7*	*79*
NN10	*92.4*	*65.5*	*77.5*	*79*
NNLC	*94.6*	*69.7*	*91.8*	*83.6*

Acknowledgements

The authors are glad to acknowledge support of the following organizations for execution of the described research: INTAS (projects 03-56-182 inno, 04-77-7076, 03-55-1969 YSF), RFBR (projects 05-07-90333, 06-01-00492, 05-01-00332). The work has been also supported by the project 2.11 of RAS Presidium's program N14, N16 and N17.

Bibliography

[1] Zhuravlev Yu.I., On algebraic approach for pattern recognition or classification // Cybernetics problems (Problemy kibernetiki), Nauka, Moscow, 1978, N33, pp. 5-68. (In Russian).

[2] Aslanyan L., Zhuravlev Yu., Logic Separation Principle, Computer Science & Information Technologies Conference // Yerevan, 2001, pp. 151-156.

[3] Luis Mingo, Levon Aslanyan, Juan Castellanos, Miguel Diaz and Vladimir Riazanov // Fourier Neural Networks: An Approach with Sinusoidal Activation Functions. International Journal Information Theories and Applications. Vol. 11. ISSN: 1310-0513. 2004. Pp. 52-53.

[4] Zuev Yu.A., Method for increasing of classification reliability based upon monotony principle in case of multiple classifiers // Journal of Calculating mathematics and Mathematical Physics, 1981, T.21, N1, pp. 157-167.

[5] Mangasarian O. L., Wolberg W.H.: "Cancer diagnosis via linear programming", SIAM News, Volume 23, Number 5, September 1990, pp 1 - 18.

[6] Harrison, D. and Rubinfeld, D.L. Hedonic prices and the demand for clean air, J. Environ. Economics & Management, vol.5, 81-102, 1978.

[7] Sigillito, V. G., Wing, S. P., Hutton, L. V., \& Baker, K. B. (1989). Classification of radar returns from the ionosphere using neural networks. Johns Hopkins APL Technical Digest, 10, 262-266.

Authors' Information

Aslanyan L.A. – Institute for Informatics and Automation Problems, NAS Armenia, P.Sevak St. 1, Yerevan-14, ARMENIA, lasl@sci.am.

Mingo L.F. – Dpto. Organización y Estructura de la Información, Escuela Universitaria de Informática, Universidad Politécnica de Madrid, Crta. de Valencia km. 7 - 28031 Madrid, SPAIN, lfmingo@eui.upm.es.

Castellanos J.B. – Dpto. Inteligencia Artificial, Facultad de Informática, Universidad Politécnica de Madrid, Boadilla del Monte – 28660 Madrid, SPAIN, jcastellanos@fi.upm.es.

Ryazanov V.V., Chelnokov F.B., Dokukin A.A. – Department of Mathematical Pattern Recognition and Methods of Combinatorial Analysis, Computing Centre of the Russian Academy of Sciences, 40 Vavilova St., Moscow GSP-1, 119991, RUSSIAN FEDERATION, rvvccas@mail.ru, fchel@mail.ru, dalex@ccas.ru.

LOGIC BASED PATTERN RECOGNITION - ONTOLOGY CONTENT (1) [1]

Levon Aslanyan, Juan Castellanos

Abstract: Pattern recognition (classification) algorithmic models and related structures were considered and discussed since 70s: – one, which is formally related to the similarity treatment and so - to the discrete isoperimetric property, and the second, - logic based and introduced in terms of Reduced Disjunctive Normal Forms of Boolean Functions. A series of properties of structures appearing in Logical Models are listed and interpreted. This brings new knowledge on formalisms and ontology when a logic based hypothesis is the model base for Pattern Recognition (classification).

ACM Classification Keywords: I.5.1 Pattern Recognition: Models

1. Introduction

Pattern Recognition is in reasonable formalization (ontology) of informal relations between objects visible/measurable properties and of object classification by an automatic or a learnable procedure. Among the means of formalization (hypotheses) - metric and logic based ones are the content of series of articles started by the current one. The stage of pattern recognition algorithmic design in 70s dealt with algorithmic models – which are huge parametric structures, combined with diverse optimization tools. Algorithmic Models cover and integrate wide groups of existing algorithms, integrating their definitions, and multiplying their resolution power. Well known example of this kind is algorithmic model of estimation of analogies (AEA) given by Yu. I. Zhuravlev [1]. This model is based indirectly on compactness hypothesis, which is theoretically related to the well known discrete isoperimetric problem (3). The optimization problem of isoperimetry is a separate theoretical issue and its pattern recognition implementations are linked alternatively to the general ideas of potential functions [4]. We present the logical separation (LSA) algorithmic model, as it is described below, to be one of the generalizations of algorithmic model of estimation of analogies. For AEA models a number of useful combinatorial formulas (algorithms) to calculate the analogy measure of objects and of objects and classes were proven [2]. These are the basic values for the decision making rules in AEA. In these models large number of parameters appears, being consecutively approximated using the appropriate optimization procedures. For this reason, a special control set besides the learning set is considered having the same formal structure as the learning set. Considering classification correctness conditions for the set of given objects by the decision procedure we get a system of restrictions/inequalities, which may not be consistent. In the simplest case a system of linear inequalities appear and then we receive a problem of approximating the maximal consistent subsystem of this basic requirements system. In terms of Boolean functions this is equivalent to the well known optimization problem of determining of one of the maximal upper zeros of a Monotone Boolean function when it is given by an operator.

LSA is based on implementation of additional logical treatments on learning set elements, and above the AEA specific metric considerations. Some formalization of additional properties on classification in this case is related to the terms of Boolean functions and especially - to the reduced disjunctive normal forms of them. Let us consider a set of logical variables (properties) $x_1, x_2, ..., x_n$ and let we have two types/classes for classification: K_1 and K_2. Let $\beta \in K_1$, and $\gamma \in K_2$, and α is an unknown object in the sense of classification. We say, that γ is separated by the information of β for α if $\beta \oplus \gamma \leq \beta \oplus \alpha$, where $\oplus$ is summation by $mod\,2$ operation. After this assumption we get, that the reduced disjunctive normal forms of two complementary partially defined Boolean functions describe the structure of information enlargement of the learning set. This construction is extending the model of estimation of analogies. It was shown that the logical separators divide the object sets into three subsets, where only one of them needs the treatment by AEA. This set is large enough for almost all weakly

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

defined Boolean functions, but for the functions with the property of compactness it is small. Let, for $0 \le k_0 < k_1 \le n$ F_{n,k_0,k_1} is the set of all Boolean functions consisting of pair of k_0 and $n-k_1$ spheres centered at 0 and 1 respectively as the sets of zeros and ones of the function. On the remainder part of vertices of n-cube the assignment/evaluation of the functions are arbitrary. This functions satisfy the compactness assumptions, and their quantity is not less than $2^{\varepsilon(n)2^n}$ for an appropriate $\varepsilon(n) \to 0$ with $n \to 0$. For these functions, also, it is enough learning set, consisting of any $n2^{n-\varepsilon(n)\sqrt{n}}$ or more arbitrary points for recovering the full classification by means of logical separators procedure. This is an example of postulations considered. The given one is relating the metric and logic structures and suppositions, although separately oriented postulations are listed as. The follow up articles will describe the mixed hierarchy of recognition metric-logic interpretable hypotheses, which helps to allocate classification algorithms to the application problems.

2. Logic Based Model

Solving the main problem of pattern recognition or classification assumes that indirect or informal information or data on classification $K_1, K_2, ..., K_l$ is given. Often this information is in form of appropriate conditions in an analogy to the compactness hypothesis, which in the very common shapes assumes, that given a metric in the space of all objects M and that closer values of classification predicate K corresponds to the pairs of "near" objects of M. We assume that objects of M are coded - characterized by the collections of values of n properties $x_1, x_2, ..., x_n$. Then each object is identified with the corresponding point of the n-dimensional characteristic space. So under the compactness of classes we assume the geometrical compactness of sets of points in the characteristic space, which corresponds to the elements of classes $K_1, K_2, ..., K_l$ and the consecutive adjustments of this property can be given in the following descriptive form: closer neighborhoods of class elements belong to the same class; the distance increase from a class element increases the class change probability; for elements pairs of different classes there exist simple paths in three parts – classes and a limited transition area in the middle.

Above we already considered the general formalization models of hypothesis by metrics and by logic. More formalizations move to more restricted sets of allowable classifications and in this regard it is extremely important to determine the level of formalisms applied. During the practical classification problem arrangements it is to check the satisfaction level of the application problem to the metric and/or logic hypothesis. Resolution is conditioned by the properties of the given learning set $\bigcup_{i=1}^{l} \mathrm{M}_i$. On the other hand there are more different conditions and methods of classification, which are very far in similarity to the model of compactness. These structures require and use other formalisms, providing the solution tools to the wide amounts of practical of pattern recognition problems. Such are the classes of algorithms of estimation of analogies, test's algorithms [2] potential function methods [4], etc.

Note that the arbitrary pattern recognition class problems can be reduced to the others, with the satisfaction of compactness type hypothesis. However this doesn't mean that the compactness hypothesis is universal because the pattern recognition problem's solution for the given space or creation of appropriate transformations to the other problems are the equivalent problems.

Now let us formulate the condition F_0, which will formalize the additional to the compactness hypothesis properties of classes. We'll consider the case of two classes ($l = 2$) intending the formalism simplifications. Particularly, in case of completing of partially defined predicate P, we will base on condition F_0. We'll apply a correspondence of considered object properties and the set of binary variables $x_1, x_2, ..., x_n$, and the same time between the partial predicate P - and it's characteristic function $f(x_1, x_2, ..., x_n)$, and will solve the modeled problem of determining (completing) of the target Boolean function F.

Let $\tilde{\alpha} = (\alpha_1, \alpha_2, ..., \alpha_n) \in \mathrm{M}$.

Consider the determination (completion) of function f in $\tilde{\alpha}$. Take the arbitrary $\tilde{\gamma}$, $\tilde{\gamma} \in \mathrm{M}_1$. If there exists such a point $\tilde{\beta}$, $\tilde{\beta} \in \mathrm{M}_0$, that $\tilde{\gamma} \oplus \tilde{\beta} \le \tilde{\gamma} \oplus \tilde{\alpha}$ (so the $\tilde{\beta}$ is different of $\tilde{\gamma}$ on a subset of the set of properties, where are different $\tilde{\alpha}$ and $\tilde{\gamma}$), then we conclude that $\tilde{\beta}$ logically separates $\tilde{\alpha}$ from $\tilde{\gamma}$, and the information, that $f(\tilde{\gamma}) = 1$ doesn't affect on the determination of the value of the function f in the point $\tilde{\alpha}$ by 1. In the opposite case we'll call $\tilde{\alpha}$ allowable in respect to the point $\tilde{\gamma}$ and to the set M_1 and decide, that information $f(\tilde{\gamma}) = 1$ influence on the determination of $\tilde{\alpha}$ by one, and the real measure of that is given by the value of the object similarity measures.

Consider the following classes of points of the n--dimensional unit cube:

N_0^f -- the set of all $\tilde{\alpha} \in \mathrm{M}$, which are allowable for the set M_0 and not allowable for M_1,

N_1^f -- the set of all $\tilde{\alpha} \in \mathrm{M}$, which are allowable for the set M_1 and not allowable for M_0,

N_2^f -- the set of all $\tilde{\alpha} \in \mathrm{M}$, which are not allowable for the sets M_0 and M_1,

N_3^f -- the set of all $\tilde{\alpha} \in \mathrm{M}$, which are allowable for both the M_0 and M_1.

[3] pointed out the general relation of condition F_0 with the notion of the reduced disjunctive normal form of Boolean functions. To see this relation let us consider the functions f and its negation $\bar{f}$, and let $\Re_f$, $\Re_{\bar{f}}$ correspondingly are the reduced forms for these functions. Denote by:

M_0^f --the collection of all points $\tilde{\alpha}$ for which $(\Re_f)_{\tilde{\alpha}} = 0$, $(\Re_{\bar{f}})_{\tilde{\alpha}} = 1$,

M_1^f --the collection of all points $\tilde{\alpha}$ for which $(\Re_f)_{\tilde{\alpha}} = 1$, $(\Re_{\bar{f}})_{\tilde{\alpha}} = 0$,

M_2^f --the collection of all points $\tilde{\alpha}$ for which $(\Re_f)_{\tilde{\alpha}} = 0$, $(\Re_{\bar{f}})_{\tilde{\alpha}} = 0$,

M_3^f --the collection of all points $\tilde{\alpha}$ for which $(\Re_f)_{\tilde{\alpha}} = 1$, $(\Re_{\bar{f}})_{\tilde{\alpha}} = 1$,

Proposition 1. $\mathrm{N}_i^f \equiv \mathrm{M}_i^f, i = 0.1.2.3.$

Proposition 2. If $\mathrm{M}_0 \cup \mathrm{M}_1 \neq 0$, then M_2^f is empty, in opposite case $\mathrm{M}_2^f \equiv \mathrm{M}$.

It is simply to prove this and some of the consecutive propositions and by this reason we omit the complete proofs and give the main idea of that. So, to prove proposition 2 consider an arbitrary point $\tilde{\alpha} \in \mathrm{M}$. If $\mathrm{M}_0 \cup \mathrm{M}_1 \neq 0$, then let us take the distance of $\tilde{\alpha}$ to the set $\mathrm{M}_0 \cup \mathrm{M}_1$ (which equals the minimal possible distance of $\tilde{\alpha}$ from any of the points of $\mathrm{M}_0 \cup \mathrm{M}_1 \neq 0$), which is in some point $\tilde{\beta} \in \mathrm{M}_0 \cup \mathrm{M}_1$. Suppose, without loss of generality, that $\tilde{\beta} \in \mathrm{M}_0$. Then the interval (binary subcube) $\mathrm{E}(\tilde{\alpha}, \tilde{\beta})$, constructed on base of points $\tilde{\alpha}$ and $\tilde{\beta}$ does not contain points from the set M_1. From here, on base of definition of reduced disjunctive normal form implies, that the point $\tilde{\alpha} \in \mathrm{M}$ is allowable in respect to the set M_0.

Proposition 3. If f_0 is an appropriate completion of function f, constructed on base of condition F_0, then $\forall \tilde{\alpha} \in \mathrm{M}_0^f (f_0(\tilde{\alpha}) = 0)$ and $\forall \tilde{\beta} \in \mathrm{M}_1^f (f_0(\tilde{\beta}) = 1)$.

Proposition 4. $\mathrm{M}_0 \subseteq \mathrm{M}_0^f$ and $\mathrm{M}_1 \subseteq \mathrm{M}_1^f$.

As a consequence from these two propositions we get, that the arbitrary completion of function f, which is made on base of condition F_0, constructs the function, allowable in respect of f. In terms of pattern recognition problems this means that arbitrary methods of recognition, which are based on the condition F_0, couldn't be "false" on the elements of the learning set $\mathrm{M}_0 \cup \mathrm{M}_1$. Write out the minimal completions of the function f, constructed on base of the condition F_0:

$$f_1(x_1,x_2,...,x_n)=\begin{cases} 0, & if\ (x_1,x_2,...,x_n)\in \mathrm{M}_0^f \\ 1 & if\ (x_1,x_2,...,x_n)\in \mathrm{M}_1^f \\ is\ not\ determined & if\ (x_1,x_2,...,x_n)\in \mathrm{M}\setminus(\mathrm{M}_0^f\cup\mathrm{M}_1^f)=\mathrm{M}_3^f \end{cases}$$

So we get some "enlargement" for the basic function f. There arose a question -- might f_1 be the new starting point (learning set, function) for the completion on base of condition F_0, and how close we can approach by this steps the final goal? The answer gives the

Proposition 5. If f_{i+1} is completion of partial function f_i, constructed on base of condition F_0, $i=1,2,...$, then $f_1 \equiv f_k, k=1,2,...$.

Let us now analyze the conditions, related to the successful continuation on base of F_0 of a partial Boolean function (that is the case of the solvable problems). Let f -- is a partially defined Boolean function and $\varphi_1,\varphi_2,...,\varphi_\tau$ -- all that functions of class $P_2(n)$ which might appear as a continuation of function f, constructed by the given assumptions. Then we are interested in conditions, when extension f_1 is allowable in respect to each of functions $\varphi_1,\varphi_2,...,\varphi_\tau$.

Consider the function f_0, defined in the following way:

$$f_0(x_1,x_2,...,x_n)=\begin{cases} 0, & if\ \varphi_i(x_1,x_2,...,x_n)=0, i=1,2,...,\tau \\ 1, & if\ \varphi_i(x_1,x_2,...,x_n)=1, i=1,2,...,\tau \\ is\ not\ defined & in\ other\ cases \end{cases}$$

Denote by $\mathrm{M}_0(f_0)$ and $\mathrm{M}_1(f_0)$ sets of all n--cube vertices, where function f_0 achieves values 0 and 1 respectively. Then our requirement can be formulated as the following: $\mathrm{M}_0^f \subseteq \mathrm{M}_0(f_0)$ and $\mathrm{M}_1^f \subseteq \mathrm{M}_1(f_0)$. Here $\mathrm{M}_3^f=\mathrm{M}\setminus(\mathrm{M}_0^f\cup\mathrm{M}_1^f)$ and $\mathrm{M}_3^f \supseteq \mathrm{M}\setminus(\mathrm{M}_0(f_0)\cup\mathrm{M}_1(f_0))$ so that this partial continuation doesn't violate the continuality of starting function to the each of functions $\varphi_1,\varphi_2,...,\varphi_\tau$. It is to mention that the conditions $\mathrm{M}_0^f \subseteq \mathrm{M}_0(f_0)$ and $\mathrm{M}_1^f \subseteq \mathrm{M}_1(f_0)$ are not convenient, which is related to the applied information on the final goal (the functions $\varphi_1,\varphi_2,...,\varphi_\tau$). Supposing the case of continuation for needs of pattern recognition problems let us show that practically useful conditions of the given type might be formulated.

Consider the structural behavior, when $n\to\infty$ and suppose a parameter $\theta\to 0$ given as. Suppose $f_0 \in P_2(n)$ (note, that the results below are true in more general cases and in more general forms). Here are some preliminary results.

1. Consider the concept $H_k^-(f_0)$ introduced by Glagolev [7]. $H_k^-(f_0)$ equals the number of vertices $\tilde{\alpha}\in E^n$, where $f_0(\tilde{\alpha})=1$, and which are covered by (involved in) maximal intervals of function f_0 of sizes not exceeding k. It was proven [7] that for almost all functions $f_0 \in P_2(n)$ $H_k^-(f_0)=o(2^n)$ when $n\to\infty$ and $k\le k_1=\log\log n-1$.

2. We'll say, [5] that the cube vertices prick the intervals including these vertexes. The set A of vertices of n-dimensional unit cube is a pricking set for the set B_k -all of the k -size intervals, if each k -subcube is pricked at least by one of the vertices of A. Denote by $K(n,k)$ the minimal number of vertices, forming a pricking set for k -subcubes. By [5] $2^{n-k} \le K(n,k) \le (n+1)2^{n-k}$. We will use the upper bound by this formulae but in our case $k \le k_1 = log\, log\, n - 1$ and a better estimation is possible as follows [4] (an extended survey on pricking is included in [6]). Let us denote by $A_i(\tilde{\alpha})$ the set of all of n-cube vertices, which lay in respect to the given vertex $\tilde{\alpha}$ on layers with numbers $\tau, \tau \equiv i(mod\, k+1)$, $i = 0,1,...,k, k \le n$. Let E^k -is an arbitrary k -subcube of an n-cube. Points of subcube E^k are placed exactly in the $k+1$ consecutive layers of E^n in respect to it's arbitrary vertex $\tilde{\alpha}$. It is correct to post the

Proposition 6. Each of the sets $A_i(\tilde{\alpha}), \tilde{\alpha} \in E^n, i = 0,1,...,k$ are pricking for the set B_k -all of the k -subcubes of n-cube, and $2^{n-k} \le K(n,k) \le 2^n / k+1$.

Proposition 7. F_0 implemented in continuation of almost all functions $f_0 \in P_2(n)$ yields the accuracy, tending to 1 as $n \to \infty$, if for the initial function f holds the condition $M_0 \cup M_1 \supseteq A_i(\tilde{\alpha})$ at least for any $i, i = 1,2,...$ and vertices $\tilde{\alpha} \in E^n$, where $A_i(\tilde{\alpha})$ is constructed for a $k \le [log\, log\, n] - 1$.

Note, that the proposiition 7 postulation is constructive, and it implies to the "sufficient" learning set, consisted no more that from $2^n / log\, log\, n$ points (which is $o(2^n)$) as $n \to \infty$. However, basically, in the pattern recognition problems it is impossible to obtain the learning set arbitrarily. Often it is formed as a random collection of any fixed size of the main collection of the studying objects.

Conclusion

Logic Separation is an element of pattern recognition hypotheses and formalisms. Structures appear in this relation based and introduced in terms of Reduced Disjunctive Normal Forms of Boolean Functions. An initial set of properties of these structures were introduced in propositions 1-7.

Bibliography

1. Zhuravlev Yu. I. On an algorithmic approach to the problems of recognition and classification. Problemi Kibernetiki, 33, (1978) 5--68.
2. Zhuravlev Yu. I. Selected Scientific Works, Publishing House Magister, Moscow, (1998) 417p.
3. Aslanyan L. H. On a pattern recognition method based on the separation by the disjunctive normal forms. Kibernetika, 5, (1975), 103--110.
4. Vapnik V. and Chervonenkis A. Theory of Pattern Recognition. “Nauka”, 1974.
5. Aslanyan L. H. The Discrete Isoperimetric Problem and Related Extremal Problems for Discrete Spaces. Problemy Kibernetiki, 36, (1979), 85--128.
6. Nechiporuk E. I. On topological principles of self-correcting. Problemy Kibernetiki, 21, (1969), 5--102.
7. Graham N., Harary F., Livingston M. and Stout Q. Subcube Fault-Tolerance in Hypercubes. Information and Computation 102 (1993), pp. 280{314.
8. Glagolev V. V. Some Estimations of D.N.F. for functions of Algebra of Logic. Problemy Kibernetiki, 19, (1967), 75--94.

Authors' Information

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

Juan Castellanos – Dpto. Intelligencia Artificial – Facultad de Informatica – Universidad Politecnica de Madrid 28660 Boadilla del Monte – Madrid, Spain; e-mail: J.Castellanos@fi.upm.es

DNA SIMULATION OF GENETIC ALGORITHMS: FITNESS COMPUTATION[1]

Maria Calvino, Nuria Gomez, Luis F. Mingo

Abstract: In this paper a computational mode is presented base on DNA molecules. This model incorporates the theoretical simulation of the principal operations in genetic algorithms. It defines the way of coding of individuals, crossing and the introduction of the individuals so created into the population. It resolves satisfactorily the problems of fitness coding. It shows also the model projection for the resolution of TSP. This is the basic step that will allow the resolution of larger examples of search problems beyond the scope of exact exponentially sized DNA algorithms like the proposed by Adleman [Adleman, 1994] and Lipton [Lipton, 1995].

Keywords: Genetic Algorithms, Fitness Function, DNA Computing, Evolutionary Computing.

ACM Classification Keywords: I.6. Simulation and Modelling, F.m. Theory of Computation

Introduction

Deoxyribonucleic acid (DNA) is the chemical inside the nucleus of all cells that carries the genetic instructions for making living organisms. A DNA molecule consists of two strands that wrap around each other to resemble a twisted ladder. The sides are made of sugar and phosphate molecules. The "rungs" are made of nitrogen-containing chemicals called bases. Each strand is composed of one sugar molecule, one phosphate molecule, and a base. Four different bases are present in DNA - adenine (A), thymine (T), cytosine (C), and guanine (G). The particular order of the bases arranged along the sugar - phosphate backbone is called the DNA sequence; the sequence specifies the exact genetic instructions required to create a particular organism with its own unique traits. Each strand of the DNA molecule is held together at its base by a weak bond. The four bases pair in a set manner: Adenine (A) pairs with thymine (T), while cytosine (C) pairs with guanine (G). These pairs of bases are known as Base Pairs (bp).

DNA and RNA computing is a new computational paradigm that harnesses biological molecules to solve computational problems. Research in this area began with an experiment by Leonard Adleman [Adleman, 1994] in 1994 using the tools of molecular biology to solve a hard computational problem. Adleman's experiment solved a simple instance of the Travelling Salesman Problem (TSP) by manipulating DNA. This marked the first solution of a mathematical problem with the tools of biology.

Computing with DNA generated a tremendous amount of excitement by offering a brand new paradigm for performing and viewing computations. The main idea is the encoding of data in DNA strands and the use of tools from molecular biology to execute computational operations. Besides the novelty of this approach, molecular computing has the potential to outperform electronic computers. For example, DNA computers may use a billion times less energy than electronic computers, while storing data in a trillion times less space. Moreover, computing with DNA is highly parallel: in principle there could be billions upon trillions of DNA or RNA molecules undergoing chemical reactions, that is, performing computations, simultaneously. Some advantages of DNA are that it is both self-complementary, allowing single strands to seek and find their own opposite sequences, and can easily be copied. Also, molecular biologists have already established a toolbox of DNA manipulations, including restriction enzyme cutting, ligation, sequencing, amplification, and fluorescent labelling, giving DNA a head start in the arena of non-silicon computing.

Despite the complexity of this technology, the idea behind DNA computing springs from a simple analogy between the following two processes, one biological and one mathematical: the complex structure of a living organism ultimately derives from applying sets of simple instructed operations (such as copying, marking, joining,

[1] This work has been partially supported by Spanish Project **TIC2003-9319-c03-03** "Neural Networks and Networks of Evolutionary Processors".

inserting, deleting, etc.) to information in a DNA sequence, and computation is the result of combining very simple basic arithmetic and logical operations.

The fact that the definition of gene implies the concept of a unit of minimum relative information as far as a functional unit and that it corresponds to the structural unit of basic molecular DNA and by association can be considered as the basic unit of mutation and of heredity, has taken it directly to trying to simulate genetic algorithms using DNA.

Overview of Genetic Algorithms

Genetic Algorithms are adaptive heuristic search algorithm premised on the evolutionary ideas of natural selection and genetic. The basic concept of GA is designed to simulate processes in natural system necessary for evolution, specifically those that follow the principles first laid down by Charles Darwin of survival of the fittest. As such they represent an intelligent exploitation of a random search within a defined search space to solve a problem.

First pioneered by John Holland in the 60s [Holland, 1975], Genetic Algorithms has been widely studied, experimented and applied in many fields in engineering worlds. Not only does GAs provide an alternative method to solving problem, it consistently outperforms other traditional methods in most of the problems link. Many of the real world problems involved finding optimal parameters, which might prove difficult for traditional methods but ideal for GAs.

GAs are based on an analogy with the genetic structure and behaviour of chromosomes within a population of individuals using the following foundations:

- Individuals in a population compete for resources and mates.
- Those individuals most successful in each 'competition' will produce more offspring than those individuals that perform poorly.
- Genes from ‘good' individuals propagate throughout the population so that two good parents will sometimes produce offspring that are better than either parent.

Thus each successive generation will become more suited to their environment. After an initial population is randomly generated, the algorithm evolves the through three operators: selection which equates to survival of the fittest; crossover which represents mating between individuals; mutation which introduces random modifications.

Selection Operator: gives preference to better individuals, allowing them to pass on their genes to the next generation. The goodness of each individual depends on its fitness. Fitness may be determined by an objective function or by a subjective judgement.

Crossover Operator: prime distinguished factor of GA from other optimization techniques. Two individuals are chosen from the population using the selection operator. A crossover site along the bit strings is randomly chosen. The values of the two strings are exchanged up to this point. If S1=000000 and s2=111111 and the crossover point is 2 then S1'=110000 and s2'=001111. The two new offspring created from this mating are put into the next generation of the population. By recombining portions of good individuals, this process is likely to create even better individuals.

Mutation Operator: with some low probability, a portion of the new individuals will have some of their bits flipped. Its purpose is to maintain diversity within the population and inhibit premature convergence. Mutation alone induces a random walk through the search space. Mutation and selection (without crossover) create parallel, noise-tolerant, hill-climbing algorithms.

One of the basic arguments of the theory of evolution is that individuals that show similarities are related. Based on this principle, Holland observed that certain groups of individuals with particular similarities in some positions in their chains had good common properties whilst others were worse. Abstracting this idea Holland defines the concept of scheme (H) in one binarian coding with chains of length ℓ, thus;

$$H = h_{\ell-1} \dots h_0 \in \{0, 1, *\}^{\ell} \leftrightarrow H = \{ s_{\ell-1} \dots s_0 / h_j \neq * \rightarrow s_j = h_j \}$$

In other words, one scheme represents a certain subgroup of the population in which the individuals differentiate themselves at most in the position of the asterisks.

For example, the scheme H = 10 ∗ 01∗ correspond with the chain group
{100010, 100011, 101010, 101011} At the same time any group of chains defines a scheme, suffise to consider the J-th Coordinate.

$$\pi_j : \{0, 1\}^{\ell} \rightarrow \{0, 1\}$$
$$s_{\ell-1} \ldots s_0 \mid\rightarrow s_j$$

And to define the scheme H thus;

$$h_j = \begin{cases} 0 \text{ if } \pi_j(H) = \{0\} \\ 1 \text{ if } \pi_j(H) = \{1\} \\ * \text{ if } \pi_j(H) = \{0, 1\} \end{cases}$$

In fact, the group of chains that can be generated by crossing the elements of the group C is exactly H. For example if C = {001011, 011111} then each chain of H = 0 * 1 * 11 can be generated by crossing the elements of C, even 011011 y 001111, that were not initially in C.

Obviously in some schemes their elements show more likeness between themselves than in others. To quantify this idea there are two concepts; The order of a scheme which is the number of fixed alleles in the scheme and the length of definition which is the distance between the first and the last of the fixed alleles. For example if H = 00 * *1*, then o(H) = 3 y δ(H) = 4.

In essence, Holland's scheme theorem affirms that the algorithm drives the search for the optimum through certain subgroups of the population. In other words, explores the space of search through those areas that on average are more adequate.

Given that during the reproduction a chromosome is selected with a probability proportional to fitness

$$\frac{f(s)}{\sum_{s \in P(t)} f(s),}$$

where P(t) denotes the population t-th. Then if we start from a population of N elements the spected number of representatives of H in the following iteration is

$$E(n(H, t+1)) = n(H, t).\ N.\ \frac{f(H, t)}{\sum_{s \in P(t)} f(s),}$$

where **E** denotes the operator hope, n(H, t) is the number of chains in the scheme
In the generation t-th and

$$f(H, t) = \frac{\sum_{s \in H} f(s)}{n(H, t)}$$

is the average value of the scheme in that generation. If we now consider the action of the operators of crossing and mutation the previous equation is transformed in:

$$E(n(H, t+1)) = n(H, t).\ \frac{f(H, t)}{\overline{f}}\ [1 - \alpha(H)]$$

where:

$$\overline{f} = \frac{\sum_{s \epsilon H} f(s)}{N}$$

Represents the average fitness of the population and **α(H)** depends of the structure of H and the probabilities of crossing and mutation, **pc** y **pm,** but not of **t**, Because:

$$\alpha(H) = p_c \cdot \frac{\delta(H)}{\ell - 1} \cdot (1 - p_m)^{o(H)}$$

This expression, ignoring the terms of grade ≥ 2 in Newton's binomial, and because pm _ 0.01, turns into the final formula of the fundamental Theorem of genetic algorithms (or Theorem of schemes):

$$E(n(H, t+1)) = n(H, t) \cdot \frac{f(H, t)}{\overline{f}} [1 - p_c \cdot \frac{\delta(H)}{\ell - 1} - o(H) \cdot p_m]$$

Thus, we have that if H is a scheme with a fitness level greater tan the average of the population it is hoped to increase the number of chains with the structure of H in the following generation as long as a(alpha) is small. In other words, the principle of this theorem's can be interpreted saying that "short schemes of lower order with greater fitness than the average increase the number of representatives in the successive generations" This type of schemes that seem to play an important role in the way that GAs act, are known as building blocks.

It seems then tha by juxtaposing solid blocks of small size, increasingly better individuals could be built. This leads us to think that functions that can be defined utilising short schemes of lower order and high suitabilty would be easy to optimise for the Gas [Mitchell, 1994]

This affirmation, known as the hypothesis of the building blocks [Holland, 2000] seems very reasonable. In fact, GAs have been designed for various applications are empirical evidence that for different types of problems such hypothesis is correct.

DNA Simulation of Genetic Algorithms

The construction of a genetic algorithm for the resolution of an optimisation problem requires the definition of the genetic architecture. In this sense the election the manner of coding of the individuals represents an important point to obtain the correct solution.

The said coding must be done in a way that each chain allows the storage of information corresponding to an individual of a genetic algorithm. [González, 2002] Later on the bits will be represented independently of the position that they are in. [Lipton, 1995]

Given that the genetic composition of an individual constitutes a multyfactorial variable, the definition of the individual within a population will be dependent of the problem to be dealt with, that is, an individual has a genome that must comply with a number of prerequisites subordinate to the problem, to be considered suitable within that population.

The process of representation of the genes as a minimum unit of information requires the analysing of the problem in question with the purpose of stipulating the number of bits assigned to the mentioned gene.

A gene will be represented as a whole of three fields; the percentage of Timine of the gene will be proportional to the fitness that represents the central field:

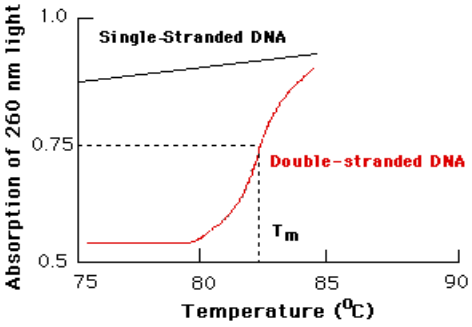
ENC(X) FITNESS xy ENC(Y)

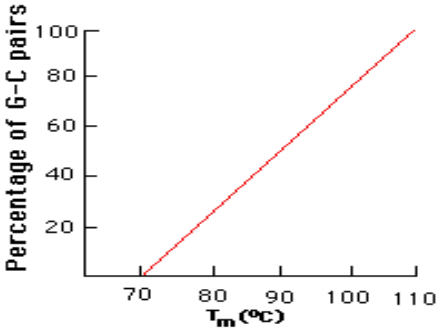
With the purpose of mapping such fields for later processing the recombining DNA will bring a linker between fields in a way that can be recognised by restriction endonucleases. [Bonen, 1996]

Given that PCR will be used for the amplification of the DNA, each individual will carry at the same time, at both ends of the chain a sequence that will hybridise with a specific primer in the annealing phase (hybridization).

The fitness must be embedded in the coding of the individuals and given its definition will be determined by the

content in G+C which implies that the fitness of an individual will be directly related with the fusion temperature and hence would be identifiable by spectophotometry (A_{260}) and separable by electrophoresis techniques (DGDE) [Macek 1997].





It is possible then to detect the perfect candidate by means of DGDE as it would be the one the possible candidates to present the greater number of GC pairs and therefore has the greater electrophotometric mobility.

The identification of the individuals of the population requires the tailing of the recombinant with a specific field that identifies the individual in a unique manner, the coding of this field will be done by means of the following function:

$$\text{CODE: } N \rightarrow \{G, C\}^*$$

Where N is the totallity of the natural numbers. It will receive a number corresponding to an individual within a population and returns a sequence of nucleotides.

The identification of the individuals in the mating zone requires again the inclusion of a field (Nm) which will be determined again by the function CODE.

The generation of the initial population has as an objective obtaining an aleatory population with a number of individulas equal to the size of the population. The complexity of the sinthesis of the sequence will be directly related to the number of genes used for the representation of the individuals within the genetic algorithm.

Basically it consists of a recombinant by means of a union of compatible fragments digested with restriction endonucleases.

The final format would look as follows:

PCR-primer Np REp XY RE0 XY RE1 … REn-1 XY REp Np -1 PCR-primer

The selection of individuals will be done by means of specific probes of the problem in question and the isolation of the individuals will be achieved by means of electrophoretic techniques (DGDE).

After selection, the individual will be introduced in the mating zone. For this he must be modified adding a specific field of such zone. In the event that a crossing of individuals is required, it is done in a temporal test tube containing the pair of individuals.

The mutation will be induced on each of the individuals results of the crossings operation in the genes in which the mutation frequency surpasses others obtained at ramdom and consists in the substitution of a gene by its complement in bases.

Later ill adapted individuals from the mating zone will be eliminated and will be substituted by the created recombinants. To determine the finalisation of the algorithm in each iteration the average of population adaptation is calculated. Once the convergence of the population is reached the best individual will be analysed by means of radioactive marking (o etching) techniques.

Fitness Computation on TSP Problem

The TSP is interesting not only from a theoretical point of view, many practical applications can be modelled as a travelling salesman problem or as variants of it, for example, pen movement of a plotter, drilling of printed circuit boards (PCB), real-world routing of school buses, airlines, delivery trucks and postal carriers. Researchers have tracked TSPs to study biomolecular pathways, to route a computer networks' parallel processing, to advance cryptography, to determine the order of thousands of exposures needed in X-ray crystallography and to determine routes searching for forest fires (which is a multiple-salesman problem partitioned into single TSPs). Therefore, there is a tremendous need for algorithms.

In the last two decades an enormous progress has been made with respect to solving travelling salesman problems to optimality which, of course, is the ultimate goal of every researcher. One of landmarks in the search for optimal solutions is a 3038-city problem. This progress is only party due to the increasing hardware power of computers. Above all, it was made possible by the development of mathematical theory and of efficient algorithms.

There are strong relations between the constraints of the problem, the representation adopted and the genetic operators that can be used with it. The goal of travelling Salesman Problem is to devise a travel plan (a tour) which minimises the total distance travelled. TSP is NP-hard (NP stands for non-deterministic polynomial time) - it is generally believed cannot be solved (exactly) in time polynomial.

TSP Solution using a DNA Genetic Algorithms Simulation. Fitness Computation.

Applying the previous protocol to the TSP of four cities in which the total size of the population is 256 (N) and the number of arches 6 (M) the individuals will be coded with an amount T inversily proportional to the length of the arches. The resulting sequence is shown in fig 1:

arch	Distance	Number of nucleotides	Fitness	Nucleotides of G
AB	1	40	1	40
BC	1	40	1	40
CD	4	40	4	10
BD	3	40	3	13
AD	2	40	2	20
AC	6	40	6	6

Figure 1.- Results of TSP simulation

The final format would look like this:

PCR-primer Np REp XY RE0 XY RE1 … REn-1 XY REp Np-1 PCR-primer

where XY ϵ { AB, BC, CD, AC, BD }

The selection criteria of the individuals in this case involves the encoding for an existing way to do this the strands of selection join two vertex of the graph neither initial nor final.

To be discarded are those individuals that do not start in the original city, those that do not finish in the final city as well as those that are found in repeated cities.

Selected are those structures which in the stretch O to N are covered by the strands of selection.

The format for the introduction of individuals in the mating zone is the following:

PCR-primer Np Nm REp XY RE0 XY RE1 … REn-1 XY REp Np-1 PCR-primer

where XY ϵ { AB, BC, CD, AC, BD }

We proceed then to the crossing mutation and evaluation of the degree of adaptation of the individuals prior to their introduction to the population and the process is repeated until the convergence of the population obtaining in each iteration the degree of adaptation

Conclusion

The generation of this work has produced a new approach to the simulation of genetic algorithms with DNA. The problem of fitness evaluation in a parallel form has been resolved satisfactorally. This does not imply that the definition would be independent of the problem at hand even though there are rules that facilitate such definitions and that can be solved by means of genetic algorithms.

In a GA simulated with DNA the concept fitness field disappears.

The coding of the individuals is closely related with the characteristics of electrophoretic migration.

The fitness of the individual is embedded in his coding and any attempt to add such field represents a grave error. The addition of such field would mean a personalisation of the individual thus preventing a massive and anonymous parallelism.

Bibliography

[Adleman, 1994] Leonard M. Adleman. Molecular Computation of Solutions to Combinatorial Problems. Science (journal) 266 (11): 1021–1024. 1994.

[Adleman, 1998] Leonard M. Adleman. Computing with DNA. Scientific American 279: 54-61.1998.

[Amos, 2005] Martyn Amos.Theoretical and Experimental DNA Computation, Springer. ISBN 3-540-65773-8. 2005.

[Baum, 1996] Eric B. Baum, Dan Bohec. Running dynamic programming algorithms on a DNA computer. Proceedings of the second Annual Meeting on DNA Based Computers. 1996.

[Bonen, 1996] Dan Boneh, Christopher Dunworth, Richard J. Lipton, and Jiri Sgall. On the Computational Power of DNA. DAMATH: Discrete Applied Mathematics and Combinatorial Operations Research and Computer Science 71. 1996.

[Darwin C] Charles Darwin. On the Origin of the Species by the Means of Natural Selection, or the Preservation of Favoured Races in the Struggle for Life. Murray, London, 1859.

[González, 2002] Eduardo González Jiménez, Valery I. Poltev. La simulación computacional de procesos genéticos a nivel molecular. Ciencia y Cultura. Elementos, 479(47): September- November. 2002.

[Holland, 1975] J.H.Holland. Adaptation in Natural and Artificial Systems. MIT Press. 1975.

[Holland, 2000] Holland, J. H. (2000). Building blocks, cohort genetic algorithms, and hyperplane-defined functions. Evolutionary Computation, 8:4, 373-391.

[Kari, 2000] Lila Kari, Greg Gloor, Sheng Yu. Using DNA to solve the Bounded Post Correspondence Problem. Theoretical Computer Science 231 (2): 192–203. 2000.

[Lipton, 1995] Richard J.Lipton. Using DNA to solve NP-Complete Problems. Science, 268:542-545. April 1995. [Paun, 1998] Gheorge Paun, Grzegorz Rozenberg, Arto Salomaa. DNA Computing - New Computing Paradigms, Springer-Verlag. ISBN 3540641963. 1998.

[Mitchell, 1994] Mitchell, M., Holland, J. H., & Forrest, S. (1994). When will a genetic algorithm outperform hill climbing? IAdvances in Neural Information Processing Systems 6, 51-58, MIT Press.

[Macek 1997] Milan Macek M.D. Denaturing gradient gel electrophoresis (DGDE) protocol. Hum Mutation 9: 136 1997.

Authors' Information

Maria Calvino – Dep. Inteligencia Artificial. Facultad de Informática, Universidad Politécnica de Madrid, Boadilla del Monte, 28660 Madrid, Spain; e-mail: maria.calvino@medicimage.com

Nuria Gomez – Dep. Organización y Estructura de la Información, Escuela Universitaria de Informática, Universidad Politécnica de Madrid, 28031 Madrid, Spain; e-mail: ngomez@dalum.eui.upm.es

Luis Fernando de Mingo Lopez – Dep. Organización y Estructura de la Información, Escuela Universitaria de Informática, Universidad Politécnica de Madrid, 28031 Madrid, Spain; e-mail: lfmingo@eui.upm.es

APPLICATIONS OF RADIAL BASIS NEURAL NETWORKS FOR AREA FOREST

Angel Castellanos, Ana Martinez Blanco, Valentin Palencia

Abstract: This paper proposes a new method using radial basis neural networks in order to find the classification and the recognition of trees species for forest inventories. This method computes the wood volume using a set of data easily obtained. The results that are obtained improve the used classic and statistical models.

Keywords: Neural Networks, clustering, Radial Basis Functions, Forest Inventory.

ACM Classification Keywords: I.5. Pattern Recognition – I.5.1. Neural Nets; I.5.3. Clustering

Introduction

The research community has developed several different neural network models, such as backpropagation, radial basis function, growing cell structures [Fritzke 1994] and self-organizing feature maps [Kohonen 1989]. A common characteristic of the aforementioned models is that they distinguish between learning and a performance phase. Neural networks with radial basis functions have proven to be an excellent tool in approximation with few patterns. Most relevant research in theory, design and applications of radial basis function neural networks is due to Moody and Darken [Moody and Darken, 1989], Renals [Renals, 1989] and to Poggio and Girossi [Poggio and Girosi, 1990].

Radial basis function (RBF) neural networks (Figure 1) provide a powerful alternative to multilayer perceptron (MLP) neural networks to approximate or to classify a pattern set. RBFs differ from MLPs in that the overall input-output map is constructed from local contributions of Gaussian axons, require fewer training samples and train faster than MLP.

In this paper, we propose a method using radial basis neural networks in order to find the classification and the recognition of trees species for forest inventories. We use an unsupervised technique called the k-nearest neighbors rule to estimate centers and widths of the functions of radial base. The centers of the clusters give the centers of the RBFs and the distance between the clusters provides the width of the Gaussian function. Computation of the centers, used in the kernels function of the RBF neural network, is being the main focus to study in order to achieve more efficient algorithms in the learning process of the pattern set.

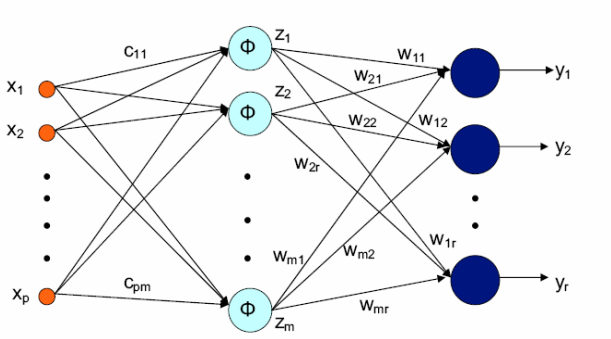


Figure 1.- Radial Basis Function Neural Network.

Problem Description

This paper seeks to estimate the wood volume for area forest inventory and find a classification of trees species, using a set of data that can be easily obtained such as: diameter, thickness bark, grow of diameter and height.

Volume parameter is one of the most important parameters in forest research when dealing with some forest inventories [Schreuder, H.T., Gregoire, T.G. and Word, G.B. 1993]. Usually, some trees are periodically cut in

order to obtain such parameters using cubical proofs for each tree and for a given environment. This way, a repository is constructed to be able to compute the volume of wood for a given area and for a given tree specie in different environments. Stem volume formula is function of a tree's height, basal area, shape, etc. Volume is one of the most difficult parameters to measure, because an error in the measure or assumptions for any one of the above factors will propagate to the volume estimate. Volume is often measured for specific purposes, and interpretation of the volume estimate will depend on the units of measurement standards of use, and others specifications. Calculations of merchantable volume may also be based on true cubic volume. Direct and indirect methods for estimating volume are available [Hamilton, F. and Brack, C.L. 1999].

The method more usual to estimate volume in forest is the tree volume tables or tree volume equations. Huber's volume equations are a very common equation used to estimating volume:

$$V = h\pi\left(\frac{d}{2}\right)^2$$ V denotes volume, h denotes length, d denotes diameter.

Another form of previous equation is:

$$V = \eta h\pi\left(\frac{d}{2}\right)^2$$ η = factor for the merchantable volume

We present a study of the potential wood forest amount, that is, the maximum amount of wood that can be obtained. All data are taken from an inventory of the M-1019 area at "Elenco" in Madrid (Spain), at "Atazar" village. Most of the trees belongs to the Pinus Pinaster family and a small amount to the Pinus Sylvestris family. All this area is focused on the wood production. The area is divided into two different sub areas with a surface of 55.6 Ha and 46.7 Ha respectively.

The main aim is to be able to forecast the wood volume and detect relationships between all the variables that are in our study. Variables taken into account are: normalized diameter, total height, thickness bark, and radial growth in the last ten years. Normalized diameter has been measured in the whole feet of the two sub areas that made up the samples, provided they are larger than 12.5 cm till the last cluster of 60 cm.

In this work we compare solutions obtained with statistical regression analysis and classical methods as Huber's, with the results using radial basis function neural network.

Classifiers for the prediction in forest products

A radial basis function neural network has been implemented with four input neurons: diameter, thickness bark, grow of diameter and height, in order to estimate the volume of wood that can be used.

The net uses a competitive rule with full conscience in the hidden layer and one output layer with the than function, all the learning process has been performed with the momentum algorithm. Unsupervised learning stage is based on 100 epochs and the supervised learning control uses as maximum epoch 1000, threshold 0.01. We have performed an initial study using 260 patterns in training set; after a 90 patterns in training set and finally with only 50 patterns in training set, and the error MSE, are similar in three cases.

Problem under study is prediction of volume of wood, and it is compared to other methods such as the Huber's formula and the statistical regression analysis in order to estimate the amount of wood using typical tree variables: diameter, thickness and diameter growth. Neural networks had approximated in a good manner tested examples, getting a small mean squared error, see table below. Radial basis function neural network learns with only a few patterns, that is the way results using only 50 patterns are really excellent. For each of the tree species tested, the RBF gives less MSE estimated than the standard formulas Huber and Multivariate Analysis Regression.

	Error-Huber	Error-RBF	Error-Regression Multivariate
MSE	0.05	0.007	0.01

Next step consists on forecasting the input variable importance (sensitive analysis) in the learning process. Our neural network is a mapping $f(x_1, x_2, x_3, x_4): \Re^4 \to \Re$ where $x_1 = diameter(cm)$, $x_2 = thickness\,bark(cm)$, $x_3 = growth\,of\ diameter(cm)$, $x_4 = height(cm)$, in order to forecast variable $x_5 = volume(dm^3)$. All centers are stable in two points that are those who signal the two main clusters, and that the net has been able to detect the two tree species.

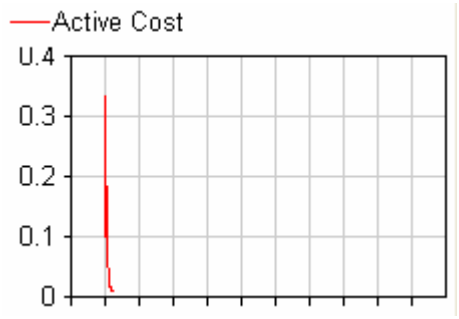
Several matrixes have been computed; where columns are input variables to forecast and rows are hidden neurons. These matrixes show the center values. Variable $x_3 = diameter\,growth$ takes the same value in both centers what it means that the study can be done without such variable obtaining similar values of MSE. Main centers of RBF approximate real clusters in the two forest areas, following table shows the real clustering.

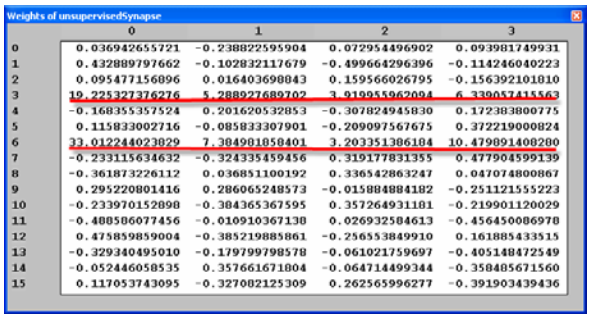
Zone - species	x_1	x_2	x_3	x_4
1	19,49	5,28	3,19	6,45
2	33,71	7,38	3,91	10,66

Previous table shows the matrix where the columns represent the input variable and the rows represent the hidden neurons. The hyperspace is divided into different regions or clusters starting from 16. Later, the number of clusters has been decreased till the minimum number of possible clusters is reached in order to solve the problem minimizing the mean squared error. The number of hidden neurons must be greater than the number of input variables to perform a correct learning.

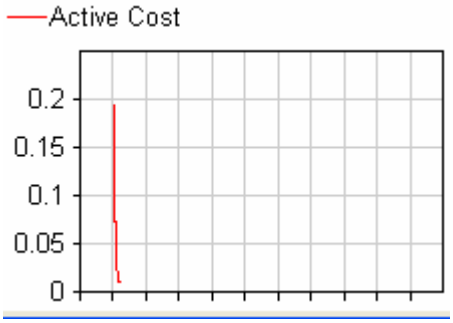
Two main centers are found in the hyperspace, see following figures.

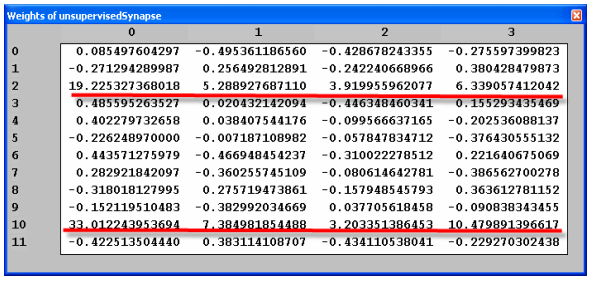
Four input variables and 16 clusters MSE=0.0079



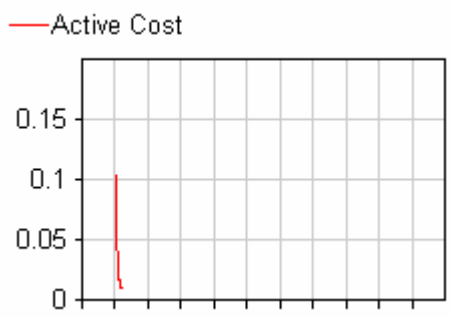


Four input variables and 12 clusters MSE=0.0078





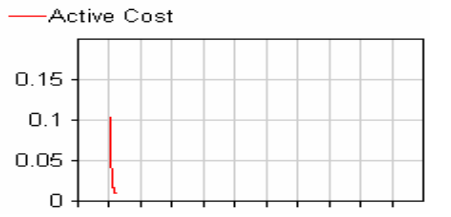
Four input variables and 8 clusters MSE=0.0075



Weights of unsupervisedSynapse

	0	1	2	3
0	0.139393292032	-0.039048432875	-0.491027558214	0.009964293344
1	0.022415845210	0.086809900204	0.144367809076	0.051072725608
2	0.098223822748	0.138782921842	-0.480559709464	0.225455488754
3	-0.001388592181	0.200430310984	-0.053514206366	0.407345805231
4	19.635513153942	5.354130101370	3.961877017807	6.443362612384
5	0.243858149968	-0.416776024659	-0.162892544328	0.434659871212
6	-0.461210974456	0.113635670034	0.494354075747	0.235343485824
7	33.531236499098	7.460325997065	3.093478779314	10.660911450110

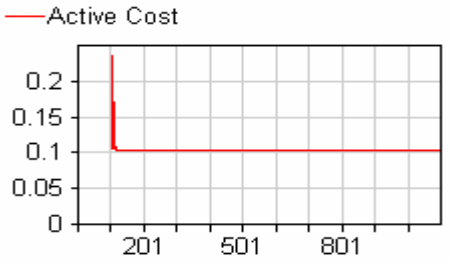
Four input variables and 5 clusters MSE= 0.0073



Weights of unsupervisedSynapse

	0	1	2	3
0	0.225424970244	0.038163396100	-0.008224738304	-0.242484817042
1	-0.326044495987	-0.496215704825	0.115955076754	-0.043595690786
2	-0.041459395123	0.219168675802	-0.375911740471	0.385494552446
3	19.635514146112	5.354130753430	3.961878389731	6.443361133609
4	33.531214807424	7.460317864873	3.093467055318	10.660923756069

Four input variables and 4 clusters MSE=0.1



Weights of unsupervisedSynapse

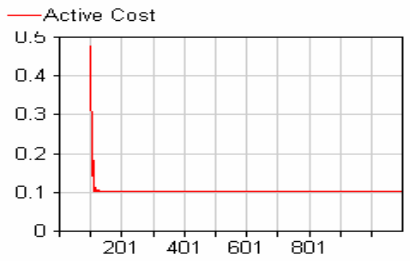
	0	1	2	3
0	25.641225758815	6.267486735499	3.586643466294	8.269150221939
1	-0.469908749657	0.400753807184	-0.449949644459	0.216422009949
2	0.180684835353	-0.020706808679	0.264641254921	-0.400875881222
3	-0.177663502915	-0.422421948912	0.222281563768	-0.498290963469

Three input variables and 4 clusters MSE=0.0078

Weights of unsupervisedSynapse

	0	1	2
0	-0.497070223090	-0.439146092105	0.293725394452
1	33.371678352723	7.454034626552	10.546546589181
2	19.492469146051	5.318156886818	6.451743076843
3	-0.314081240272	0.053331095309	0.129078035829

Three input variables and 3 clusters MSE=0.104



Weights of unsupervisedSynapse

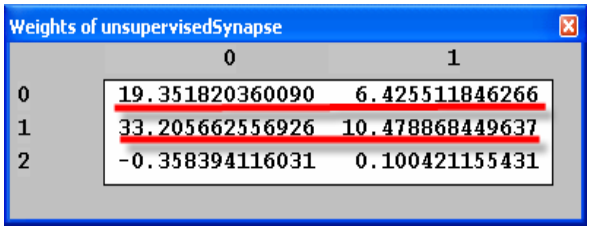
	0	1	2
0	-0.451506088443	0.463896603290	0.436979277932
1	-0.466643269143	0.177205725272	0.006576738792
2	25.641225758815	6.267486735499	8.269150221939

Two input variables and 4 MSE=0.0079

Weights of unsupervisedSynapse

	0	1
0	-0.219809564501	0.099536118656
1	19.351824656801	6.425514018775
2	33.205695811530	10.478871907419
3	-0.105517746513	-0.362300485244

Two input variables and 3 MSE=0.008



Conclusion

A radial basis function neural network has been trained with a few patterns in order to forecast the volume of wood in a given forest area. The network performs a clustering process of the trees using different input variables. A sensitive analysis can be computed observing the weight of unsupervised synapse. A previous clustering process of input data permits a better forecasting process in the output variable, in our case the amount of volume of wood in a forest area. These results improve commercial and classical forecasting methods in forest inventories, and proposed method can be applied to any tree specie or forest area without taking into account environment variables that appears in classical mathematical equations. As the number of classes that needs to be discriminated decreases, classifier accuracy increases; until obtain the real number of classes. Once the correct number of classes has been obtained using the RBF and with a supervised learning the volume of wood for a forest inventory can be estimated.

Bibliography

[Fritzke 1994] Fritzke B. Supervised learning wiht growing cell structures. Advances in neural information processing system 6, Pp. 25. Morgan Kaufmann Publishers 1994.

[Hamilton, F. and Brack, C.L. 1999] Hamilton, F. and Brack, C.L. 1999. Stand volume estimates for modelling inventory data. Australian Forestry 62(4): 360 - 367

[Kphonen 1989] Kohonen T. Self organization and associative memory. Springer-Verlag 1989

[Moody and Darken, 1989] Moody J., Darken C. Fast learning in networks of locally-tunned processing units. Neural computation, vil1, pp 281-294. 1989.

[Poggio and Girosi, 1990] Poggio T. and Girossi F. Networks for approximation and learning. IN procceding of the IEEE, volume 78, pages 1481-1497.

[Schreuder, H.T., Gregoire, T.G. and Word, G.B. 1993] Schreuder, H.T., Gregoire, T.G. and Wood, G.B. 1993. Sampling Methods for Multiresource Forest Inventory. John Wiley and Sons, Inc. New York.

Authors' Information

Angel Castellanos – Dpto. de Ciencias Basicas aplicadas, a la Ingeniería Forestal. Escuela Univ. de Ingeniería Técnica Forestal. Universidad Politécnica de Madrid, Avda. de Ramiro de Maeztu s/n 28040 Madrid, Spain. e-mail: angel.castellanos@upm.es

Ana Martinez Blanco – Dpto. de Ciencias Basicas aplicadas, a la Ingeniería Forestal. Escuela Univ. de Ingeniería Técnica Forestal. Universidad Politécnica de Madrid, Avda. de Ramiro de Maeztu s/n 28040 Madrid, Spain. e-mail: ana.martinez@upm.es

Valentín Palencia – Dpto. de Arquitectura y Tecnología de Sistemas Informáticos. Facultad de Informática. Universidad Politécnica de Madrid (Campus de Montegancedo) 28600 Boadilla del Monte, Madrid, Spain. e-mail: vpalencia@fi.upm.es

DECISION TREES FOR APPLICABILITY OF EVOLUTION RULES IN TRANSITION P SYSTEMS

Luis Fernandez, Fernando Arroyo, Ivan Garcia, Gines Bravo

Abstract: Transition P Systems are a parallel and distributed computational model based on the notion of the cellular membrane structure. Each membrane determines a region that encloses a multiset of objects and evolution rules. Transition P Systems evolve through transitions between two consecutive configurations that are determined by the membrane structure and multisets present inside membranes. Moreover, transitions between two consecutive configurations are provided by an exhaustive non-deterministic and parallel application of active evolution rules subset inside each membrane of the P system. But, to establish the active evolution rules subset, it is required the previous calculation of useful and applicable rules. Hence, computation of applicable evolution rules subset is critical for the whole evolution process efficiency, because it is performed in parallel inside each membrane in every evolution step. The work presented here shows advantages of incorporating decision trees in the evolution rules applicability algorithm. In order to it, necessary formalizations will be presented to consider this as a classification problem, the method to obtain the necessary decision tree automatically generated and the new algorithm for applicability based on it.

Keywords: Decision Tree, ID3, Evolution Rules, Applicability, Transition P System.

ACM Classification Keywords: I.2.6 Learning – Decision Tree; D.1.m Miscellaneous – Natural Computing

Introduction

Membrane computing is a new computational model based on the membrane structure of living cells [Păun, 1998]. This model has become, during last years, a powerful framework for developing new ideas in theoretical computation. Main idea was settled in the base of connecting the Biology with Computer Science in order to develop new computational paradigms.

An overview of membrane computing software can be found in literature, or tentative for hardware implementations [Fernández, 2005], or even in local networks is enough “to understand how difficult is to implement membrane systems on digital devices” [Păun, 2005].

Transition P Systems evolve through transitions between two consecutive configurations that are determined by the membrane structure and multisets present inside membranes. Moreover, transitions between two consecutive configurations are provided by an exhaustive non-deterministic and parallel application of an evolution rules subset inside each membrane of the P system. Evolution rules subset we are studding here will be composed by applicable rules. Moreover, It exist algorithms of application for evolution rules [Fernández, 2006] that, recurrently to its end, need the computation of applicable evolution rules subset. Hence, computing applicable evolution rules is critical for the whole evolution process efficiency, because it is performed in parallel inside each membrane in each one of the evolution steps.

At the present time, computation of applicable evolution rules subset falls on redundancies in a directly or indirectly way. Incorporating decision trees in this computation avoids these redundancies and improves global efficiency of P system evolution.

This work is structured as follows: firstly, evolution rules applicability over a multiset of objects problem is formalized together with its corresponding traditional algorithm. Following section, briefly describes essential elements of decision trees. Afterwards, they are presented new formalizations that permit considering applicability problem as a classification problem solvable through decision trees. In next section, it is presented the algorithm based on decision trees. Finally, efficiency between both algorithms is compared and we expose our conclusions.

Applicability of Evolution Rules

This section defines concepts about multisets, evolution rules and applicability which are needed to follow the developed work presented here. Moreover, it is presented the traditional algorithm, without decision trees, for applicability evolution rules on multisets and its complexity.

From now on, let U be a finite and not empty set of symbols with $|U| = m$.

Let ω be a multiset over U, where ω is a mapping from U to N. Hence, $\omega(u) = p / \forall u \in U \; \exists! \, p \in N$. Let us present the set of all multisets as $\mathcal{M}(U) = \{\omega / \omega \; is \; a \; multiset\}$.

Weight of a symbol $u \in U$ is defined over a multiset $\omega \in \mathcal{M}(U)$ as $\omega(u)$ and it is represented by $|\omega|_u$.

Inclusion of multiset is a binary relation defined as $\omega_1 \subset \omega_2 \Leftrightarrow |\omega_1|_u \leq |\omega_2|_u$, $\forall u \in U \;\; \forall \omega_1, \omega_2 \in \mathcal{M}(U)$.

Any $\omega \in \mathcal{M}(U)$ can be represented as the m-tuple of natural number by the Parikh vector associated to the multiset w with respect to U. The problem is that the Parikh vector representation depends on the order of the elements of U. To avoid this problem, an order over the set U is defined as an ordered succession of symbols through a one to one mapping Φ from $\{1..m\}$ to U that is:

1. $\forall i \in \{1,...,m\} \; \exists u \in U / \Phi(i) = u$
2. $\forall u \in U \; \exists i \in \{1,...,m\} / \Phi(i) = u$
3. $\forall i, j \in \{1,...,m\} / \Phi(i) = \Phi(j) \Rightarrow i = j$

This fact permits us to represent every $\omega \in \mathcal{M}(U)$ as an element of N^m in a congruent manner. Hence,

$\omega = (p_1,..., p_m) \in N^m / |\omega|_u = p_{\Phi(u)} \; \forall u \in U$.

On the other hand, let T be a finite and non empty set of targets.

Evolution rule with symbols in U and targets in T is defined by $r = (a, c, \delta)$ where $a \in \mathcal{M}(U)$, $c \in \mathcal{M}(U \times T)$ and $\delta \in$ {dissolve, not dissolve}. The set of evolution rules is defined as $\mathcal{R}(U,T) = \{r / r \; is \; a \; evolution \; rule\}$.

Antecedent of $r = (a, c, \delta) \in \mathcal{R}(U,T)$ is defined as $input(r) = a$.

Finally, it is said that $r \in \mathcal{R}(U,T)$ is applicable over $\omega \in \mathcal{M}(U)$ if and only if $input(r) \subset \omega$.

Applicability Algorithm. On the one hand, a set of useful evolution rules R and a multiset of objects ω, will be the input to the process. On the other hand, output of process will be R_A, the evolution rules subset of R that are applicable over the multiset. Traditional algorithm [Fernández, 2005] checks weights of each evolution rules antecedent symbol with the corresponding from multiset of objects.

```
(1) R_A ← ø
(2) FOR-EACH r_i IN R DO BEGIN
(3)     j ← 1
(4)     WHILE j ≤ |ω| − 1 AND |input(r_i)|_j ≤ |ω|_j DO
(5)         j ← j + 1
(6)     IF |input(r_i)|_j ≤ |ω|_j THEN
(7)         R_A ← R_A ∪ {r_i}
(8) END
```

Algorithm 1. Evolution rules applicability (without decision trees).

Complexity of algorithm 1 consider, in the worst case, situation in which every evolution rule are applicable over the multiset of objects: loop in (4) will reach as many iterations as symbols exists in U on each iteration of loop (2) to each evolution rule present in R. In the worst case, complexity order will be $O(n)$ being $n = |R| \cdot |U|$

Analysis of previous algorithm will reveal possible redundancies in checks: in a direct and indirect way. So,

- A redundant check in a direct way will occur when weight of a same symbol is equal in more than one evolution rule antecedent, executing several times the same comparison (for example, let be $input(r_1) = (3, 1, 4, 1)$, $input(r_2) = (3, 2, 4, 4)$, and $\omega = (7, 3, 5, 4)$ where comparisons for the fist and third symbol of $input(r_2)$ are redundant in a direct way with its respective symbols in $input(r_1)$).
- A redundant check in an indirect way will occur when, after result of a checking which is false, it will be performed checks between greater weights of that symbol in others evolution rules antecedent (for instance, let it be $input(r_1) = (3, 1, 3, 1)$, $input(r_2) = (5, 2, 5, 1)$, and $\omega = (1, 3, 5, 4)$ where comparison for first symbol of $input(r_2)$ is redundant in an indirect way with its respective symbol in $input(r_1)$).

Furthermore, any checking of the weight of a symbol from an evolution rule antecedent with 0 will be unnecessary because $0 \leq n \ \ \forall n \in N$.

Decision Trees

A decision tree is a tree that permits us to determine the class which one element belongs to, depending on the values that take some attributes of it. Every internal node represents one attribute and edges are possible values of that attribute. Every leaf node in the tree represents one class. So, one unknown element can be classified processing the tree: every internal node studies the value of one attribute for the element and takes the appropriate edge, depending on its value; it continues until a leaf node is reached and, therefore, to the element classification.

E	a_1	...	a_j	...	a_q	C
e_1	v_{11}	...	v_{1j}	...	v_{1q}	c_1
...	...	...	...	...	...	...
e_i	v_{i1}	...	v_{ij}	...	v_{iq}	c_j
...	...	...	...	...	...	...
e_p	v_{p1}	...	v_{pj}	...	v_{pq}	c_s

Figure 1. Example of values table for ID3 algorithm input

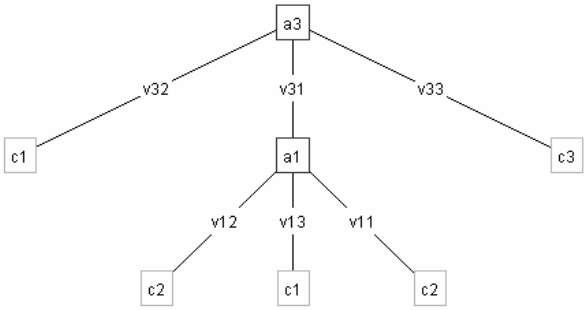


Figure 2. Decision tree generated by ID3 for values table from fig 1.

There are a lot of algorithms to generate decision trees [Rasoul, 1991]. In particular, ID3 algorithm is based on entropy information and it generates an optimum decision tree from a non incremental set of instances and without repetitions.

ID3 algorithm requires as input (Fig. 1): let E be a finite set of instances $\{e_1,...,e_p\}$; let A be a finite set of attributes $\{a_1,...,a_q\}$;let V_j be a finite set of values $\{v_{1j},...,v_{rj}\}$ for each attribute a_j, (where a_j attribute value for instance e_i fulfils $v_{ij} \in V_j$); and finally, let C be a finite set of classes for the classification $\{c_1,...,c_s\}$. On the other hand, ID3 algorithm outputs the optimum decision tree for any element classification (Fig. 2).

Decision Trees for Applicability of Evolution Rules

This section presents evolution rules applicability as a classification problem. This way, it will be posible to design a new algorithm that, being based on a decision tree, avoid direct and indirect redundancies of algorithm 1 presented above.

In order to it, we invert evolution rules applicability problem terms: for a given multiset, we compute the applicable evolution rules subset. Hence, we consider:

- Multisets of objects will be the elements to be classified: $\omega = (p_1,..., p_m) \in \mathcal{M}$(U);
- The set of attributes will be a settled as a set of checks between the objects weights from the multiset and the same object from the evolution rules antecedents having a non null weight. Hence, the finite set of attributes will be:

$$A = \{a \equiv | \omega |_u \geq k \ / \ | input(r) |_u = k \wedge k \neq 0 \ \ \exists r \in R \ \forall u \in U \ \};$$

- Consequently, the finite set of values for every attribute will be true or false, result from comparison relationship between weights.
- Finally, classes to consider will be the different applicable evolution rules subsets. Therefore, the finite set of classes will be: $C = \{c \equiv R_A / \exists R_A \subset R \ \}$.

To obtain automatic generation of decision tree from ID3 algorithm, it will be necessary a non incremental and without repetitions battery of finite instances. In order to it, domain is defined as a set of multisets having the same values for all of their attributes. Consequently, each domain is characterized because every multiset responds to the same applicable evolution rules subset, that is, to the same class. Finally, examples battery will be formed by a representative from each domain.

Fig. 3 shows an example with disjoint domains of multisets of symbols for $U = \{x, y\}$ and rules set: $R = \{r_1, r_2, r_3, r_4\}$ where *their antecedents are:* $r_1 = (y^5)$, $r_2 = (x^2, y^2)$, $r_3 = (x^6, y^2)$, $r_4 = (x^2, y^3)$.

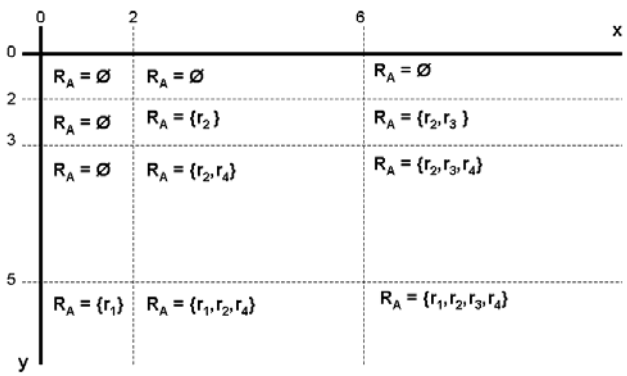


Fig. 3. Disjoint domains of multisets of objects for rules set and its corresponding applicable evolution rules.

Next, they are presented necessary definitions for formalizing the finite set of representative domains that are needed for the generation of decision trees.

It is defined projection of $u \in U$ over $R \subset \mathcal{P}(\mathcal{K}$(U,T)) as:

$$P_u(R) = \{n \in N / \exists r \in R \wedge | input \ r |_u = n\} \subset \mathcal{P}(N)$$

Hyperplane of $d \in \{1,...,m\}$ in $k \in N$ over N^m is defined as:

$$H_d^k(N^m) \rightarrow \{(x_1,..., x_d,..., x_m) / x_d = k\} \subset \mathcal{P}(N^m)$$

Thus, it is considered the grid over $R \subset \mathcal{P}(\mathcal{K}$(U,T)) as:

$$\mathcal{H}(R) = \{h / h = H_{\Phi(u)}^k \ \forall u \in U \wedge \forall k \in P_u(R)\}$$

Moreover, $\mathcal{D}(R)$ is defined as the partition N^m in disjoint subsets formed from every hyperplane of the grid $\mathcal{H}(R)$. It is named domain D to each one of the elements from partition $\mathcal{D}(R)$. Where it is fulfilled:

$$d = | \mathcal{D}(R) | = \prod_{\forall u \in U} | P_u(R) | < \infty$$

$$N^m = \bigcup_{k=1}^{d} D_k \ \wedge \ D_i \cap D_j \neq \emptyset, \forall i, j \in \{1,...,d\} \wedge i \neq j$$

Finally, it is defined representative of $D \in \mathcal{D}(R)$ as

$$\Lambda(D) = \min(\{dist(m,(0,...,0)) \ \forall m \in D\})$$

Fig. 4 shows an example obtained from values of table for the evolution rules set of figure 3. This table includes a row by each representative of the domains, its values for checking relations and applicable evolution rules subset that corresponds to each domain. Fig. 5 shows the classification tree generated by ID3 algorithm for the corresponding figure 4 values table.

Incorporation of decision trees avoids unnecessary null weights comparisons from algorithm 1 because they are not incorporated as in starting instances. Same, direct way redundancies are avoided, the weight of a symbol is compared with the same value just once. Finally, indirect way redundancies are also avoided due to the optimum decision tree ensured by ID3 algorithm, avoiding relations of transitive comparisons.

E	x≥6	x≥2	y≥5	y≥3	y≥2	C
$x^0 y^0$	no	no	no	no	no	Ø
$x^0 y^2$	no	no	no	no	yes	Ø
$x^0 y^3$	no	no	no	yes	yes	Ø
$x^0 y^5$	no	no	yes	yes	yes	$\{r_1\}$
$x^2 y^0$	no	yes	no	no	no	Ø
$x^2 y^2$	no	yes	no	no	yes	$\{r_2\}$
$x^2 y^3$	no	yes	no	yes	yes	$\{r_2,r_4\}$
$x^2 y^5$	no	yes	yes	yes	yes	$\{r_1,r_2,r_4\}$
$x^6 y^0$	yes	yes	no	no	no	Ø
$x^6 y^2$	yes	yes	no	no	yes	$\{r_2,r_3\}$
$x^6 y^3$	yes	yes	no	yes	yes	$\{r_2,r_3,r_4\}$
$x^6 y^5$	yes	yes	yes	yes	yes	$\{r_1,r_2,r_3,r_4\}$

Fig. 4. Examples battery for the evolution rules set of figure 3: in each row there is representative of each domain, values it takes for comparison relations and corresponding applicable evolution rules subset.

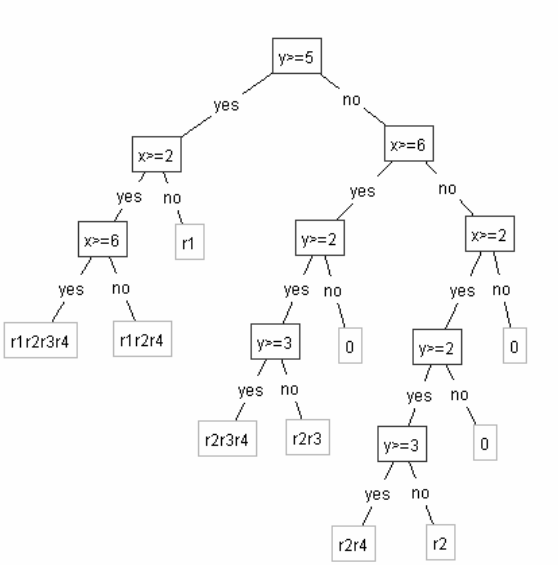


Fig. 5. Example of decision tree generated by ID3 algorithm for the examples battery from figure 4.

Applicablity Algorithm based on Decision Trees

Previously to the algorithm presentation, we will expose the appropriate data structure for supporting the decision tree.

- On the one hand, they are disposed four correlative tables *left*, *symbol*, *value* and *right* for attribute nodes, with one cell in each table by each attribute node; root node is located in position 0 cells;
- On the other, It is disposed a table *classes* for classification nodes, with one cell for each classification node;
- Correlative cells of tables *symbol* and *value* determine which object weight from the multiset of objects has to be compared with which weight. Cells of tables *left* y *right* indicate, whether or not it is respectively accomplished previous relation comparison, which cell is the following attribute node in, whether index is positive; otherwise, indicate which cell of classification nodes table is the solution in.

Figure 6 shows an example of data structures of corresponding generated decision tree from figure 3.

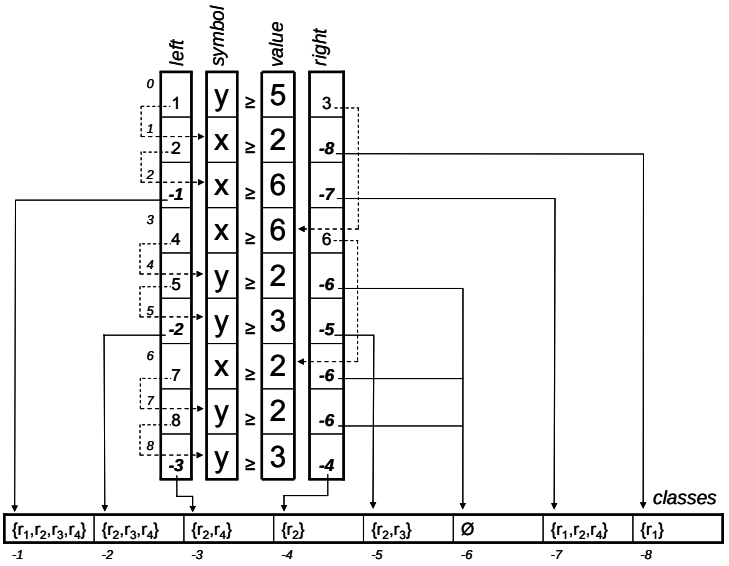


Fig. 6. Data structure for decision tree of figure 5.

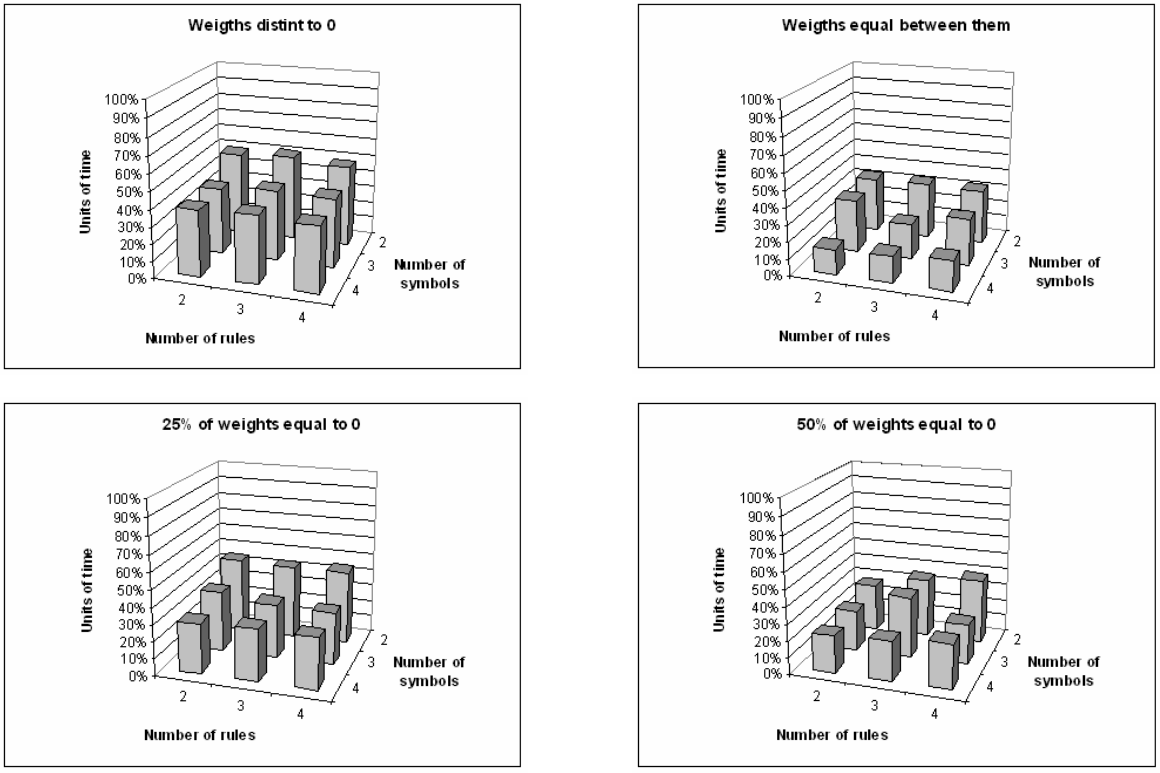


Fig. 7. Execution time reduction carried out by applicability algorithm based on decision tree respect traditional algorithm.

Then, the input to applicability algorithm is ω, multisets of objects, and the supporting decision tree data structure: *left*, *symbol*, *value*, *right* and *classes*. On the other hand, output is A, an evolution rules subset of R that is applicable over that multiset. Following code processes rows of the indexes of branches *left* or *right*, depending on the comparison of symbol indicated by *symbol* weight with established value on *value* until it is reached a classification leaf, indicated by a negative value.

(1) $f \leftarrow 0$
(2) **WHILE** $f \geq 0$ **DO**
(3) **IF** $|\omega|_{\Phi^{-1}(symbol[f])} \geq value[f]$ **THEN**
(4) $f \leftarrow left[f]$
(5) **ELSE**
(6) $f \leftarrow right[f]$
(7) $A \leftarrow applicable[f]$

Algorithm 2. Evolution rules applicability based on decision trees.

In the worst case, complexity of algorithm 2 considers to process the longest branch of the decision tree which length will always be lesser than $n = |R| \cdot |U|$. We will reach this conclusion by reduction to the absurd: in order to the longest branch of the decision tree requires n attribute nodes, It must be carried out the following a) every weight of each symbol in the antecedent of evolution rules is not null and different between them and, b) the d existing domain leads to applicable evolution rules different subsets. That is impossible because, in such circumstances, always exist more than one domain that would lead to the empty set. Specifically, a number of domains equal to $\left(\sum_{i=1}^{|U|} |P_i(R)| \right) - |U| + 1$.

Comparative

This section presents the experimental results obtained from evolution rules applicability using the two algorithms presented here. The test set has been randomly generated and it is composed by 48 different evolution rules sets (composed between 2 and 4 evolution rules composed between 2 and 4 symbols per antecedent), over these tests, it has been calculated the applicability of more than a million of randomly generated symbols multisets.

A first global analysis presents a reduction of execution time of this new algorithm based on decision trees respect to traditional algorithm in an average of 33%, with a variance of 7%.

Particularly, they has been made tests directed to four different situations to analyse the behaviour of new algorithm in extreme cases: with every different weight in antecedents of evolution rules, with every weight of same value, and with presence of 25% and 50% null weights.

In the worst case, with every weight being different between them, it has been reached at least 50% of execution time reduction. With all weights with same value, execution time is reduced to a 15% (for 4 evolution rules with 4 symbols per antecedent). In presence of 25% and 50% of null weights in antecedents of evolution rules, time is reduced to a 35% and a 29%, respectively, always in favor of the new algorithm with decision trees.

Conclusions

This work presents a new approach to the calculus of evolution rules applicability over a symbols multiset. This approach is based on decision trees generated from the set of evolution rules of a membrane. This way, they are avoided unnecessary and redundant checking in a direct or indirect way. Consequently, It is always obtained a lesser complexity than the corresponding traditional algorithm. So, execution time is optimized in the calculation of evolution rules applicability over a symbols multiset. All of this has repercussions in global efficiency of the P System evolution, because applicability calculation is carried out in parallel in each membrane in each evolution step.

Bibliography

[Fernández, 2005] L.Fernández, V.J.Martínez, F.Arroyo, L.F.Mingo, *A Hardware Circuit for Selecting Active Rules in Transition P Systems*, Workshop on Theory and Applications of P Systems. Timisoara (Rumanía), september, 2005.

[Fernández, 2006] L.Fernández, F.Arroyo, J.Castellanos, J.A.Tejedor, I.García, *New Algorithms for Application of Evolution Rules based on Applicability Benchmarks*, BIOCOMP06 International Conference on Bioinformatics and Computational Biology, Las Vegas (USA), july, 2006 (accepted).

[Paun, 1998] Gh.Paun, *Computing with Membranes*, Journal of Computer and System Sciences, 61(2000), and Turku Center of Computer Science-TUCS Report nº 208, 1998.

[Paun, 2005] Gh.Paun, *Membrane computing. Basic ideas, results, applications*, Pre-Proceedings of First International Workshop on Theory and Application of P Systems, Timisoara, Romania, September 26-27, 2005, 1-8

[Rasoul, 1991] S.R.Safavian, D.Landgrebe, *A Survey of Decision Tree Classifier Methodology*, IEEE Transactions on Systems, Man, and Cybernetics, Vol. 21, No. 3, pp 660-674, May 1991.

Authors' Information

Luis Fernandez – Natural Computing Group of Universidad Politécnica de Madrid (UPM); Ctra. Valencia, km. 7, 28031 Madrid (Spain); e-mail: setillo@eui.upm.es

Fernando Arroyo – Natural Computing Group of Universidad Politécnica de Madrid (UPM); e-mail: farroyo@eui.upm.es

Ivan Garcia – Natural Computing Group of Universidad Politécnica de Madrid (UPM); e-mail: igarcia@eui.upm.es

Gines Bravo – Natural Computing Group of Universidad Politécnica de Madrid (UPM); e-mail: gines@eui.upm.es

A PARTITION METRIC FOR CLUSTERING FEATURES ANALYSIS

Dmitry Kinoshenko, Vladimir Mashtalir, Vladislav Shlyakhov

Abstract: A new distance function to compare arbitrary partitions is proposed. Clustering of image collections and image segmentation give objects to be matched. Offered metric intends for combination of visual features and metadata analysis to solve a semantic gap between low-level visual features and high-level human concept.

Keywords: partition, metric, clustering, image segmentation.

ACM Classification Keywords: I.5.3 Clustering - Similarity measures

Introduction

There has been a tremendous growth of the image content analysis significance in the recent years. This interest has been motivated mainly by the rapid expansion of imaging on the World-Wide Web, the availability of digital image libraries, increasing of multimedia applications in commerce, biometrics, science, entertainments etc. Visual contents of an image such as color, shape, texture and region relations play dominating role in propagation of feature selection, indexing, user query and interaction, database management techniques. Many systems combine visual features and metadata analysis to solve a semantic gap between low-level visual features and high-level human concept, i.e. there arises a great need in self-acting content-based image retrieval task-level systems.

To search images in an image database traditionally queries ‘ad exemplum’ are used. In this connection essential efforts have been devoted to synthesis and analysis of image content descriptors. However, a user’s semantic understanding of an image is of a higher level than the features representation. Low-level features with mental concepts and semantic labels are the groundwork of intelligent databases creation. Short retrieval time

independent of the database size is a fundamental requirement of any user friendly content-based image retrieval (CBIR) system. Characteristics of different CBIR schemes, similarities or distances between the feature vectors of the query by example or sketch and those of the database images are sufficiently full explored [see, e.g. 1-3]. To optimize CBIR schemes it is necessary to minimize a total number of matches at a retrieval stage. Thus there arises a problem of finding novel partition measures for the fast content-based image retrieval in video databases and we have to be able to compare different partitions obtained for instance as a result of segmentation.

Motivation to Synthesis of a Partition Metric

As retrieval is computationally expensive, one of the most challenging moments in CBIR is minimizing of the retrieval process time. Widespread clustering techniques allow to group similar images in terms of their features proximity. The number of matches can be greatly reduced, but there is no guarantee that the global optimum solution is obtained. We propose clustering of image collections with objective function encompassing goals to number of matches at a search stage.

The problem is in that under given query $y \in \mathrm{Y}$ one needs to find the most similar image (or images) $x_v \in \mathrm{X}$. In other words, it is necessary to provide $min_{v \in \mathrm{V}} \rho(y, x_v)$ (here $\rho(\circ,\circ)$ is arbitrary distance function, V is an indexing set) during minimum possible warranted time. If $\mathrm{Y} \subseteq \mathrm{X}$, retrieval by exact matching is required. We shall name elements $[\mathrm{X}]_\alpha$, $\alpha \in \mathrm{A}$ of power set 2^{X} as clusters if they correspond to the partition of set X. Let us consider such partitions that any elements of one cluster do not differ from each other more than on ε, i.e. $\forall x' \neq x''$ we have $[x'] = [x'']$, if $\rho(x', x'') \leq \varepsilon$ and $[x'] \cap [x''] = \varnothing$ otherwise. The given or obtained value ε used at a clustering stage is connected with required accuracy of retrieve δ, if it is specified, as follows. There arise two cases:

$\delta > \varepsilon$ – any representative of the cluster nearest to the query y can be used as the image retrieval result, i.e. minimal number of matches is defined by the number of clusters; in other words it is necessary to provide

$$N_1 = card\ \{[\mathrm{X}]_\alpha\} \rightarrow min; \tag{1}$$

$\delta \leq \varepsilon$ – the element of more detailed partition will be the result of the image retrieval. In simplest situations it is necessary to fulfill a single-stage clustering, i.e. to optimize retrieval under worst-case conditions we have to ensure

$$N_2 = card\{[\mathrm{X}]_\alpha\} + max(card\ [\mathrm{X}]_\alpha) \rightarrow min. \tag{2}$$

At the multilevel clustering the repeated clusters search inside of already retrieved clusters is fulfilled and only on the last step required image is searched by complete enumeration. Let us assume that the cluster $[\mathrm{X}^{(i-1)}]_p$ is selected on $(i-1)$ level of hierarchy from a condition $\rho(y, [\mathrm{X}^{(i-1)}]_q) \rightarrow min$, $q = \overline{1, card\{[\mathrm{X}^{(i-1)}]\}}$, i.e. $[\mathrm{X}^{(i-1)}]_p = = [\mathrm{X}^{(i)}]_1 \cup [\mathrm{X}^{(i)}]_2 \cup \ldots \cup [\mathrm{X}^{(i)}]_{\alpha_p}$ where for any k and l the equality $[\mathrm{X}^{(i)}]_k \cap [\mathrm{X}^{(i)}]_l = \varnothing$ holds. Then the minimization of matches amount is reduced to the clustering with the goal function

$$N_3 = \sum_{i=1}^{m-1} \left\{ card\ [\mathrm{X}^{(i)}]_{p,(i)} \middle| x \in [\mathrm{X}^{(i-1)}]_{p,(i-1)} \right\} + max\left(card\ [\mathrm{X}^{(m-1)}]_{p,(m-1)} \right) \rightarrow min, \tag{3}$$

where m is a number of hierarchy levels, $[\mathrm{X}^{(0)}]_{1,(0)} = \mathrm{X}$. Minimization task (1) was solved in [4], the solution of problem (2) was offered in [5], searching of (3) one could see in [6].

Content of an image may be often summarized by a set of homogeneous regions in appropriate feature space. Therefore, there is a great need for automatic tools to classify and retrieve image content on the base of segmentation.

Segmented images are formed from an input image by gathering its elements into sets likely to be associated with meaningful objects in the scene. That is, the main segmentation (clustering) goal is to partition the entire image into disjoint connected or disconnected regions. Unfortunately, the effectiveness of their direct interpretation depends heavily on the application area and characteristics of an acquisition system. Possible high-

level region-based interpretations are associated with a priori information, measurable region properties, heuristics, plausibility of computational inference. Whatever the case, often it is necessary to have dealings with a whole family of partitions and we must be able to compare these partitions, which are produced by a variety of segmentation algorithms. At least splitting and merging techniques make us to match segmentation results, which ultimately may be corresponded to indirectly images comparisons.

For region-based similarity analysis novel approaches are required since usually early processing scheme consists of following steps: images are segmented into disjoint (or weakly intersecting) regions, features are extracted from each region, and the set of all features is used for high-level processing. It should be emphasized that quite often simultaneous processing of partitions or coverings is wanted to produce reliable true conclusion. In this connection we propose and vindicate a new metric providing arbitrary partitions (and consequently images) matching. Generally, these partitions are results of any clustering procedures.

A Metric for Partitions Matching

Let Ω be arbitrary measurable set with given measure $\mu(\circ)$, which can be interpreted as length, area, volume, mass distribution, probability distribution, cardinality, etc. Consider a set Π_Ω of finite (from the standpoint of the factor set cardinality) partitions of Ω, i.e. $\alpha \in \Pi_\Omega$, $\alpha = \{X_1, X_2, \ldots, X_n\}$, $X_i \subseteq \Omega$, $i = \overline{1,n}$, $\Omega = \bigcup_{i=1}^{n} X_i$, $\forall i, j \in \{1,2,\ldots,n\} : i \neq j \Rightarrow X_i \cap X_j = \varnothing$, $\forall i \in \{1,2,\ldots,n\} \Rightarrow \mu(X_i) < \infty$. We denote subsidiary partitions which will be used in further proofs as $\beta = \{Y_1, Y_2, \ldots, Y_m\}$ and $\gamma = \{Z_1, Z_2, \ldots, Z_l\}$.

Note, these partitions could be image segmentation results, representing pairwise disjoint family of nonempty subsets whose union is the image and each subset may contain required target, may belong to a carrier of object image or may be a part of it. Generally, partitions are results of arbitrary clustering.

Let us introduce on $\Pi_\Omega \times \Pi_\Omega$ the functional

$$\rho(\alpha,\beta) = \sum_{i=1}^{n} \sum_{j=1}^{m} \mu(X_i \Delta Y_j) \mu(X_i \cap Y_j) \tag{4}$$

(here $X_i \Delta Y_j = (X_i \setminus Y_j) \cup (Y_j \setminus X_i)$ is a symmetric difference) and prove that the functional $\rho(\alpha,\beta)$ may be interpreted as a distance function for partitions matches. Before we verify that (4) is a metric we shall begin with subsidiary formal propositions.

Lemma 1. The functional $\rho(\alpha,\beta)$ can be represented in the tantamount form

$$\rho(\alpha,\beta) = \sum_{i=1}^{n} [\mu(X_i)]^2 + \sum_{j=1}^{m} [\mu(Y_j)]^2 - 2 \sum_{i=1}^{n} \sum_{j=1}^{m} [\mu(X_i \cap Y_j)]^2 . \tag{5}$$

Proof. It should be noted that for arbitrary measurable sets P and Q the equality

$$\mu(P \Delta Q) = \mu(P) + \mu(Q) - 2\mu(P \cap Q) \tag{6}$$

takes place. Indeed, from the definition of a symmetric difference we directly get $P = (P \setminus Q) \cup (P \cap Q)$ and $Q = (Q \setminus P) \cup (Q \cap P)$ where sets $P \setminus Q$ and $P \cap Q$ do not intersect. Then by virtue of measure $\mu(\circ)$ is additive we have

$$\begin{cases} \mu(P) = \mu(P \setminus Q) + \mu(P \cap Q), \\ \mu(Q) = \mu(Q \setminus P) + \mu(P \cap Q). \end{cases} \tag{7}$$

Adding up equations (7) and taking into account that $P \Delta Q = (P \setminus Q) \cup (Q \setminus P)$ we obtain

$$\mu(P) + \mu(Q) = \mu(P \setminus Q) + \mu(Q \setminus P) + 2\mu(P \cap Q) = \mu(P \Delta Q) + 2\mu(P \cap Q) . \tag{8}$$

It is clear that (8) and (6) are equivalent equations.

With due regard (6) we could rewrite (4) with reference to X_i and Y_j as

$$\rho(\alpha,\beta) = \sum_{i=1}^{n}\sum_{j=1}^{m}\mu(X_i \cap Y_j)[\mu(X_i)+\mu(Y_j)-2\mu(X_i \cap Y_j)] =$$
$$= \sum_{i=1}^{n}\sum_{j=1}^{m}\mu(X_i)\mu(X_i \cap Y_j)+\sum_{i=1}^{n}\sum_{j=1}^{m}\mu(Y_j)\mu(X_i \cap Y_j)-2\sum_{i=1}^{n}\sum_{j=1}^{m}[\mu(X_i \cap Y_j)]^2 = \quad (9)$$
$$= \sum_{i=1}^{n}\mu(X_i)\sum_{j=1}^{m}\mu(X_i \cap Y_j)+\sum_{j=1}^{m}\mu(Y_j)\sum_{i=1}^{n}\mu(X_i \cap Y_j)-2\sum_{i=1}^{n}\sum_{j=1}^{m}[\mu(X_i \cap Y_j)]^2 .$$

In (9) we distinguish components $\sum_{j=1}^{m}\mu(X_i \cap Y_j)$ and $\sum_{i=1}^{n}\mu(X_i \cap Y_j)$. Emphasize sufficiently obvious but significant property of similar sums for all $i \in \{1,2,\ldots,n\},\ \ j \in \{1,2,\ldots,m\}$

$$\begin{cases} \sum_{j=1}^{m}\mu(X_i \cap Y_j) = \mu(X_i), \\ \sum_{i=1}^{n}\mu(X_i \cap Y_j) = \mu(Y_j). \end{cases} \quad (10)$$

The correctness of (10) immediately follows from fig. 1 (according to $\beta,\alpha \in \Pi_\Omega$ and measure $\mu(\circ)$ additivity).

Substituting (10) into (9) we finally get

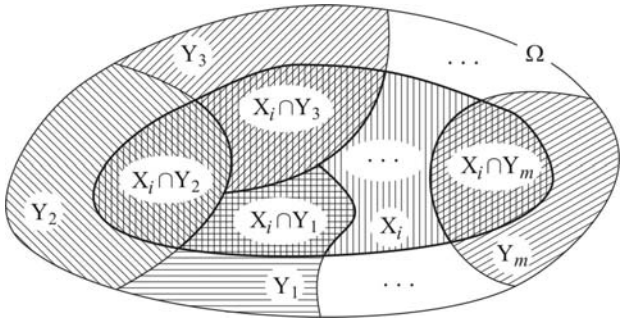


Fig. 1. To explanation of any factor set element intersections with any partition

$$\rho(\alpha,\beta) = \sum_{i=1}^{n}\sum_{j=1}^{m}\mu(X_i \Delta Y_j)\mu(X_i \cap Y_j) = \sum_{i=1}^{n}[\mu(X_i)]^2 + \sum_{j=1}^{m}[\mu(Y_j)]^2 - 2\sum_{i=1}^{n}\sum_{j=1}^{m}[\mu(X_i \cap Y_j)]^2,$$

which was required, i.e. we get equivalent form of functional (4).

Let us introduce triple intersections of partition elements into consideration, viz sets of following kind

$$X_i \cap Y_j \cap Z_k,\ \ i = \overline{1,n},\ \ j = \overline{1,m},\ \ k = \overline{1,l}.$$

Lemma 2. Values $\nu_{ijk} = (X_i \cap Y_j \cap Z_k)$ are in accordance with the equalities

$$\mu(Z_k) = \sum_{i=1}^{n}\sum_{j=1}^{m}\nu_{ijk}\ , \quad (11)$$

$$\mu(X_i \cap Y_j) = \sum_{k=1}^{l}\nu_{ijk}\ , \quad (12)$$

$$\mu(X_i \cap Z_k) = \sum_{j=1}^{m}\nu_{ijk}\ , \quad (13)$$

$$\mu(Y_j \cap Z_k) = \sum_{i=1}^{n}\nu_{ijk}\ . \quad (14)$$

Proof. First let us consider $\alpha,\beta \in \Pi_\Omega$. Notice that $\alpha \cap \beta = \{X_i \cap Y_j\}_{i=\overline{1,n},\ j=\overline{1,m}} \in \Pi_\Omega$, i.e. an intersection of partitions is a partition also. Indeed, if we choose arbitrary element $\varpi \in \Omega$ we get that, owing to α,β are partitions, there exist indices $i \in \{1,2,\ldots,n\},\ j \in \{1,2,\ldots,m\}$ s.t. $\varpi \in X_i$ and $\varpi \in Y_j$ or $\varpi \in X_i \cap Y_j$, i.e.

$$\bigcup_{i=\overline{1,n},\ j=\overline{1,m}}(X_i \cap Y_j) = \Omega\,. \quad (15)$$

On the other hand, we can write $(X_i \cap Y_j)\cap(X_{i'} \cap Y_{j'}) = (X_i \cap X_{i'})\cap(Y_j \cap Y_{j'})$ due to virtue of set intersection associativity and commutativity. Since pairs $(i,\ j)$ and (i',j') do not coincide, inequalities $i \neq i'$ and

$j \neq j'$ hold separately or both. Thereby at least one of sets $X_i \cap X_{i'}$ or $Y_j \cap Y_{j'}$ is empty one as they belong to partitions α and β correspondingly. As a result we have

$$\forall(i,\ j), (i', j') \in \{1,2,\ldots,n\} \times \{1,2,\ldots,m\} \Rightarrow (X_i \cap Y_j) \cap (X_{i'} \cap Y_{j'}) = \varnothing. \qquad (16)$$

The validity of (15), (16) suggests correctness of the inclusion $\alpha \cap \beta \in \Pi_\Omega$, which was required. Now applying first equality from (10) to the sets Z_k and $(X_i \cap Y_j)$ respectively we obtain

$$\mu(Z_k) = \sum_{i=1}^{n} \sum_{j=1}^{m} [Z_k \cap (X_i \cap Y_j)] = \sum_{i=1}^{n} \sum_{j=1}^{m} \nu_{ijk},$$

i.e. equality (11). Absolutely analogously using by pairs sets $X_i \cap Y_j$, $X_i \cap Y_k$, $Y_j \cap Z_{kj}$ and γ, β, α pro tanto we get equalities (12)–(14), q.e.d.

Now we are ready to prove that $\rho(\alpha, \beta)$ is the distance function.

Theorem. Let Ω be arbitrary measurable set with given measure $\mu(\circ)$ and let Π_Ω be a set of its partitions then for arbitrary $\alpha, \beta \in \Pi_\Omega$ functional (4) is a metric.

Proof. To prove the theorem it is sufficiently to show that (4) is nonnegative and satisfies axioms of reflexivity, symmetry and the triangle inequality.

Note, the nonnegativity and the symmetry directly follow from the definition.

To prove reflexivity it is necessary to verify that $\rho(\alpha, \beta) = 0 \Leftrightarrow \alpha = \beta$. First we consider $\rho(\alpha, \alpha)$ for arbitrary $\alpha \in \Pi_\Omega$. In accordance with the symmetry property we have

$$\rho(\alpha, \alpha) = \sum_{i=1}^{n} \sum_{j=1}^{n} \mu(X_i \Delta X_j) \mu(X_i \cap X_j) = \sum_{i=1}^{n} \mu(X_i \Delta X_i) \mu(X_i \cap X_i) + 2 \sum_{\substack{i,\ j=1 \\ i > j}}^{n} \mu(X_i \Delta X_j) \mu(X_i \cap X_j).$$

First term from the right consists of n summands with the identical partition elements and so it is equal to zero as $\mu(X_i \Delta X_i) = 0$ for all $i = \overline{1,n}$. The second one consists of $(n^2 - n)$ different partition elements only and it also equals to zero since for all $i, j = \overline{1,n}, i \neq j$ we obtain $\mu(X_i \cap X_j) = 0$, which establishes direct reflexivity.

Now let us show the validity of reflexivity in a reverse order. Let $\rho(\alpha, \beta) = 0$ for arbitrary $\alpha, \beta \in \Pi_\Omega$ such that $\alpha \neq \beta$. By virtue of terms nonnegativity in (4) we have equality of each summand to zero. Choose an element X' in the partition α. Note, in (4) it belongs to zero summands of kind $\mu(X' \Delta X_j) \mu(X' \cap X_j) = 0$ where $j = \overline{1,m}$. Suppose X' does not belong to β hence the inequality $\mu(X' \Delta Y_j) \neq 0$ is fulfilled for all $Y_j \in \beta$. Then the equality $\mu(X' \cap Y_j) = 0$ holds true for all indexes j. Though it is possible while as $X' = \varnothing$ since $X' \subset \Omega$ and family $\beta = \{Y_1, Y_2, \ldots, Y_m\}$ covers the set Ω. But X' is the element of covering α of the same set Ω and naturally $X' \neq \varnothing$ then there exist elements $Y_1^*, Y_2^*, \ldots, Y_p^* \in \beta$ which cover subset $X' \subset \Omega$ and have nonempty intersection with it, i.e. $\mu(X' \cap Y_r^*) \neq 0, r = \overline{1, p}$. We get a contradiction, i.e. one can assert that any element X' from α is an element of β, i.e. $\alpha \subset \beta$. Due to symmetry we have $\beta \subset \alpha$ and finally $\alpha = \beta$ which proved reflexivity.

Now we are going to examine whether (4) satisfies the triangle inequality.

Let us consider three arbitrary partitions α, β, γ of the set Ω. We have to prove $\rho(\alpha, \beta) \leq \rho(\alpha, \gamma) + \rho(\gamma, \beta)$. Suppose $\alpha = \{X_1, X_2, \ldots, X_n\}$, $\beta = \{Y_1, Y_2, \ldots, Y_m\}$, $\gamma = \{Z_1, Z_2, \ldots, Z_l\}$. Using lemma 1 (expression (5)) we have

$$\rho(\alpha,\gamma)+\rho(\gamma,\beta)-\rho(\alpha,\beta)=$$
$$=\sum_{i=1}^{n}[\mu(X_i)]^2+\sum_{j=1}^{m}[\mu(Y_j)]^2+2\sum_{k=1}^{l}[\mu(Z_k)]^2-$$
$$-2\sum_{i=1}^{n}\sum_{k=1}^{l}[\mu(X_i\cap Z_k)]^2-2\sum_{k=1}^{l}\sum_{j=1}^{m}[\mu(Z_k\cap Y_j)]^2-$$
$$-\sum_{i=1}^{n}[\mu(X_i)]^2-\sum_{j=1}^{m}[\mu(Y_j)]^2+2\sum_{i=1}^{n}\sum_{j=1}^{m}[\mu(X_i\cap Y_j)]^2\geq 0.$$

Collecting like terms and dividing by 2 we arrive at the equivalent expression

$$\sum_{k=1}^{l}[\mu(Z_k)]^2+\sum_{i=1}^{n}\sum_{j=1}^{m}[\mu(X_i\cap Y_j)]^2\geq\sum_{i=1}^{n}\sum_{k=1}^{l}[\mu(X_i\cap Z_k)]^2+\sum_{k=1}^{l}\sum_{j=1}^{m}[\mu(Z_k\cap Y_j)]^2\ .$$

Further, if we apply lemma 2 (expressions (11)–(14)) this inequality can be rewritten as follows

$$\sum_{k=1}^{l}(\sum_{i=1}^{n}\sum_{j=1}^{m}\nu_{ijk})^2+\sum_{i=1}^{n}\sum_{j=1}^{m}(\sum_{k=1}^{l}\nu_{ijk})^2\geq\sum_{i=1}^{n}\sum_{k=1}^{l}(\sum_{j=1}^{m}\nu_{ijk})^2+\sum_{j=1}^{m}\sum_{k=1}^{l}(\sum_{i=1}^{n}\nu_{ijk})^2. \quad (17)$$

There is no difficulty in understanding that at raising to the second power all summands ν_{ijk}^2 in both sides of this inequality are canceled. Furthermore, all doubled products in the right part are contained in the first summand at the left of (17). Indeed, with the notation $M=\{1,2,\ldots,n\}\times\{1,2,\ldots,m\}$, the first item has the form

$$\sum_{k=1}^{l}(\sum_{i=1}^{n}\sum_{j=1}^{m}\nu_{ijk})^2=\sum_{k=1}^{l}\sum_{i=1}^{n}\sum_{j=1}^{m}\nu_{ijk}^2+2\sum_{\substack{(i,j),(i',j')\in M\\(i,j)\neq(i',j')}}\nu_{ijk}\nu_{i'j'k}\ . \quad (18)$$

The first summand at the right of (17) one can rewrite as follows

$$\sum_{i=1}^{n}\sum_{k=1}^{l}(\sum_{j=1}^{m}\nu_{ijk})^2=\sum_{i=1}^{n}\sum_{k=1}^{l}\sum_{j=1}^{m}\nu_{ijk}^2+2\sum_{\substack{(j,j')\in M\\j\neq j'}}\nu_{ijk}\nu_{ij'k}\ . \quad (19)$$

It is clear that all pairs $\nu_{ijk}\nu_{ij'k}$ lie among doubled products $\nu_{ijk}\nu_{i'j'k}$ since in (19) all variations are connected with alteration of the second index whereas both the first and the second indices are varied in (18). The same property is fulfilled for the second summand in the right part of (17) as all variations are connected with alteration of the first index i. As a result we get that all items at the right of (17) are canceled and all summands at the left are nonnegative, i.e. inequality (17) holds. Hence the triangle inequality is valid and (4) is a metric, q.e.d.

Thus, we have proved that functional (4) is a metric. We think that obtained results have to be applied to the set coverings that will provide analysis of arbitrary weak clustering when elements can belong to more that one cluster concurrently.

Results and Conclusion

An intensive experimental exploration with the collection of histologic specimens images with final goal classification as an aid in cancer detection vindicates the efficiency of proposed metric.

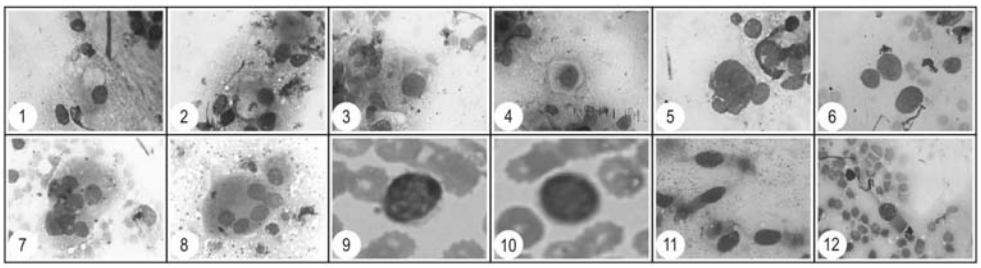


Fig. 2. Examples of input images

The analysis of experimental results has shown that the application of partitions as features provides a sufficient relevance at retrieval of the images in database with queries 'ad exemplum'. Figures 2 and 3 illustrate images

and partitions that were compared by means of traditional and proposed metrics. Examples of dependences, query image and its partition are shown in fig. 4. One can see comparability of obtained results for Euclidean metric and distance function (4). The reliability of matching can be increased by an intellectual processing (via relations analysis of region-based models) which provides conditions for entirely correct and complete segmentation.

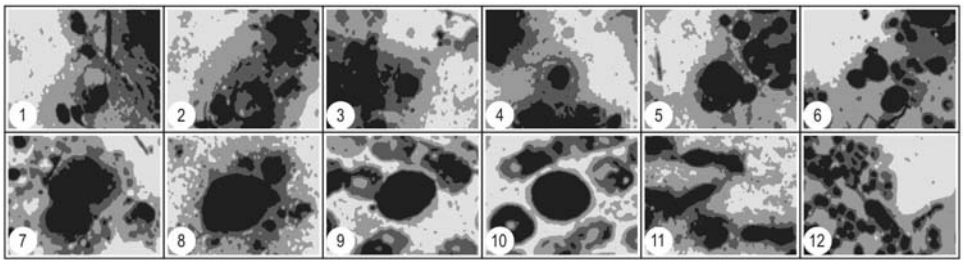


Fig. 3. Multithresholding segmentation of images shown in fig. 1

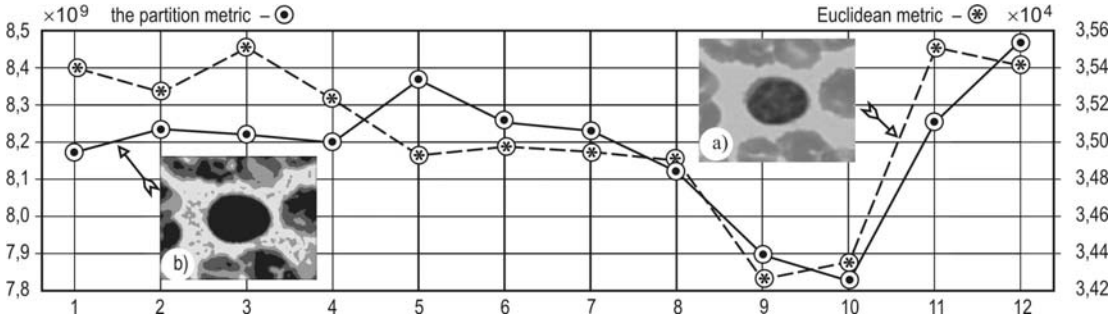


Fig. 4. Examples of image and partitions matches (query a) and b) correspondingly)

Bibliography

1. Müller H., Müller W., Squire, D.McG., Marchand-Maillet S., Pun T. Performance Evaluation in Content-Based Image Retrieval: Overview and Proposals. Pattern Recognition Letters, Vol. 22, 2001, pp. 593-601.
2. Li J., Wang J.Z. Automatic linguistic indexing of pictures by a statistical modeling approach. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 25, 2003, pp. 1075-1088.
3. Käster T., Wendt V., Sagerer G. Comparing Clustering Methods for Database Categorization in Image Retrieval. Lecture Notes in Computer Science, Springer-Verlag, Berlin Heidelberg, Vol. 2781, 2003, pp. 228-235.
4. Mashtalir V.P., Yakovlev S.V. Point-Set Methods of Clusterization of Standard Information. Cybernetics and Systems Analysis, Kluwer Academic Publishers, Vol. 37, 2001, pp. 295-307.
5. Kinoshenko D., Mashtalir V., Yegorova E. Clustering Method for Fast Content-Based Image Retrieval. Computer Vision and Graphics, K. Wojciechowski et al., eds., Computational Imaging and Vision, Springer, Vol. 32, 2006, pp. 946–952.
6. Kinoshenko D., Mashtalir V., Yegorova E., Vinarsky V. Hierarchical Partitions for Content Image Retrieval from Large-Scale Database. Machine Learning and Data Mining in Pattern Recognition, P. Perner., A. Imlya, eds., Lecture Notes in Artificial Intelligence, Springer-Verlag, Vol. 3587, 2005, pp. 445-455.

Authors' Information

Kinoshenko Dmitry – Ph.D. student of Computer Science Department, Kharkiv National University of Radio Electronics, Lenin Ave., 14, Kharkiv, Ukraine, 61166, e-mail: kinoshenko@kture.kharkov.ua

Mashtalir Vladimir – Doctor of Technical Sciences, Professor of Computer Science Department and Dean of Computer Science Faculty, Kharkiv National University of Radio Electronics, Lenin Ave., 14, Kharkiv, Ukraine, 61166, e-mail: mashtalir@kture.kharkov.ua

Shlyakhov Vladislav – Candidate of Technical Sciences, Senior Researcher of Computer Science Department, Kharkiv National University of Radio Electronics, Lenin Ave., 14, Kharkiv, Ukraine 61166.

STRUCTURAL ANALYSIS OF CONTOURS AS THE SEQUENCES OF THE DIGITAL STRAIGHT SEGMENTS AND OF THE DIGITAL CURVE ARCS

Vladimir Kalmykov

Abstract: Recognition of the object contours in the image as sequences of digital straight segments and/or digital curve arcs is considered in this article. The definitions of digital straight segments and of digital curve arcs are proposed. The methods and programs to recognize the object contours are represented. The algorithm to recognize the digital straight segments is formulated in terms of the growing pyramidal networks taking into account the conceptual model of memory and identification (Rabinovich [4]).

Keywords: contour, segments of digital lines arc of digital curves

Introduction

The structural analysis of the object contours in the image as the sequences of straight segments and of curve arcs is one of the tasks of image processing in the artificial intelligence systems.

In most cases the **IMAGE** can be **CONSIDERED** as the **PART** of a **PLANE**, where the objects are located. The objects optical parameters – optical density, color, texture – are different from the similar parameters of rest of the **IMAGE** – background. A **BORDER**, i.e. contour, **IS** inalienable **PROPERTY** of every object and is a simply connected sequence consisting of segments of lines and arcs of curves. The **IMAGE**, as a rule, is discrete. Accordingly, the straight segments and curve arcs, the contours consist of, **ARE** the *digital straight segments* and the *digital curve arcs*.

Automatic segmentation of arbitrary contour on the digital straight segments and/or the digital curve arcs is the purpose of the work.

There are algorithms to select the *digital straight segments* in a contour [1,2]. The curvilinear image elements, represented as splines, Bezie curves, etc., are used in a great number of applications. At the same time the arcs of arbitrary curves, as elements of the description of contours, practically is not used for recognition of image contours mainly by reason of the absence of common determination of the arc of arbitrary digital curve. The use of curve arcs as the structural elements of image contours description would approach its description to intuitional, natural representation of images by a man, substantially would shorten the expenses of memory for storage of image and image processing time. As an example we will consider the description of contours of binary images, which are obtained, with the use of tools of widespread graphics editor Corel Draw.

On fig.1 the contours of three identical objects are represented, which are not to be distorted by noises. Each of the objects contains the arc of ellipse and differs from the other objects by spatial position and rotation angle.

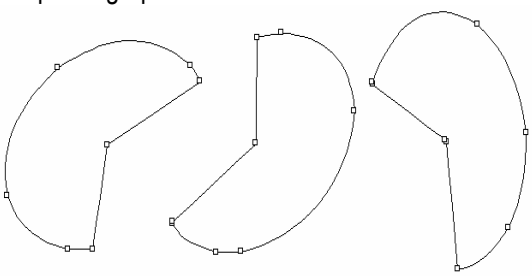


Fig.1 Selection of arcs of ellipses in three identical, rotated in relation to each other objects by means of Corel Draw

The boundary points which divide contours into the curve arcs and the straight segments are marked by the squares. Identical arcs, which belong to different objects, are represented by sequences containing the different amount of different arcs of curves. Each of the identical objects in the image is represented with the different elements. Such description of objects can not be directly used in intelligence systems for interpretation of images supposes hard image processing enough.

The represented example shows existence of the problem with the images not distorted by noises and the actuality to solve it.

Contour as the sequence of L-elements

When the image contours to be digitized the straight segments and the curve arcs transform to digital straight segments and digital curve arcs. We will consider the discrete image, as two-dimensional cell complex [1]. The pixels are two-dimensional square elements for the given case. Besides pixels *there* are *cracs* and points. Cracs are the sides of pixels, being one-dimensional elements. Points are the end points of *cracs* and angular points of pixels. Then the image object contour is the connected *crac* sequence, to be a *boundary* between the object and background pixels. The characteristic features of straight segments and curve arcs to a great extent are lost as a result of digitization.

In fig. 2 the example of the objects initial contour, formed by a curve arc and a straight segment, and also its digital equivalent, as the *crac* sequence, are represented. Some connected parts of the *crac* sequence can be united in *L-elements*.

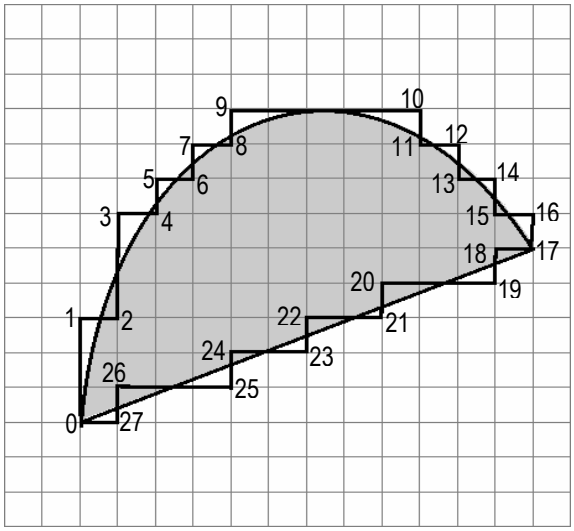


Fig.2 The example of the contour digitization

An L-element is the connected crac sequence of vertical (horizontal) orientation, which contains no more than one crac of horizontal (vertical) orientation.

As in the work [2], we will use such L-element configuration in which the *crac* (if it is present) different in the orientation relatively to the other cracs of the L-element is contained at the end of the L-element. Each L-element is defined with such parameters: g – direction in relation to initial its point *of* (using g = 0 – for direction upward, 1 – to the right, 2 – downward, 3 – to the left); l – amount of cracs of the same direction (l = 1,2,.); q – the last crac direction in relation to previous cracs value g (q = -1, if the last crac is directed to the left in relation to direction *of g*, +1 - to the right, 0 - coincides with direction *of g*). For the L-element (0-2) g=0, l=3, q=+1. For the L-element (27-0) g=3, l=1, q=0.

Determination of segments of digital lines in the sequence of L-elements of contour

Lets (x_1,y_1), (x_2,y_2) are integer-valued co-ordinates of straight segment first and last points. The segment slope ratio is defined by the relation of differences of its co-ordinates $n=\Delta x=|x_1 - x_2|$ and $m=\Delta y=|y_1 - y_2|$, which, in the general case, is not an integer. We will put for definiteness, that $n > m$. The digital straight segment of arbitrary slope ratio can be set by means of two types of L-elements, having identical directions, their lengths are equal accordingly l,l+1, thus $l \le n/m \le l$+1. The order of L-elements alternation determines the digital straight segment structure and is defined by the values of members of continued fraction [l; k_1, k_2,., k_t] or

$$\frac{n}{m} = l + \frac{r}{m} = l + \frac{1}{k_1} + \frac{r_1}{r} = \ldots = l + \frac{1}{k_1} + \frac{1}{k_2} + \frac{r_2}{r_1} = \ldots = l + \frac{1}{k_1} + \frac{1}{k_2} + \frac{1}{k_3} + \ldots + \frac{1}{k_t}. \qquad (1)$$

As follows from the formula (1), l is the integer part from the division *of n* by m - corresponds l cracs of the same direction in succession in the digital straight segment. Together with the joining perpendicular crac they form the L-element of l length.

*K1-element is the sequence in succession of k_1 L-elements having identical with each other parameters, but the last L-element having the length equal **l**+1 or **l**, according to n,m values; k_1 defines the "length" of the K1-element.*

Likely, K2-element is the sequence in succession of k_2 L-elements having identical parameters with each other, but the last K1-element having the length equal k_1+1 or k_1 or according to n,m values; k_2 defines the "length" of the K2-element; et cetera to the exhausting of members of continued fraction. The numerator r determines the

amount of L-elements of length l+1 in this digital straight segment, and also the K1-element amount. In turn numerator r_1 determines the amount K2- element lengths k_1±1 in this digital straight segment. In general numerator r_{t-1} determines the amount K_{t-1}- element lengths k_{t-1}±1 in this digital straight segment.

Under the digital straight segment with the co-ordinates of first and end points (x_1,y_1), (x_2,y_2) we will understand the sequence of L-elements having identical directions g,q, where integer-valued lengths are equal according to $\boldsymbol{l}$, $\boldsymbol{l}$*+1, and* $\boldsymbol{l \le n/m \le l+1}$*, where* $n = |x_1 - x_2|$ *and* $m = |y_1 - y_2|$*, thus, the order of L-elements alternation is defined by the values of members of continued fraction* n/m.

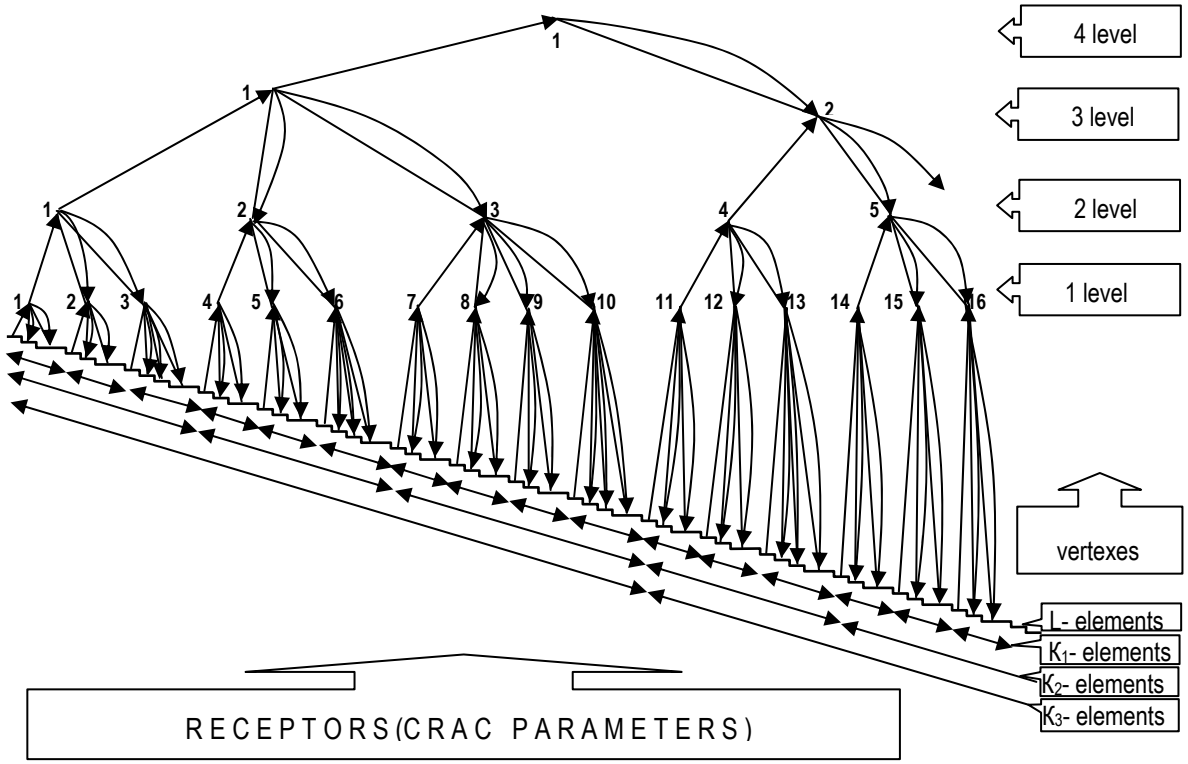


Fig.3 The process of digital straight segment image forming in terms of growing pyramidal networks

The algorithm to select digital straight segments in crac sequence was formulated in the work [3]. It is possible to represent the process of digital straight segment image forming by the mentioned algorithm in terms of growing pyramidal networks [5] (fig.3) taking into account the work by Z. L. Rabinovich [4]. The special case of the growing pyramidal network, when the vertexes subsets have not intersections, is examined. The crac parameters, which form L-elements, are used as receptors. The vertexes of the network are tied-up with conceptors and feed-backs. Conceptors are represented by ascending direct pointers and lines without pointers. Feed-backs are represented by descending pointers. Successive procedure to form the digital straight segment image is examined - from left to right. The algorithm to form digital straight segment image in the arbitrary crac sequence consists in the following.

Starting conditions. The L-elements counter s_0=1; the common amount of L-elements in the sequence is equal N; the vertexes counters of t level: s_1= 1, s_2= 1,., s_t= 1; t = 1; the work constant p = 0.

The list XY of the co-ordinates of digital straight segment ends is empty.

1. The L-element number s_0 parameters is defined and is passed to the vertex number s_1. To go to the following L-element s_0:= s_0 +1; Jump to point 2. If s_0>N, to store the value s_1 and to set s_1 at the starting state*:* $p = s_1$; s_1=1; Jump to point 3.

2. Using the feed-back arc, the L-element parameters of vertex number s_1 is compared to the parameters of the following in order L-element number s_0. If the result of comparison is positive (the examined L-elements form K1-

element) Jump to point 1. Otherwise the K1- element construction is completed. If the built K1- element is completed, go to the next vertex of the first level s_1:= s_1 +1; Jump to point 1. Otherwise, the digital straight segment is completed, the vertexes counters are set to the starting state; the co-ordinates of the digital straight segment ends are stored; Jump to point 1.

3. Parameters of the Kt-element are passed from the vertex of *t* level number s_t to the vertex of *t*+1 level number s_{t+1} using a conceptor, go to the new vertex of *t* level s_t:= s_t +1; Jump to point 4. If s_t > p, the value of s_t is stored and s_t is set to the starting state: p:= s_t; s_t:=1; go to the following level t:=t+1. If amount of vertexes of the next level s_t>1, Jump to point 3, otherwise, go to End of algorithm.

4. Using the feed-back arc the Kt-element parameters of vertex number s_{t+1} are compared to the Kt-element parameters of the vertex number s_t. If the result of comparison is positive (the examined Kt-elements form Kt+1-element) Jump to point 3. Otherwise, the K1- element construction is completed. If the built Kt+1- element is completed, go to the next vertex of the *t* level: s_t+1:= s_{t+1} +1; Jump to point 3. Otherwise, the digital straight segment is completed, the vertexes counters are set to the starting state; the co-ordinates of the digital straight segment ends are stored the list *XY*; Jump to point 1.

5. The End of algorithm: forming the list of the co-ordinates of the digital straight segment ends.

The offered algorithm allows restoring of the objects contours of digitized image from the crac sequence to the digital straight segment sequence. However the contours of the initial image consisted not only of straight segments but also of curve arcs. In the next sections of the article methods and algorithms to restore the digital curve arcs of the digital straight segment will be developed. So, the object contours of digitized image will be represented not only as the *sequence of digital straight segments* but the *sequence of digital straight segments and/or of digital curve arcs*.

Determination of arc of digital curve

The determination of digital curve arc will be formulated in this section. This determination allows setting or rejecting a fact: some part of contour sequence, which consists of digital straight segments, is formed as a result of digitizing of some arbitrary curve arc.

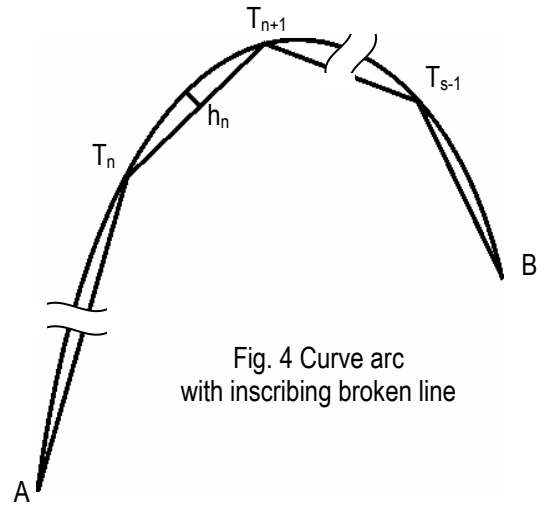


Fig. 4 Curve arc with inscribing broken line

We will suppose that arcs of curves using in the graphic images, display the segments of continuous functions with continuous derivative ones.

Under the continuous curve lines [6], prescribed functions $x = \varphi(t)$, $y = \phi(t)$, we will understand the Zhordan curves without multiple points or simple arcs, in other words such, that for any two different values t' and t'' points proper to them on a plane M′ [$\varphi(t')$, $\phi(t')$] and M'' [$\varphi(t'')$, $\phi(t'')$] are different. Lets $x = \varphi(t)$, $y = \phi(t)$, where $\varphi(t)$, $\phi(t)$ are the continuous functions of parameter *t*, defined on the segment [*a*, *b*]. If *t* is increasing from *a* to *b,* the point with the co-ordinates of *x*, *y* describes the arc *AB* (fig.4). We will consider breaking up of segment [*a*, *b*] with the points of division

$$a = t_0 < t_1 < \ldots < t_{s-1} < t_s = b, \qquad (2)$$

and let curve points *A*, T_1, …, T_{s-1}, *B* correspond to these points of division. Let us connect consistently by the straight segments the point *A* with the point T_1, the point T_1 with the point T_2, ., the point T_{s-1}, with the point *B*, so we will build the broken line, and will name this broken line as entered in the arc *AB*. The arc segment T_n, T_{n+1} (n = 0,1,…,s) is the figure, limited by the segment of broken line T_n, T_{n+1} and the proper arc link $\cap T_n, T_{n+1}$. The maximal length of line between a segment T_n, T_{n+1} and $\cap T_n, T_{n+1}$, which is perpendicular to the segment T_n, T_{n+1} is the height of arc segment h_n. Lets

$$\beta = \max_{n=0,1,\ldots,s-1} l(T_n, T_{n+1}) \qquad (3)$$

if β will tend to zero while s is increased, the length of any link of the entered broken line will tend to zero, as well as the height of every arc segment, for functions $\varphi(t)$, $\phi(t)$ are continuous.

While the curve arc and its entered broken line are mapped onto discrete space, the value of discontinuity is d, the entered broken line segments will be displayed by digital straight segments. The co-ordinate values of discrete space are integer-valued and multiple to d. So since a moment, when $h_n < d$, the segment heights will not be mapped in this space, because their lengths will become equal to zero. While $h_n < d$, discrete mapping of arc links will coincide with the proper links of entered broken line – the digital straight segments. We will name these digital straight segments, representing certain curve arc, as ***digital curve arc***. Thus, the contour, which initially consists of segments of lines and of arbitrary curve arcs, after digitizing is mapped as sequence of digital straight segments.

The task is to determine digital curve arcs in the contour sequence of the digital straight segments.

We will examine the pair of neighboring digital straight segments in a sequence which correspondents to curve arc. The pair of neighboring segments determines the finite difference of the second order. We will name two pairs which have a general segment as neighboring pair. Neighboring pair determines the finite difference of the third order.

If the finite differences of the second order are not equal to zero (for the integer-valued co-ordinates of points finite differences must be more than 1), maybe the pair of digital straight segments is part of a digital curve arc. In general, it is possible to conduct many curves through three points, defined by the pair of neighboring digital straight segments. At the same time, the arc segment heights values h_n does not exceed the space value of discontinuity d ($h_n < d$). Thus, for the pair of digital straight segments $T_{n-1}T_n$, T_nT_{n+1} to be the part of a digital curve arc, it is necessary to prove the existence of the curve which passes through points T_{n-1}, T_n, T_{n+1}, and the condition is executed for: $(h_{n-1}<d)\&(h_n<d)$.

Usually the curvature of a flat curve is equated with curvature of contiguous circumference arc [7]. By the tangent circumference to plane curve in the point $T1$ is named the limit position of the circumference passing through two points T_2 and T_3 neighboring to the point T_1, while T_2 and T_3 tending to T_1. Abandoning aside the decision of this task in general case, further we will examine the partial case of this task, when a circumference is the curve passing through three points. It is possible to formulate the next ***determination*** as result of the consideration.

Under the digital curve arc in two-dimensional discrete space of discontinuity d we will understand such sequence of digital straight segments, that through ending three points of every pair of neighboring segments it is possible to conduct such circumference, that its segment heights does not exceed d.

This determination is correct as far as it is correct to equate the segment of arbitrary curve arc, which corresponds to the pair of neighboring segments, with the arc of tangent circumference.

The offered determination allows to define, whether a given pair of neighboring digital straight segments corresponds to some arc of the digital arbitrary curve. Lets build the circumference arc on points $Tn-1, Tn, Tn+1$ in accordance with the determination of digital curve arc, and estimate the deviation size of end of each segment from direction of line of preceding segment (fig.5). As the deviation size it is possible to choose segment length $T_{n+1}T^*$ when condition that $l(T_n'') \cong l(T_n, T_{n+1})$. Define the length T_nR_2 as a height of triangle $T_{n-1}T_n\ T_{n+1}$. Maximal distance is between the points of curve arc and proper segment of broken line SR_1 is equals d. At the same time

$$SR_1 = OT_{n-1} - OT_{n-1} \times \cos\alpha = r - r \times \cos\alpha = r(1 - \cos\alpha).$$

The length of height $\Delta(T_{n-1}T_n\ T_{n+1})$

$$T_nR_2 = OT_n - OT_n \times \cos 2\alpha = r - r \times \cos\alpha = r \times (1 - \cos 2a) = 2r \times (1 - \cos 2\alpha).$$

$$T_nR_2 / SR_1 = 2 \times (1 + \cos\alpha); \text{ or } T_nR_2 = 2 \times (1 + \cos\alpha) \times SR_1.$$

If $SR_1 \approx d$ and $\alpha \leq 30°$, the height of triangle $(T_{n-1}T_n\ T_{n+1})\ T_{n2} \leq 3{,}85d$. It is not hard to see that the maximal deviation size is

$$Tn+1T' = 2' \ TnR2 \approx 7.7d. \qquad (4)$$

It means that the considered pair of segments could be corresponded to the digital curve arc, if the maximal deviation value T_nT^* is not more than 7.7d. Minimum deviation value must be more than d: $T_{n+1}T^* > d$, because if it is less one of the pair of segments grows into one digital straight segment.

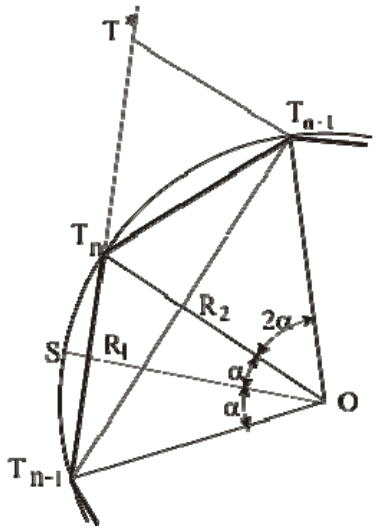


Fig. 5 Estimation of the deviation size of neighbouring segments ends

The example is demonstrated on a fig. 6 the program to recognize the object contours in the image as the sequences of digital straight segments and of digital curve arcs. The image is used from the fig. 1, but the offered algorithms are implemented in the program. Unlike the contours of fig. 1, got by means of the program of Corel Trace, the arcs of contours are represented without lying out of intermediate points on a few arcs regardless of different spatial positions and rotation angles for the each of objects.

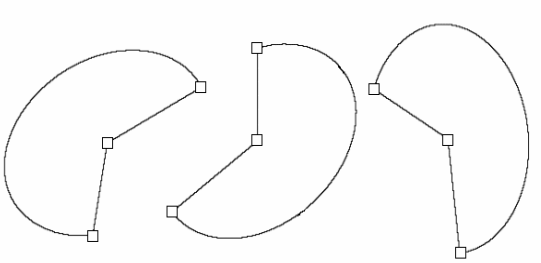


Fig. 6 Object segmentation using describing methods

Conclusion

The automatic construction of reflex of digital straight segment is considered, as the special case of growing pyramidal network taking into account the conceptual model of identification and memory.

Determination of digital curve arc is offered as the sequences of digital straight segments. The algorithms and programs developed on the basis of the attained results allow, unlike known, to execute segmentation of contours in a natural way on the digital straight segment and arcs of digital curves regardless of different spatial positions and rotation angles for the each of objects in the image.

The binary images contours processing had been developed without the loss of information. The questions of the half-tone image segmentation by the contours consisting of digital straight segment and of arcs of digital curves are the subject of the next works.

Bibliography

[1] Kovalevsky V.A. Applications of Digital Straight Segments to Economical Image Encoding, In Proceedings of the 7th International Workshop, DGCI'97, Montpellier, France, December 3-5, 1997, Springer 1997, P. 51-62.

[2] Вишневский В.В., Калмыков В.Г. Структурный анализ цифровых контуров изображений как последовательностей отрезков прямых и дуг кривых. // Искусственный интеллект 4'2004, Институт проблем искусственного интеллекта НАНУ – Донецк: "Наука и образование" 2004 С.450-457

[3] Власова Т.М., Калмыков В.Г. Алгоритм и программа распознавания контуров изображений как последовательности отрезков цифровых прямых // Математичні машини і системи. – 2005. -№4 С.84-95

[4] Рабинович З.Л. О естественных механизмах мышления и интеллектуальных ЭВМ // Кибернетика и системный анализ. – 2003. -№5 С.82-88

[5] Гладун В.П. Планирование решений. – Киев: Наукова думка, 1987. – 167с.

[6] Макаров И.П. Дополнительные главы математического анализа. – Москва: Просвещение, 1968.

[7] Корн Г. и Корн Т. Справочник по математике для научных работников и инженеров. – Москва: Наука, 1974

Author's Information

Vladimir Kalmykov – senior researcher, candidate of engineering sciences, Institute of problems of mathematical machines and systems, prosp. akad. Glushkova 42, 03680, Kiev 187, Ukraine; e-mail: kvg@immsp.kiev.ua

COMPRESSION TECHNOLOGIES OF GRAPHIC INFORMATION

Nikolay Fesenko

Abstract: The classification of types of information redundancy in symbolic and graphical forms representation of information is done. The general classification of compression technologies for graphical information is presented as well. The principles of design, tasks and variants for realizations of semantic compression technology of graphical information are suggested.

Keyword: semantics, abundance, compression, graphical information, decompression, classification, ontology.

ACM Classification Keywords: A.1. Introductory and Survey; I.2. Artificial Intelligence; I.2.7 Natural Language Processing - Text analysis;

Introduction

Presentation of graphic files in the compact compressed kind is needed for comfort of storage, provision of reception and information transfer on the ducting of connection. Thus redundancy in presentation of information allows this information to squeeze, i.e. to shorten the resources expended on its presentation. When diminishing the sizes of memory, occupied by files, or at preparation of files for sending, the specialized programs are used in a compact form, accepted to talk that files are exposed to compression, or the compression [1]. Technologies of compression of files use, as a rule, programs diminishing file sizes graphic arts due to the change of method of data organization, for example, replacing repetitive elements by information that is more effective for storage. Thus graphic information compresses only on a form presentation, but not on maintenance. This appears to be actual development of semantic compression technology for graphic information. To that end we shall explore compression technologies of graphic information, on the article of application in them, approaches and methods of artificial intelligence.

Varieties of Informative Redundancy

Every form or model, different presentation actual to the same thought differs by the informative redundant - from minimum to considerable. It can be where sketch, sketching or the high-quality colored computer image is synthesized. Thus, the more adequate the chosen model, so much the better attainable degree of compression. We will consider *the types of informative redundant*, presentations of information characteristic for different organizational forms, and will execute their classification with the purpose of the determination of the possible hidden reserve for the lead through additional, deeper compression (Fig.1). The basic types of surplus [2] are near to the character and graphic forms of presentation of information. Thus for every type of redundant there are the models of compression. The design in one or another degree reflects possibility of the prediction of probability of the offensive of possible events. *On probability of reiteration*, when different events have different probabilities of appearance, takes place a distributing redundant. If a few identical events can follow friend by a friend, takes place surplus *of reiteration*. Surplus *of order* appears then, when the offensive of certain events can be predicted depending on some order of the following. Reiteration of events group results in a *group* (chainlet) redundant. Possibility for the account of probability of certain events, appearance on some positions in an events stream determines the *position* redundant. Probability of the values of closeness to the neighboring pixels of a graphic image, as well as their spatial derivative, characterizes the spatial redundant *on a form placing*: *linear, 2D, 2.5D, 3D*. *On composition* the *homogeneous* redundant is uncommon; more frequently the redundant is *mixed*. *On the redundant element* it is shown up by the *character* and *objective*.

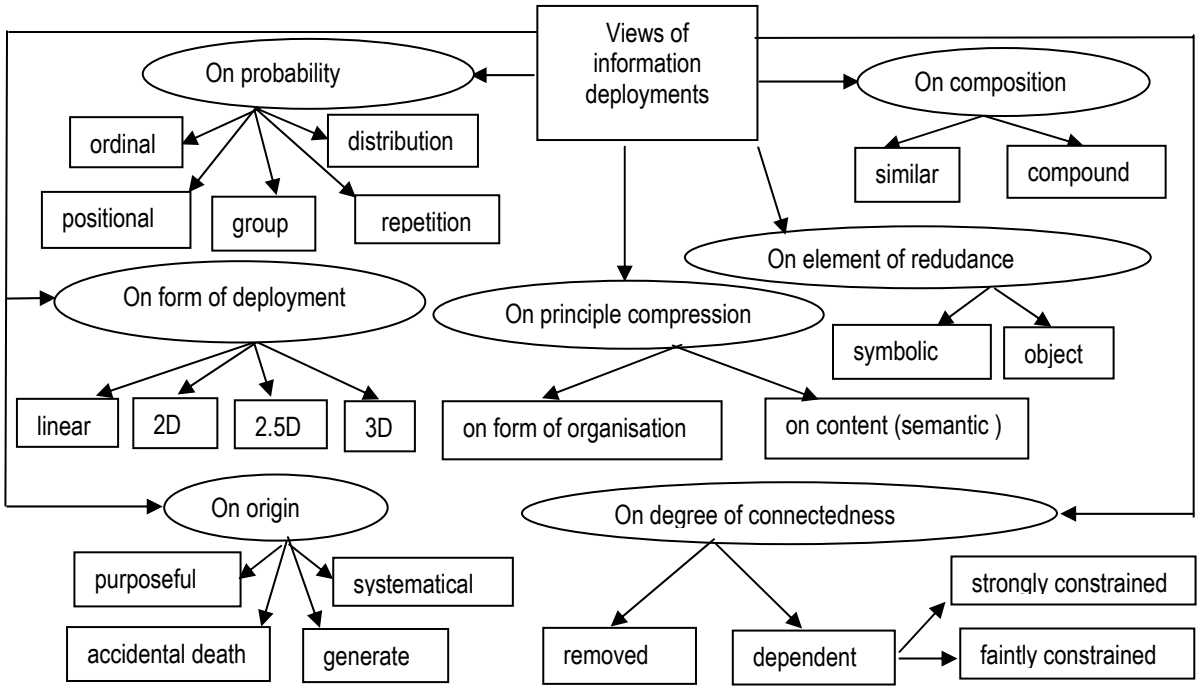


Fig.1 Classification of informative surplus

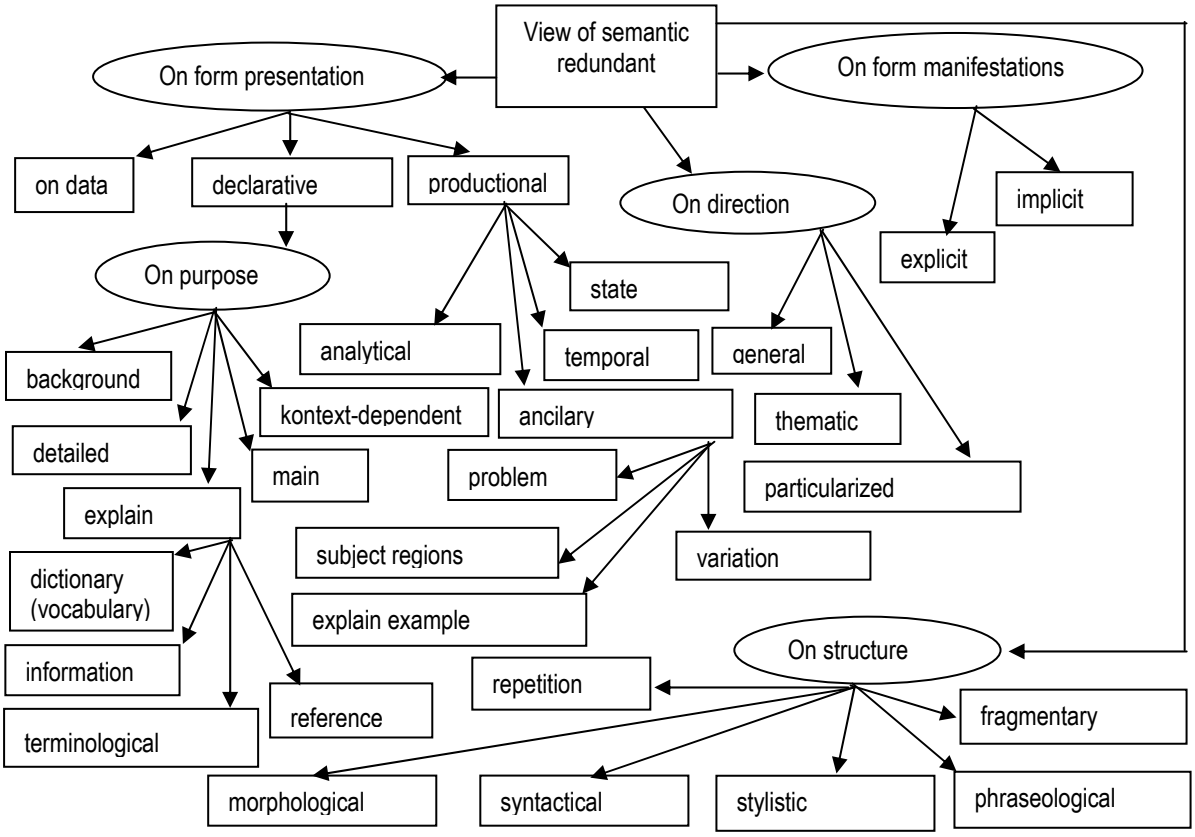


Fig.2 Classification of semantic redundant.

On the principle of compression we will distinguish the redundant *on a form organization* placing, either the locations of character or appearance, and *semantic* or semantic redundant *on maintenance*. *Originally* we will select a *purposeful*, or intentional redundant; *casual*, *originative* and *systematic*. *On the degree of compendency* we will mark the *unrelated*, or the independent redundant, and also the *dependent*, which then can be both *loosely-coupled* and *strongly*. It ensues that from the conducted analysis and classification of redundant, that less of the other semantic aspect of redundant is investigational.

Varieties of Semantic Redundant

We will conduct the analysis of types of semantic redundant and will execute their classification for determination of tasks on creation of technology of semantic compression of graphic information (Fig.2).
Semantic redundant differentiates *on a form presentation*, *on a form the display*, *to direction*, *on a structure* and *in grain settings*. *On a form presentation* we will select the redundant *from data*, *declarative* and productional to the redundant. *In grain settings* we will divide *declarative* surplus into *basic*, *base-line*, *context*, *explaining* and the redundancy *of working out in detail*. Probability of appearance of character in a context we will name *context* redundant. In same queue, *explaining* the redundant shows up as *dictionary*, *information*, *terminology* and *reference*. The productional redundant includes the *analytical*, *temporal* and the redundant *of the states* and *auxiliary*, to which we will deliver the auxiliary containers of modifiers also. *On a form the display* we will select *obvious* redundant and *non-obvious*. *General* character of the redundant, *thematic* and *specialized* shows up *to direction*. The *thematic* redundant is subdivided into *a problem*, *subject domain*, *variation* and redundant *of explaining examples*. *On a structure* we will mark *morphological*, *syntactic*, *stylistic*, *fragmentary*, *phraseological* and the redundant *of reiteration*. The offered classification allows to give mind on the semantic redundant in the context of compression of graphic information, as its programmatic realization is practically absent for today.

Analysis and Classification of Technologies of Compression of Graphic Information

With the purpose of systematization we will conduct the analysis and general classification of technologies of compression of graphic information (Fig.3). *On a form presentation of graphic information* we will distinguish *a raster*, *vectorial*, *programmatic*, *text*, *multimedia* and *combined* forms of presentation. A *raster* form is characterized by a considerable spatial redundant. The type of redundant of *vectorial* form depends on the large number of factors and does not possess sufficient universality for creation of general formal approaches to its removal [2]. Among the formats of vector graphics we will select two: CDR (CORELDRAW) and DXF (AUTOCAD), which are standard for the proper graphic packages. Besides rasters, vectorial, texts formats, also the mixed formats or metafiles are developed. For example, the formats of ESP (Encapsulated PostScript) and CGM (Computer Graphics Metafile) allow the keeping of rasters, vectorial and text information. The universal data files of UDF (Universal Data File) engulf different data structures and can enter in the complement of the documents realized in other formats. Presentation of image stage as programmatic files both in programming languages and instructions languages of graphic packages, allows largely shortening the volume of information about the image, as such approach supposes application of other form of representation specification and assumes the use in the graphic files of programmatic procedures. A compiler, or presence of the proper graphic package, is however here obligatory. For more difficult images the complete cycle of the repeated creation of image is required.

In relation to the file structure of presentation of information analyzable technologies execute the compression at once all file wholly or the compression plugged in the structure of file; and also the combined forms. Formats in which the compression is part of a file structure are less dependent at their further use. A fully compressed file is usually impossible to use until it is not be recovered to the initial state. Therefore, in future, compressed picture-size rasters by the decompression proper compatible program. The formats of compression of ZIP and RAR, which are used for the compression of both character and graphic information, are most widespread.

The Wavelet-compression of graphic information, which we will deliver to technologies **by appearance transformations**, consists in the compression of all images wholly to the so-called WIF-files (Wavelet Image File) with the purpose of maintenance of temporal and frequency descriptions of signals. The basic idea of lifting-transformations urgently and also the wavelet second generation, is expressed in that during every operation must change only one constituent, therefore, for example, for construction of reverse transformation it is enough to execute the mirror reflection of direct transformation and to add inverting of signal wherein it is needed. The algorithm of compression *of EZW* (Embedded Zerotree Wavelet) supposes the simple casting-out of coefficients with the values less some size. Not far the algorithm of SPIHT (Set Partitioning in Hierarchical Trees) more difficult in the calculable relation gives the much best results to the compression.

On plenitude of maintenance of coercible information we will divide compression algorithms into the executing ones of the compression *without the losses*, *with the losses* of information and *with semantic correction of coercible information*. The known technologies of compression *without the losses* practically do not execute semantic interpretation of coercible information, and at decompression aim to recover both a file and picture in full initial. Widespread technologies of compression of graphic information, executing the compression *with the losses* of information, do not take into account semantic maintenance of this information. Consequently, *lost* maybe to be exactly that part of graphic image for the sake of which it, actually, was all saved and was passed. Even half-tones distortions at the compression would carry electoral character, for example, to spread not on the image of face of portrait image, and to affect a background and clothes only.

In-process *algorithms* on the basis of *successive scan-out*, images oriented to the compression *without the losses*, it is possible to select the following phases: prediction, design of error and encoding, thus the prediction of value of next pixel is made on the basis of values of the already coded neighboring pixels. In basis of technologies of compression *on the keywords of KWE* (Key Word Encoding) principle of encoding of lexical units is fixed to, for example ordinary words, by byte groups of fixed length. The result of such encoding is placed in a special dictionary. Technology of Lempel-Ziv-Welsh (*LZW*), which is the modification of technology of Lempel-Ziv (*LZ*), is based on the search maintenance of templates into the set structure. Algorithm *of LZW* is built round the table of phrases (dictionary), which substitutes characters strings of coercible report by the codes of fixed length. The compression on this technology executes the raster format *of TIFF*, which supports plenty of algorithms of compression for the different types of images. In a raster format the *GIF* image is examined as a unidimensional sequence of pixels, which are encoded on lines with the use of algorithm *of LZC* (modification of algorithm *of LZW*). A format is counted on the images with index colors and possesses more high degree of compression, by what format *of PCX*. There are modifications of this format, which differ by the amount of the supported types of blocks of expansions, including animation. On the whole the algorithms *of KWE* execute the compression *without the losses* of information, and are most effective for text information of large volumes, but from the necessity of creation and maintenance of dictionary ineffective for the files of small sizes. In basis *of Huffman algorithm* is also productive the compression *without the losses* of information, the idea of encoding by bits groups lies.

After the lead through of frequency analysis of entrance sequence of information, that determines the arrival rate of every character, which meets in it, characters are assorted on diminishing of this frequency. A basic idea consists of the following: the more frequent the character, the less encoded are bats. The result of encoding is added to the dictionary necessary for decoding. Algorithm *of RLE* (Run-Length Encoding) is based on the exposure of repetitive sequences of information and replacement by their more simple structure the code of information and coefficient of reiteration is specified in which, therefore he gives the best effect of compression at greater length of repetitive sequence of information. Consequently, the algorithm *of RLE* is more effective at the graphic data compression (in particular case for the solid images). Encoding on the algorithm *of RLE* supports the graphic format *of PCX*.

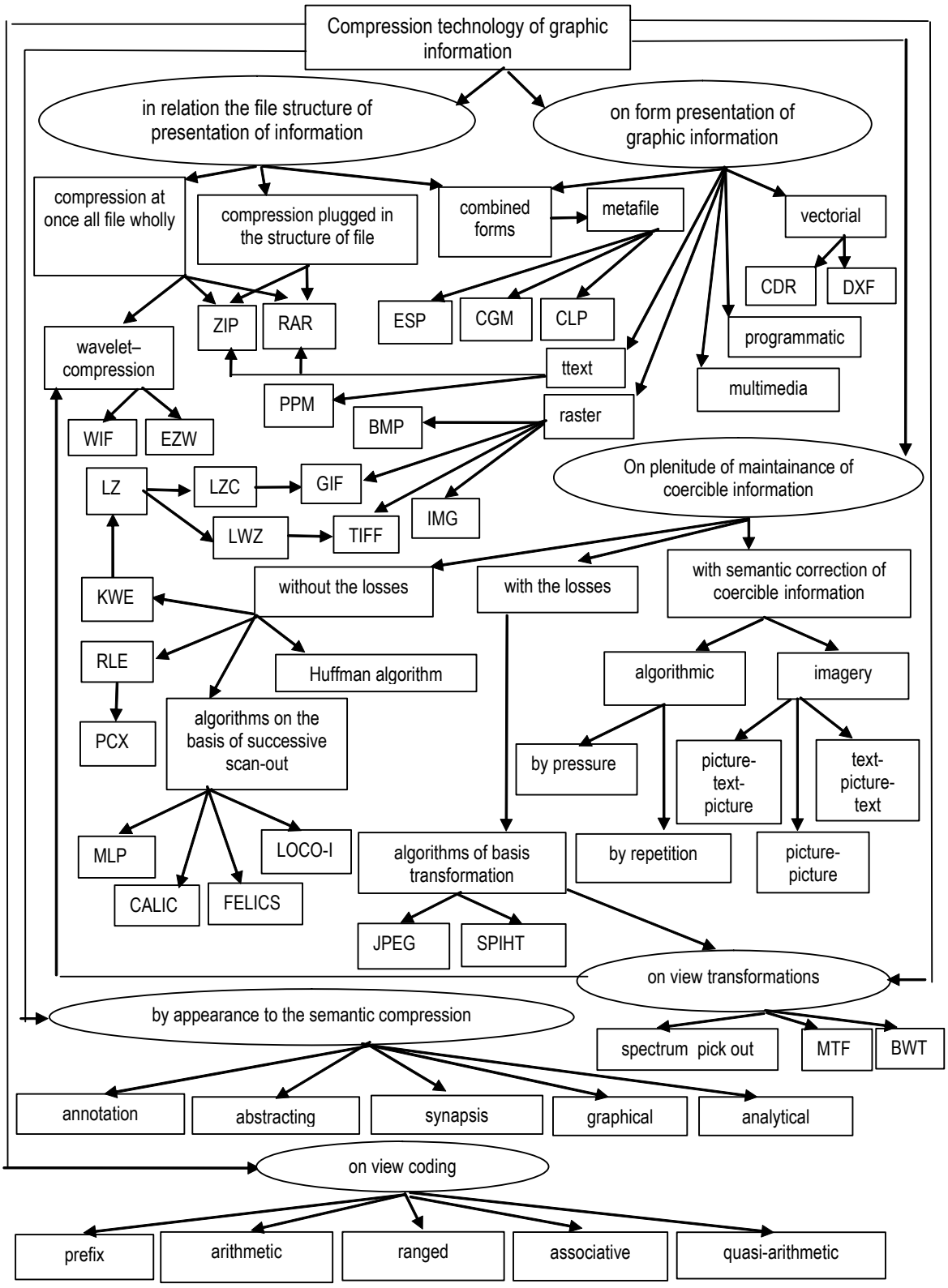


Fig.3 General classification of compression technology of graphic information

Algorithms *by appearance transformations* are intended for the compression of images *with the losses* of quality. The model chart of algorithm is such a transformation, quantisation, design of regenerate coefficients, encoding. *There* are *two approaches* to implementation of transformations. *The first* is expressed in frequent implementation of transformation separating the highest-frequency constituent of image. *The second approach* consists in implementation. Transformation of the result of which there is the spectrum of image. The high calculable complication of such transformations does not allow us to apply them for the fragments of image of largeness. Then at the compression *with the losses*, for example in the format *of JPEG*, there is the casting-out of part of information of the initial image. The features of sensitiveness of our sight are thus used on perception of changes of brightness, colour or tone. At first the image is divided into brightness and two colour constituents. A brightness constituent is major at perception of image by a human eye; it is therefore necessary to squeeze it with the best quality. Then these constituents are broken up on squares measuring 8x8 pixels, and for each of which is executed the discrete cosine-transformation of Fourier. After it the quantisation of results and encoding is made. For the encoding of coefficients *encoding of Huffman* is used with the fixed tables, as variety of the *prefix* encoding, or *arithmetic* encoding. *By appearance* we will select *encoding: prefix*, *arithmetic*, *ranged*, *associative* and *quasi-arithmetic* encoding.

The division *by appearance to the semantic compression* allows to select *annotating*, *abstracting*, *summarizing*, or exposition, as different on the degree of compression and generalization universal forms of retelling of maintenance, and also *graphic*, or vivid, and *analytical* (formalization) forms of presentation of maintenance. Thus two last forms execute the function of compression of graphic information only in the system of some limitations.

Semantic Compression of Graphic Information

A picture, as the most simple from the possible and widespread forms of presentation of information, can be intended to expel the text description, or vice versa. Therefore we will analyze *three charts*: *text-picture-text, picture–text-picture* and *picture-picture*. *The first chart* is expedient for subject domains with the developed system of compression of graphic information. Excelling the algorithms of compression of texts naturally linguistic or programmatic structural descriptions. *The second chart* is applicable for the case of subject domain, in which compression of naturally linguistic texts and programmatic formalities is executed well, than the compression of graphic presentations of informative material. *The third chart* supposes the semantic compression directly linked to graphic images without intermediate auxiliary texts forms. The task of understanding of graphic languages joins to the same chart: pictures, drafts, illustrations, photos and other such *charts* assume a few variants of realization. In the data computer center of Russian academy of science, [3] the system of TEKRIS is built, are generations of pictures on the basis of texts, using a chart: the analysis of text is the synthesis of graphic appearance.

We suggest to make the process of compression of graphic information on the basis of semantic interpretation. Introduction is the successful decision to technology of compression of graphic information of ontological approaches. One of variants of practical realization of such approach is text description of pictures and application to the description of methods of semantic reduction, to which we will deliver summarizing of naturally linguistic texts, in particular, on the basis of growing pyramidal networks.

As the example of realization the summarizing of naturally linguistic texts we will present the program KONSPEKT [4], which passes sense, but does not restore a source code after the conducted compression. During work of this program of phrase of source code is exposed to syntax -semantic interpretation with the purpose of selection of full-composition suggestions. After it on the basis of ontology of associations the thematic analysis of text is carried out. The dictionary of terms, in which terms are indexed by the sets of associative signs designating associations of concepts, is ontology of associations. As a result of work of the program KONSPEKT the generalized text, which on volume and to the compendency near to the folded pictures of properties of

compendia is formed. The use of a dictionary complex is other variant RUSLAN [5] as the instrument of compression of maintenance of text. A function similar on the producible effect of decompression is executed by the standard information-search systems: using the keywords and terminologies of combinations of words, which are the maximum compressed form of presentation of theme, set from outside, they find complete maintenance of text.

A graphic object can be considered certain only then, when its form is fixed for visualization, sizes are filled in and the spatial orientation and location [6] is executed. One of the first programs, which in the interactive mode could perceive the mutual location of blocks of different forms and colors and to make their description, there is the program of Vine of SHRDLU [7], however much its developers did not succeed to decide the task of abstracting. Semantic compression directly linked to graphic image also close to the semantic downscaling. The difference is only in that at the semantic compression brief information goes down or mutates, and at the semantic down-scaling it is temporally eliminated at the increase of scale, but after, at the diminishing of scale, comes back again. At decompression a semantic aspect is also possible, I.e. decompression of image with opening of the represented theme and its addition by information from a knowledge or database base, for example, in the simplest variant, by the list of literature. Rows of algorithms of computer graphics [8,9] use the generating of families of segments of lines, circumferences, ellipses, parabolas and hyperbolae, and assume application of compression and reiteration. Winning from application of method of reiterations the more than the longer generated segment and than anymore coefficient of reiteration. For every quadrant sequences, which the points most close located to the designed curve of line and a brightness is properly distributed between the basic point chosen algorithmically and auxiliary are calculated on, concern taking into account asymptotic branches and symmetry, the choice of which depends on distance relatively by a curve. Such algorithms are applicable at programs development of semantic decompression.

Basic Principles of Construction of Technology of Semantic Compression of Graphic Information

In the general context of development of existent compression technologies we will formulate basic principles of construction and task on the creation of technology of semantic compression of graphic information. *By the main principle* which determines the offered approach to the problem *of compression of graphic information on its sense*, *there* is *extraction of knowledge from the information represented in a graphic form*. General strategy of achievement of this purpose must be built on principles, which are set forth below.

The first principle is *presentation of stage of image by drawing primitives*. The tasks of analysis and synthesis of stages of graphic images are based on that such stages preliminary are described through the dictionary of drawing primitives: point, segment, line, triangle, rectangle, square, rhombus, polygon, arc, circumference, circle, ellipse, hyperbola, parabola. From them it is possible to make many other graphic objects, as lines and circumferences with the use of some receptions of interface develop the methods of quasi-general presentation of graphic objects. Basic spatial graphic objects usually consider parallelepipeds, pyramids, cylinders, and cones of rotation, sphere and torus. In the case of simple graphic object the task of his presentation consists of association of these bodies by addition or deduction and in implementation of operations of descriptive geometry, transfers, turns, sections, impositions. However, in most cases by bodies urgent analytical (i.e. in fact drawing primitives) it is impossible to describe the real mechanical objects. For this purpose it is possible to use the existent methods of determination of form of objects, based on successive adjustment.

Second principle is *formalization of presentation of stage of image* on the basis of division of stage of image on the separate graphic objects baselined in composition.

The third principle is recognition of graphic objects. Graphic object it is possible both to recognize at once and to decide this task for a few steps, and also to copy and pick up an alike object with the known analytical description. For this purpose it is possible to apply sewing together surfaces. In simplest case the circumflex

spheres of permanent radius, which provide the smooth change of tangent in transition from one sewn together surface to other, are such surfaces. The decision of tasks of recognition or distinction of objects on their images behaves functionally to machine sight. It is necessary to notice that for the surfaces of difficult form, such as: surfaces of bicurvature, surfaces of variable curvature, distorted surfaces the decision of task is begun with implementation of sequence of steps of approximation. A task consists of the mathematically exact reproducing of form of graphic object, coming the co-ordinates of the points located on-the-spot it's from. Measuring of inclinations of tangent also it can be used for this purpose, however much it does not provide achievement of exactness, and comparable with exactness of the methods based on measuring of linear co-ordinates. Thus classification of superficial forms of graphic objects comes forward to one of intermediate tasks.

Fourth principle is ***drafting of ontology of graphic associations***. The mechanism of semantic compression must be stopped up in the system carrying out the automatic understanding of text representation specification. Such functions must be also taken into account at planning of in-use of dictionary of grapheme. If to unite the dictionary of graphic terms and primitives with the dictionaries of more commons concepts characterizing thematic subject domains, it is possible to make ontology of graphic associations for some their subset.

Fifth principle - ***the semantic compression can be conducted in the interactive mode with participation an expert.***

Sixth principle is ***translation of stage of graphic image in the intermediate text form of presentation.***

Seventh is ***semantic compression of text form of presentation of graphic image***. The lead through of preliminary translation of graphic image in text description and application to such form of modification of the known technologies of semantic compression of character information is not uniquely the only possible decision, because to text description it is possible to additionally apply technologies of character compression of form of presentation, for example, *PPM* or *combined* variants.

Eighth principle is ***renewal of image from the intermediate text form of presentation in graphic one.*** After the lead through of semantic reduction of text presentation of picture, we make renewal of it's image in the semantically compressed variant.

Ninth principle is ***semantic decompression of graphic information.***

Conclusions

Semantic information, which is contained in graphic presentation, can be extracted by the ontological description of the problem region, built on base drawing primitives and objects, and used after for a compression and decompression of graphic images. For the compression of maintenance of picture (images) translation of graphic arts can also serve as procedural basis in text description with the subsequent semantic compression of text through the programs of type KONSPEKT. Future intellectual technologies of semantic treatment of graphic information must develop taking into account achievements in the area of fox messages and to use the ontological constructions specialized for the graphic appendixes.

Thus, in-process real on results the analysis of types of informative surplus and systematization of technologies of compression of graphic information as their general classification *tasks*, *basic principles of construction* of technology of semantic compression of the graphic information, based on application of semantic compression to its intermediate intermediary forms of presentation, formalization and text description, are indicated, and the variants of its realization are offered.

Bibliography

1. Корриган Дж. Компьютерная графика: Секреты и решения. -М.: Энтроп, 1995.-352с.
2. Балашов К.Ю. Сжатие информации: анализ методов и подходов. // Препринт / Ин-т техн. Кибернетики НАН Беларуси; № 6– Минск, 2000. – 42 с.

3. Валькман Ю.Р., Быков В.С. О моделировании образного мышления в компьютерных технологиях: общие закономерности мышления. // Сборник научных трудов KDS-2005 20-30 июня 2005 г., т.1., София, 2005, с.37-45.
4. Гладун В.П., Величко В.Ю. Конспектирование естественно-языковых текстов. // Сборник научных трудов KDS-2005 20-30 июня 2005 г., т.2., София, 2005, с.344-347.
5. Леонтьева Н.Н., Семенова С.Ю. Семантический словарь РУСЛАН как инструмент компьютерного понимания**.** // Понимание в коммуникации. Материалы научно-практической конф. 5-6 марта 2003 г. — М., МГГИИ, 2003. – С.41-46.
6. Жермен-Ликур П., Жорж П., Пистр Ф., Безье П. Математика и САПР: в 2-х кн. Кн.2-М.: Мир, 1988.- 264с.
7. Люгер Дж.Ф. Искусственный интеллект: Стратегии и методы решения сложных проблем. - М.: Изд. дом "Вильямс". 2003.- 864с.
8. Эгрон Ж. Синтез изображений : Базовые алгоритмы. -М.: Радио и связь, 1993.-216с.
9. Роджерс Д. Алгоритмические основы машинной графики. - М.: Мир, 1989. -512с.

Author's Information

Nikolay Borysovitch Fesenko – V.M.Glushkov Institute of cybernetics NAS of Ukraine, Kiev-187 ГСП, 03680, Akademik Glushov avenue, 40,e-mail: glad@aduis.kiev.ua

AUTOMATED PROBLEM DOMAIN COGNITION PROCESS IN INFORMATION SYSTEMS DESIGN

Maxim Loginov, Alexander Mikov

Abstract: An automated cognitive approach for the design of Information Systems is presented. It is supposed to be used at the very beginning of the design process, between the stages of requirements determination and analysis, including the stage of analysis. In the context of the approach used either UML or ERD notations may be used for model representation. The approach provides the opportunity of using natural language text documents as a source of knowledge for automated problem domain model generation. It also simplifies the process of modelling by assisting the human user during the whole period of working upon the model (using UML or ERD notations).

Keywords: intellectual modeling technologies, information system development, structural analysis of loosely structured natural language documents.

ACM Classification Keywords: I.2 Artificial Intelligence: I.2.7 Natural Language Processing – Text analysis

Introduction

The term "Problem domain" is usually used when the problem of Information Systems (IS) design is discussed. This term represents the aggregation of knowledge about objects and active subjects, tied together with specific relations and pertaining to some common tasks.

Usually the scope of the problem domain is not described strictly and different problem domains intersect. Let us take two problem domains for example: a school education service and a public health service.

An information system designed for automating reporting at schools and another one designed for decision-making for health authorities of a city council can not be completely independent. There are medical consulting rooms at schools and the rate of sickness certainly depends on the school working conditions and so on. After all,

both information systems share some personality information: many people are citizens and students at the same time.

Nevertheless, a description (a model) of the problem domain is a very important part of an information system project. But, anyway, if this model is not comprehensive then it is incomplete.

Documents and experts usually play a part of the knowledge sources circumscribing the problem domain. There are several types of documents that may be used: legal documents, ones that describe business processes, databases of employees and customers, etc. Human experts may provide information on informal rules, conventions, relative importance of concepts, etc, in the given problem domain. Documents of listed types denote objects and formalize some relations in the problem domain concerned. To a first approximation they may be considered as local models of these relations.

The difficulty is that most local models are built using different approaches, because there is no unified approach that may be applied to a problem domain (excepting some narrow-ranged technical domains, where local models can be combined together into a global model using some strict mathematical rules; information systems built upon such problem domains are called “systems of automatic control”).

We are concerned here about information systems of a different kind – systems where the human element is of primary importance. Investigation into such kinds of problem domains is a type of empirical research, related to the “sciences of the artificial”.

Nowadays most CASE tools (Computer-Aided Software Engineering tools) can automatically build source program code for a projected information system, using some initial formal model of the problem domain (usually the model is represented as a framework, or graph). The urgent problem is to automate the process of building the formal model, e. g. to automate the process of cognition in the given problem domain.

Goals of the Research

The main purpose of this research is development of the special cognitive approach, referred to a class of Intellectual Modelling Technologies (IMT). This approach is designed for automating the process of information system development. Attention is focused on the very early stage of project development, the stage of analysis.

The problem domain of the class of IS under consideration includes a very large amount of legal documents (articles, assizes, bans, etc.), which regulate the status of objects, the behaviour of subjects related to an institution, etc. It also includes a settled system of document circulation. All this information, as a rule, is poorly structured. So, the development of a conceptual model of the problem domain (by means of UML language, for example) using knowledge from documents of these types, is a very difficult task and usually is done manually.

The suggested cognitive approach is aimed toward the problem of automating the conceptual-level model development by using loosely structured natural language documents.

Since the problem under consideration refers to a class of logical lexical systematization problems (as an example from the adjacent area of study we may take translation of natural language text into the language of predicate logic), it has no solution using only a computation system. That is why the suggested approach is developed to work in conjunction with the human user. Human interference is needed during the automated analysis of problem domain described in source documents. Nevertheless, some self-learning capabilities in the context of approach allow us to depend on the self-development of the analyzer during persistent dialogue with the human user, so that subsequently it could be able to solve similar tasks without direct human assistance.

The suggested approach is oriented to be utilized at the earliest stage of the information system project development process. As indicated on fig. 1, the suggested approach is supposed to be used at the very beginning of the spiral loop, at the boundary between the stages of requirements determination and analysis, also including the stage of analysis.

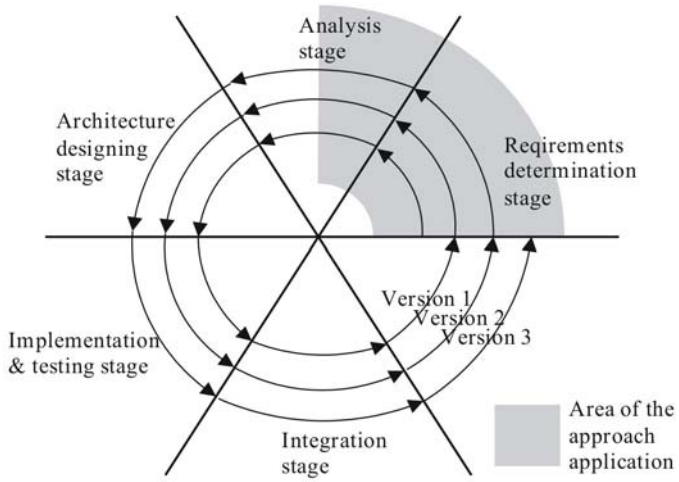


Figure 1. The Spiral Model of Software Life Cycle

It is important to mention that most existing IMT methods, used in CASE systems, automate, in general, stages of projecting, implementing, testing and integrating, but never touch the stages of requirements determination and analysis. Transition from the stage of requirements determination to the early stage of analysis is usually done by hand. Then the user develops a conceptual level model of the problem domain.

The model is developed usually using some special diagram language (UML language, for example). Conceptual level models describe a part of the real world for which the information system is being developed, so conceptual level class diagrams are right for describing the set of terms of the problem domain vocabulary.

When developing a model, the analyst usually processes by hand a large amount of documents referred to the problem domain in order to pick out key terms, properties, functions and relations between them. The proposed approach enables automation of this process. The intellectual cognitive analyzer being implemented according to the described approach should act as a user's assistant. It will do the most routine part of the work in the early stage of analysis.

The suggested approach also includes some other capabilities that let the computer become quite a good assistant for the human user not just at the beginning of analysis, but along its whole length. One of those capabilities, in particular, is the automatic problem domain thesaurus building during interaction with the user. And it is possible to use preinstalled thesauruses too, different ones for each problem domain, describing their specific components, features, etc.

Conceptual-level Modeling

As was said earlier, the purpose of the approach is automated construction of conceptual-level model diagrams of the problem domain. The UML language (static class section), was chosen as a model representing language, because it is the most popular standard for CASE tools nowadays.

UML static class diagrams define object classes and different kinds of static relations between them in the system. Also such things as class attributes, operations and relation limitations are usually shown on class diagrams.

There are three different points of view on how to interpret the sense of class diagrams:

- Conceptual-level point of view. If we take a class diagram from this point of view, then it reflects a set of terms and relations (called vocabulary) of the examined problem domain. Conceptual-level model considered independent from any software programming language.
- Specification-level point of view. In contrast to the above, this affects the software development range, but focuses attention over interfaces, not implementation. Looking at the class diagram from this point of view, designers have to do rather with types, not classes.

– Implementation-level point of view. In this case we really deal with graphical representation of the class structure of software. So the designer goes down to the level of implementation.

Understanding which point of view should be used and when, is extremely important either for developing or for reading class diagrams. Unfortunately, distinctions between them are not understood clearly, so the majority of developers often mix some different points of view when developing a diagram model.

The idea of the point of view on diagrams is not actually a formal part of UML language, but it is extremely important. UML constructions can be used with any of the three points of view in mind.

As has already been said, the suggested approach is going to be used for the automation of the process of conceptual-level problem domain model development. First of all, it is because of the fact that the approach should work at the most initial stage of IS development process. Apart from that, the nature of the documentation used in the problem domain of the considered range of IS (sphere of education) means that the description of objects and their mutual relations is of a sufficiently high level. This fact automatically determines the point of view on a problem domain as conceptual.

However, such a strict binding model to a conceptual level is not obligatory. In some cases the model can get an interpretation from some other point of view. This mainly depends on the nature of the source documents.

Conceptual-level diagrams describe the problem domain vocabulary. Of course, it is doubtful that diagrams developed using the suggested approach could be immediately used for generation of skeleton program code, but it can be used for subject domain database logic structure generation.

IES Architecture

Fig. 2 shows the diagram reflecting the principle according to which the projected system is organized.

Let us consider in more detail the principles assumed for the basis of the suggested approach.

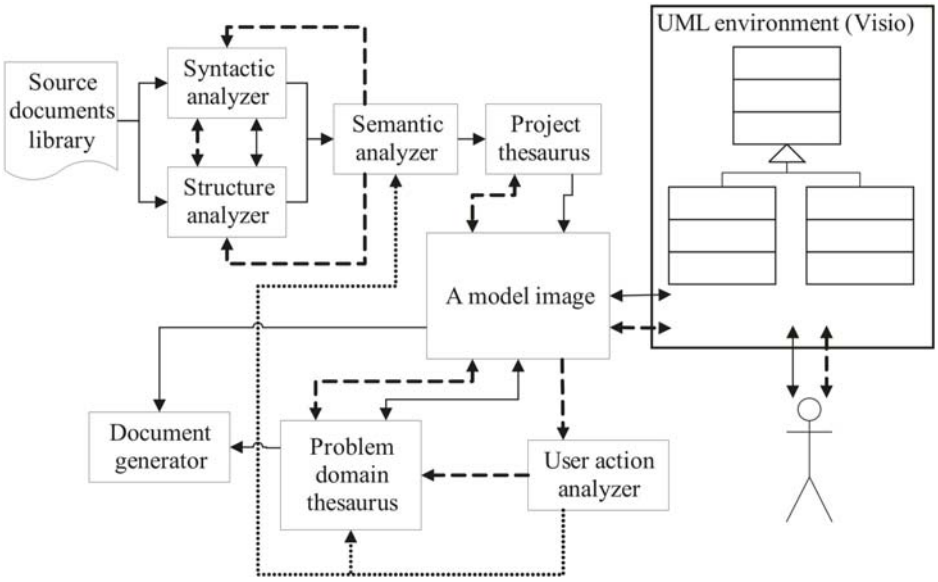


Figure 2. IES Architecture Framework

Natural language expresses relations between items in a problem domain in the form of statements. For example, the statement “children study at schools” binds together the concept “school” belonging to the class “educational institutions” and the concept “children” belonging to the class “person”. Any statement can be either correct, or wrong, when established during correlation with reality. So, statements singled out from source documents should be compared to the problem domain thesaurus which reflects the current actuality. In the case of detection of a discrepancy of the obtained propositions to ones from the problem domain thesaurus, the latter should be

brought into accord with reality, or the source proposition should be corrected in an appropriate way. When the system cannot make such a decision independently, it can apply for the human user's assistance.

The proposition (statement) is an expression that claims or disclaims the existence of an item, the relation between an item and its attribute, or the relation between two items. A sentence is the language form of the proposition. Propositions in natural language texts are expressed by narrative sentences, for example: “institutions of primary vocational training may be state, municipal and private”. The proposition of connexion of an item and its attribute consists of propositional subject, and a predicate reflecting an attribute of an item. Except for subject and predicate, the proposition includes a copula which can be put into words (for example, “not is”, “is”, etc.).

Depending on what is claimed or disclaimed in the proposition – either the existence of an item, or the relation between an item and its attribute, or the relation between two items – it may be classified as attributive proposition, proposition with relation and existential proposition. A proposition is called compound if it consists of several simple propositions, combined together in an expression.

A conceptual-level model is usually developed using source natural language description. Sentences in this description are propositions of listed types. Some of them concern certain objects; others are general as they concern some class of objects in the problem domain. Source documents consist of compound sentences that describe objects and relations between them in the problem domain.

In the course of linguistic analysis using knowledge of language structure, initial compound sentences are split into simple propositions of three listed types, and the type of each proposition can be determined during the process of decomposition.

The whole totality of concepts and relations between them, expressed by means of natural language, forms a system thesaurus. Thus, we can say it schematically represents the matter of the source documents text. The idea of a thesaurus is frequently applied to problems of semantic search in documents. Within the suggested approach, another variant of its application is offered.

Concepts are extracted from source documents during the linguistic analysis process. One of the basic relation types that make a thesaurus hierarchical, is the relation type named “is a”. It realizes the idea of generalization, allowing reference of a narrower concept to a wider one.

Another relation type named “referred to” designates a certain reference between terms. It can be marked by a verb (predicate) extracted from the source sentence (this verb should describe the nature of the relation with certainty). This mark can also be a word-combination, consisting of a verb and a noun: “is operation”, “is an attribute”.

To avoid the possibility of the appearance of a vicious circle of interdependence of terms in a thesaurus, there are some rules to be obeyed:

- no term can be wider than itself, either directly, or indirectly (this limits the usage of the “is a”);
- no term can be “referred to” a term which is wider or narrower than itself, either directly, or indirectly.

The structure of the thesaurus can be represented by a graph (semantic net), its nodes correspond to terms, and arches are relations between terms. One set of arches forms a directed acyclic graph for the relation of generalization (“is a”). Another set forming the directed graph, represents the relation of referred terms (“referred to”). Relation types “is a” and “referred to” form subgraphs.

Principles of IES Operation

The thesaurus of the model should be populated and refreshed using an automatic mode. Thus there are the certain difficulties concerning natural language text processing. To overcome these difficulties successfully, the approach offers the multilevel circuit of text processing using the relaxation method to eliminate ambiguities.

At the initial stage of text processing the syntactic analyzer (figure 2) works. It implements the syntactic analysis and decomposition of compound sentences to the simple propositions consisting of subject, predicate and object. While these operations are being accomplished, the semantics of the sentence is not taken into account. During decomposition, the text of the source documents is transformed into a set of simple statements (propositions) of

three listed types (attributive proposition, proposition with relation, existential proposition) which then can be easily subjected to semantic analysis.

It is important to note that relations between concepts are not necessarily conveyed syntactically in text. They can also be conveyed by the structure of the document. There are two types of structural compositions most frequently used in documents: table structure, determines attributive relations; list structure, determines relations of various kinds, between the concept located in the list heading and concepts located in lines.

In order to assure the completeness of analysis it is necessary to allocate relations, set by structures of listed types. This task is done by the structural analyzer, whose output, as well as for syntactic analyzer, consists of simple propositions reflecting relations between concepts. The analyzer generates them using structural information extracted from the source text as the basis.

The semantic analyzer obtains the data processed by syntactic and structural analyzers, handles them for its turn and populates the system thesaurus with prepared data. If the semantic analyzer finds any variance in source data, caused by its ambiguity or uncertainty, it can address previous level of processing – syntactic or structural analyzer – with the requirement to give another variant of the text treatment. This idea accords with principles of the relaxation method. Some missing branches of concept relations may also be evoked from the existing thesaurus knowledge base.

There is one more task assigned to the semantic analyzer – to eliminate insignificant data. In fact the final model should not be formed by the whole totality of concepts and relations, allocated in the initial documents. First of all, some concepts may just slightly touch the scope of the given problem domain. Sometimes some errors in allocation of concepts and relations may take place because of text specificity or its author's verbiage. Anyway, some mechanism is required that could free the user from dealing with a lot of insignificant details. To achieve this, the semantic analyzer uses a special self-learning filter as a part of the project thesaurus. This filter determines a list of concepts that should not be included in the thesaurus. Relations of a special type “not relevant” may also be settled between the concepts in the thesaurus in order to solve the problem more effectively.

The filter is trained by tracking actions which are user made when editing a diagram. This way we can reveal insignificant concepts and relations in the problem domain to use this knowledge later.

We need to mention that there is one more important opportunity the approach can offer: an opportunity of distribution “on a turn-key basis” of an IS designing tool assigned for usage in the context of a certain problem domain. Such a tool would possess a prepared thesaurus establishing the set of basic concepts and relations and include a trained semantic filter focused over the scope of the problem domain being aimed at. In the architecture framework of IES which is being developed according to the suggested approach, this thesaurus is represented by the separate component called “Problem domain thesaurus” (figure 2).

The project thesaurus directly delivers data needed for production of model diagrams. The structure and sense of the thesaurus content allows translation of it into the model diagram. This is in spite of the fact that there are some minor distinctions between specifications of diagrams that could be used within the approach: UML diagrams and ER diagrams.

Diagrams are displayed in some external modelling environment which is connected to IES through the special buffer module of the model image. Of course, the user may want to correct the obtained model diagram, which is initially generated by the system. But nevertheless, it continues to cooperate with the user, helping him to perform the work.

Upon the user's demand it can generate some new fragments of the model diagram, if there are any new source documents obtained, or expand the model diagram in order to restore missing relations, applying knowledge from the problem domain and the project thesauruses, etc.

The system also traces user's actions made during model diagram editing. Such a feedback mechanism is absolutely necessary for implementing the idea of self-training as applied to the problem domain thesaurus and

the semantic filter. Actually, during editing of the model diagram, the user "teaches" the system, providing it with the information about concepts and relations that are of first interest to him and ones that should be excluded from consideration. In such a way, the problem domain thesaurus containing the most authentic and clean information on key concepts and typical relations between them is built. It is populated automatically during editing of the model diagram. Thus, the resulting model diagram and successive modifications made by the user are also a source of information for the IES.

The system tries to recognize semantics in the model diagrams. So a diagram which the user works with is not a senseless set of blocks and connections for a computer any more. Attention is focused on names of elements, their structure, interfacing, etc. All these aspects are analyzed by the system.

Objects and relations allocated in a problem domain, organize a model. When the diagram is built, they remain connected with texts in the source documents library. It is necessary for the user to have an opportunity to supervise the correctness of the constructed model, verifying it directly with the key information from source documents. Reverse referencing from source documents to elements of a model diagram is also needed, because documents are not something immutable. The documents library has a dynamic nature – precepts may be cancelled, or changed in some points, etc. Direct and reverse referencing between source texts and the model assure an opportunity of efficient model updating.

Examples

Now we give an example demonstrating some aspects of the approach.

Please note that the approach is being developed for use jointly with the Russian language, where the concepts' mutual interdependence in sentences is expressed much less ambiguously than in English, at the syntax level.

Let us show how a certain expression is going to be analyzed by the system:

"Educational institutions with government accreditation grant certificates to persons who passed attestation".

During syntactic analysis the given sentence is split into some connected simple statements which can be easily represented by the semantic network shown on fig. 3.

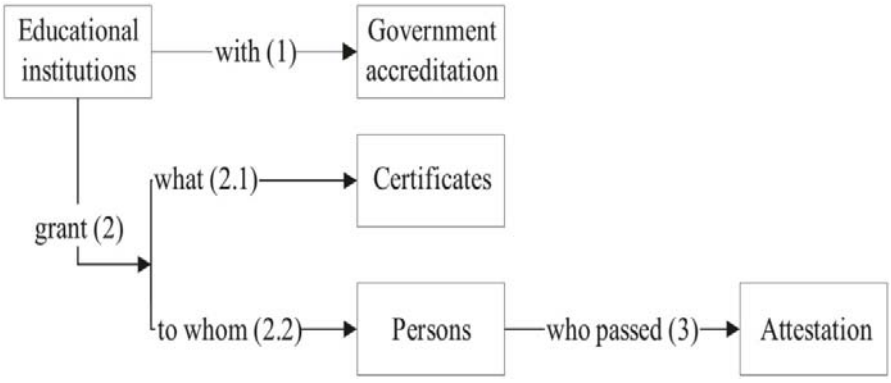


Figure 3. Semantic Network Representing Sample Sentence Structure

The semantic analyzer qualifies propositions (1, 2 and 3) such as ones with relations. Thus the verb predicate representing the action "grant" is interpreted by the semantic filter as an operation (class method). But let us assume that such interpretation is not known to semantic filter.

Simple propositions obtained which form marked section of a semantic network after the stage of semantic analysis, are directed to the problem domain thesaurus.

Propositions with relations of such a kind are displayed in the model as objects connected by the relation "referred to"; connection is directed from a subject to an object and represents the predicate (see fig. 4).

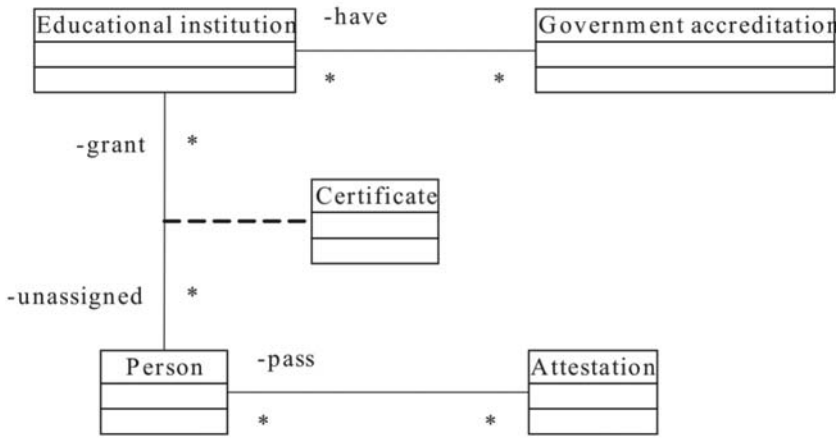


Figure 4. Model Framework Created on the Sentence

Part of the model received as a result of analysis of a given sentence, could be automatically attached to the existing model by a set of "is a" connections, revealed by the semantics comparison.

Besides that, if the problem domain thesaurus contains information about other connections between these objects and ones in the problem domain, these connections will also be restored in the model.

So, let us return to the necessity that the action "grant" should be interpreted as a method.

If it does not happen automatically, then the user manually creates the method "grant" in the object "Education institution". After that, as a result of the semantics comparison of the operation name assigned by the user with the text of source sentence, the semantic filter is trained to interpret the verb "to grant" as the method (operation) at a later time.

Analyzing a similar text subsequently, the system should automatically add a corresponding object operation to the model. The thesaurus of the model is populated and refreshed in an automatic mode.

Bibliography

[Aiello, 2000] M. Aiello, C. Monz, L. Todoran. Combining Linguistic and Spatial Information for Document Analysis, In: Content-Based Multimedia Information Access. CID, 2000, pp. 266-275.

[Connolly, 1999] T.M. Connolly, C.E. Begg. Database Systems. A Practical Approach to Design, Implementation, and Management. Addison Wesley Longman, Inc, 1999.

[Fowler, 1997] M. Fowler, C. Scott. UML Distilled: Applying the Standard Object Modeling Language. Addison Wesley Longman, Inc, 1997.

[Johnson-Laird, 1983] P.N. Johnson-Laird. Mental Models: Towards a Cognitive Science of Language, Inference, and Consciousness. Harvard University Press, 1983.

[Katz , 1964] J.J. Katz, J.A. Fodor. The Structure of Language, Prentice-Hall, 1964.

[Larman, 2000] C. Larman. Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design. Prentice Hall, Inc, 2000.

[Miller, 1990] G. Miller. Wordnet: An on-line lexical database. In: International Journal of Lexicography, 3(4), 1990.

Authors' Information

Maxim Loginov - Perm State University, Assistant of the Department of Computer Science; PSU, 15, Bukirev St., Perm, 614990, Russia; e-mail: login@perm.ru

Alexander Mikov – Institute of Computing, Director; 40/1-28, Aksaiskaya St., Krasnodar, Russia; e-mail: alexander_mikov@mail.ru

GENERALIZING OF NEURAL NETS: FUNCTIONAL NETS OF SPECIAL TYPE

Volodymyr Donchenko, Mykola Kirichenko, Yuriy Krivonos

Abstract: Special generalizing for the artificial neural nets: so called RFT – FN – is under discussion in the report. Such refinement touch upon the constituent elements for the conception of artificial neural network, namely, the choice of main primary functional elements in the net, the way to connect them(topology) and the structure of the net as a whole. As to the last, the structure of the functional net proposed is determined dynamically just in the constructing the net by itself by the special recurrent procedure. The number of newly joining primary functional elements, the topology of its connecting and tuning of the primary elements is the content of the each recurrent step. The procedure is terminated under fulfilling "natural" criteria relating residuals for example. The functional proposed can be used in solving the approximation problem for the functions, represented by its observations, for classifying and clustering, pattern recognition, etc. Recurrent procedure provide for the versatile optimizing possibilities: as on the each step of the procedure and wholly: by the choice of the newly joining elements, topology, by the affine transformations if input and intermediate coordinate as well as by its nonlinear coordinate wise transformations. All considerations are essentially based, constructively and evidently represented by the means of the Generalized Inverse.

Keywords: Artificial neural network, approximating problem, beam dynamics with delay, optimization.

ACM Classification keywords:F.1.1.Models of Computation: Self modifying machines(neural networks; G.1.6. Optimization; H.1.m. Models and principles; I.2.6. Artificial Intelligence: learning, connectionism and neural nets.

Introduction

Artificial neurons nets are the technological elements, used in various applications ([Amit, 2002], [Veelenturf, 1995] for example) especially under model uncertainty. It may be approximation problem for the functions represented by its observations, the task of the control, classification problem and so on.

The power of the artificial neural nets (ArtNN) substantially specified by the using virtually the composition of the functions [Donchenko, Kirichenko, Serbaev, 2004]. The might of this instrument was certified by [Kolmogorov, 1966] and [Arnold, 1967].

But some faults of the ArtNN are obvious, namely, the constraints on the primary functional elements represented by neurons, constraints on topology, constraints on the optimizations, properly in the appearance of the Back Propagation.

In the report the attempt is undertaking to extend the ArtNN up to Functional nets taking as the primary the special functional element – so called ERRT: elementary recursive regressive transformation, spreading the topology and introducing "natural" optimizing parameter.

The fulcrum for the implementing this attempt arises from the theory of Generalized Inverse Matrixes (for ex. [Алберт, 1976]) developed by one of the authors and represented in [Кириченко, Лепеха, 2002].

The approach proposed is realized in the conception of so called Recursive nonlinear Functional Transformation - Functional Net (RFT-FN. The idea of such type transformations in the variant of inverse recursion was proposed in [Кириченко, Крак, Полищук, 2004]], focused on the optimizing procedure for ERRT. We represent here another variant of RFT - FN – the variant of so called inverse recursion, introduced and discussed in the paper [Donchenko, Kirichenko, 2005] by the authors. As it has been already noticed earlier, RFT – FN embodies the main ideas of the classical ArtNN: using the composition of standard primary functional elements (artificial neurons – a.n.) connected according some topology for constructing a complex object. The primary elements of the composition: a.n. – represents mathematically comparatively simple function: a composition of linear function – linear functional for input multi-dimensional variable, – and simple standard scalar function. Conception of the RFT-FN leaves the main idea of ArtNN unchangeable: that the idea of constructing a complex objects through a composition of standard simple ones. But it is proposed and implemented some substantial refinements.

The main of them are actually in the next:

- Expanding the domain of the feasible functions for the basis standard elements and introducing the special procedure for the adjusting for the newly connecting ERRT;

Introducing the special recursive dynamic procedure for the constructing the RFT-FN. The term "dynamic" means the nor a priory structure is fixed for the RFT-FN, but it determines exclusively by the quality of the constructions, characterized by some "natural" functional of the quality, the residuals in the approximation for example. So the termination of the recursive procedure of the constructing the RFT-FN determines dynamically in the in the course of the procedure by itself.

- Expanding the variants of the feasible connections for the newly connecting elements and its number.

There some more additional enhancements for the ArtNN, proposed and implemented in the conception of the RFT – FN. These are: coordinate wise nonlinear transformations of the input and intermediate inner coordinate along with the linear transformations of them.

All the refinements make it possible to optimize the constructing of the RFT-FN on the each recursive step and wholly by the next ways:

- By the choice of the feasible functions for the ERRT, by the choice of linear transformations of its coordinates and coordinate wise nonlinear transformations of the coordinate for the input variables for the newly connected ERRT: single or a number of them;
- By the choice of a topology for newly connected elements. There are three mains types of the connections for the newly connecting elements: parallel by input (parinput), parallel by output (paroutput) or sequential (seq).

The implementation of the RFT–FN conception will be demonstrated on the approximation problem for the functions, represented by its observations. Draw the attention of the reader that this problem is represented widely in the publications as in deterministic as well in statistical enunciations. As regarding the last, we refer to [Линник, 1962], [Вапник, 1979], [Ивахненко, 1969].

1. General Conception of the Functional Net: RFT -FN

The RFT-FN constructing procedure conclude in joining the recurrently EERT to RFT-FN already has been constructed during previous steps in compliance with one of the three certain types of connecting,

These types will be denoted as parinput, paroutput and seq. The connecting types correspond to natural transformations of input signal: parallel by input or output, and also sequential.

1.1. Description of the Primary Functional Elements: ERRT

The basic constructive element for RFT-FN is the ERRP-element [9], which is defined as a map from R^{n-1} in R^m designed according to the next form:

$$y = A_+ \Psi_u \left(C \begin{pmatrix} x \\ 1 \end{pmatrix} \right), \qquad (1)$$

It approximates the dependence, represented by the learning sample

$(x_1^{(0)}, y_1^0), \ldots, (x_M^{(0)}, y_M^{(0)})$, $x_i^{(0)} \in R^{n-1}$, $y_i^{(0)} \in R^m$, $i = \overline{1, M}$, where:

C is (n×n))-matrix, defining an affine map between R^{n-1} and R^m, fixed when synthesizing the ERRT;

Ψ_u – coordinate-wise nonlinear map from R^n in R^n; each of nonlinear real functions u_i, $i = \overline{1, n}$, transforming the coordinates, belongs to finite set $\Im$ of the functions. This set is fixed, but open for extension. The set $\Im$ include also identity function. We will consider each of such functions to be smoothly enough; one chose the nonlinear map Ψ to minimize the discrepancy between input and output on the learning sample;

$A+$ – trace-norm minimal solution of the next matrix equation

$$AX_{\Psi_u C} = Y, \qquad (2)$$

in which $X_{\Psi_u C}$ – matrix assembled from vectors-columns $\Psi_u(C\begin{pmatrix} x_i^{(0)} \\ 1 \end{pmatrix}) = \Psi_u(z_i^{(0)})$ and matrix Y – from $y_i^{(0)}, i = \overline{1, M}$

1.2. Recursive Procedure and Topology of Connections in RFT – FN Constructing

Composition-recursion in function-building procedure in the proposed below variant of direct recursion will be considered in generalized version. In such version the number $\mathrm{k_m}$ of newly joined ERRT may be more, than one. Total number of ERRP used will be denoted by T: $T = \sum_{m=1}^{N} k_m$, N being the number of recursive calls of the procedure.

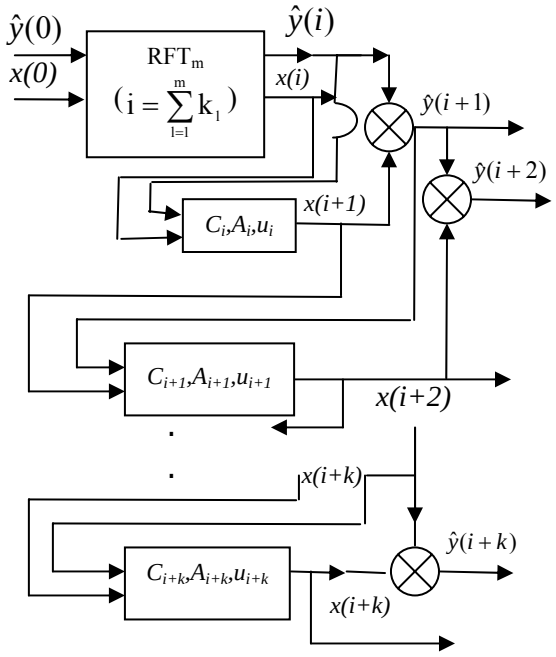


Fig. 1. Parinput

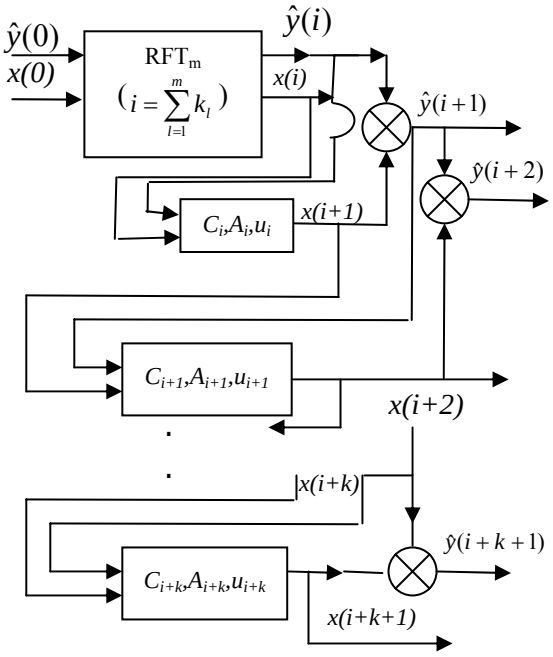


Fig. 2. Paroutput

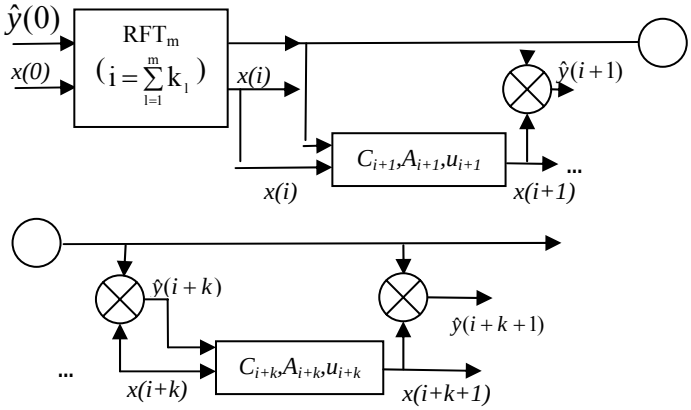


Fig. 3. Seq

Direct recursion is represented by the figures 1, 2, 3 and corresponding equations depending on type of the joining.

- Parinput (fig. 1)

The input-output equations represented parinput topology in the recursive procedure are tot the next form:

$$\begin{aligned} x(i+j) &= A_{+i+j-1}\Psi_{u_{i+j-1}}(C_{i+j-1}x(i)), \\ \hat{y}(i+j) &= \hat{y}(i+j-1) + A_{+i+j-1}\Psi_{u_{i+j-1}}(C_{i+j-1}\cdot x(i)), \\ i &= \sum_{l=1}^{m} k_l,\ j = \overline{1,k_{m+1}} \end{aligned} \qquad (3)$$

- Paroutput (fig. 2)

Paroutput topology is calculated by the next input-output equations:

$$\begin{aligned} x(i+j) &= A_{+i+j-1}\Psi_{u_{i+j-1}}(C_{i+j-1}x(i+j-1)), \\ \hat{y}(i+j) &= \hat{y}(i+j-1) + A_{+i+j-1}\Psi_{u_{i+j-1}}(C_{i+j-1}\cdot x(i+j-1)), \\ i &= \sum_{l=1}^{m} k_l,\ j = \overline{1,k_{m+1}}\ . \end{aligned} \qquad (4)$$

- Seq (fig.3)

And at last for sequential type we have correspondingly:

$$\begin{aligned} x(i+j) &= A_{+i+j-1}\Psi_{u_{i+j-1}}(C_{i+j-1}x(i+j-1)), \\ \hat{y}(i+j) &= \hat{y}(i) + A_{+i+j-1}\Psi_{u_{i+j-1}}(C_{i+j-1}\cdot x(i+j-1), \\ i &= \sum_{l=1}^{m} k_l,\ j = \overline{1,k_{m+1}} \end{aligned} \qquad (5)$$

2. Special Class of Beam Dynamics with Delay

The optimization for RFT – FN as it is follows from (3)-(5)is reduced virtually to solving the optimization problem for the beam dynamics of special type, determined below. Namely, we will introduce and consider two classes of special discrete dynamic systems with delay named below simple and combined. Classical results about conjugate systems and Hamilton functions will be extended on the systems introduced as well as the results about functional differentiating respectively controls.

2.1. Special Class of Beam Dynamics with Delay : Basic Definitions

These two types of beam dynamics: with simple delay and combined – are defined in the next way:

- simple delay:

$$x(j+1) = f(x(j-s(j)), u(j), j)\ , \qquad (6)$$

combined:

$$x(j+1) = f(x(j), x(j-s(j)), u(j), j)\ ;\quad j = \overline{0,N-1}\,; \qquad (7)$$

set of the initial states $\mathrm{Ini} = \{x_1^{(0)},\ldots,x_M^{(0)}\}:\ x(0) = x^{(0)} \in \mathrm{Ini};$

- functional on the set of the trajectories

$$I(U) = \sum_{x^{(0)} \in Ini} \Phi(x(N)), \qquad (8)$$

delay function $s(j) \in \{2,..., j\}$, $s(0)=0, s(j) \in \{2,..., j\}$, $j = \overline{0, N-1}$.

Simple systems are defined by the collection of the functions $f(z, u, j)$, $j = \overline{0, N-1}$ and combined ones – by the $f(z, v, u, j)$, $j = \overline{0, N-1}$.

2.2. Conjugate Systems and Hamilton Function

Given the system dynamics with delay: simple or combined – define the conjugate systems and the Hamilton functions depending on the type of the delay beam dynamics.

Simple delay:

- Conjugate system $p(k)$, $k = \overline{N,0}$

$$p(N) = -grad_{x(N)} \Phi(x(N)),$$

$$p(k) = \sum_{j \in J(k)} grad_{x(k)} \{p^T(j+1) f(x(j), u(j), j)\},$$

$$J(k) = \{j : j - s(j) = k, j \geq k\}, \quad k = \overline{N-1,0};$$

- Hamilton function:

$$H(p(k+1), x(k-s(k)), u(k), k) = p^T(k+1) f(x(k-s(k)), u(k), k), \quad k = \overline{N-1,0}.$$

Combined delay:

- Conjugate system $p(k)$, $k = \overline{N,0}$

$$p(N) = -grad_{x(N)} \Phi(x(N)),$$

$$p(k) = grad_z \{p^T(j+1) f(x(k), x(k-s(k)), u(j), j) + \sum_{j \in J(k)} grad_v \{p^T(j+1) f(x(k), x(j), u(j), j)\},$$

J(k) is the same as for simple systems, $k = \overline{N-1,0}$;

- Hamilton function:

$$H(p(k+1), x(k), x(k-s(k)), u(k), k) = p^T(k+1) f(x(k), x(k-s(k)), u(k), k).$$

2.3. Gradient in Beam Dynamics with Delay

The classical results take place for the beam dynamics with delay within the classical assumptions as to $f(z,u,j)$ or $f(z,v,u,j)$, $j = \overline{0, N-1}$ and Φ. These results are captured in the next two theorems.

Theorem 1. For the simple delay beam dynamics gradients respectively controls are represented by the next relations

$$grad_{u(k)} I(U) = -grad_{u(k)} \sum_{i=1}^{M} H^{(i)}(p^{(i)}(k+1) x^{(i)}(k), u(k), k), \quad k = \overline{0, N-1}.$$

Theorem 2. For the combined delay beam dynamics gradients respectively controls are represented by the next relations

$$grad_{u(k)} I(U) = -\sum_{i=1}^{M} \text{grad}_{u(k)} H^{(i)}(p^{(i)}(k+1), x^{(i)}(k), x^{(i)}(k - s(k), u(k), k)), \quad k = \overline{0, N-1}.$$

Index i: $i = \overline{1, M}$ corresponds to trajectories with initial states $x_i^{(0)} \in Ini$.

3. Functional Nets and Beam Dynamics with Delay

Combined delay beam dynamics are very important regarding their role in representation of RFT-FN constructions.

Theorem 3. RFT-FN – predictor with the direct N recursions in using $k_m, m = \overline{1, N}$ ERRT respectively can be represented by the combined delay beam dynamics on the time interval $\overline{0, T}$, $T = \sum_{l=1}^{N} k_l$. The elements of such representations are constructive, depending on the type of the joining. The quality functional of the system is of the next form

$$I(C) = \sum_{k=1}^{M} \| y_k^{(0)} - z_2(T) \|^2,$$

where $z_2(T)$ is one of two output components for the beam dynamics.

4. Functional Nets Optimal Design

Theorem 3 enables to choose optimally $C_0, C_1, ..., C_{T-1}$ for RFT – FN.

Theorem 4. Under assumption that any element from $\Im$ has the continuous second-order derivation, an RFT-FN is feasible to be constructively optimized by gradient methods respectively matrixes C.

Conclusion

Special kind of the "functional nets": so called RFT – FN, generalizing classical functional nets, namely, artificial neural nets, has been proposed and investigated in the report. The RFT – FN permit multilevel optimization. First level optimization is the optimization in the primary functional element and the pseudo inverse is principally in this stage. Another principal level of the optimization is the optimization due to optimization in the beam dynamics.

Bibliography

[Amit, 2002] Y. Amit., 2D Object Detection and Recognution: models, algorithms and networks.–The MIT Press, Cambridge, Massachusetts.– 2002.–306 p.

[Veelenturf,1995] L.P.J.Veelenturf, Analysis and applications of artificial Neural Networks.–Prentice Hall (UK).– 1995.–259 p.

[Donchenko, Kirichenko, Serbaev,2004] V.S. Donchenko, N.F.Kirichenko,D.P.Serbaev. Recursive regression transformations and dynamical systems.// Proceedings of the Seventh International Conference "Computer Data Analysis and modelling: robustness and computer intensive methods".– V.1.–September 6-10, 2004. – Minsk (Belarus).– p.147-151.

[Колмогоров,1956] А.Н.Колмогоров О представлении непрерывных функций нескольких переменных суперпозициями непрерывных функций меньшего числа переменных.// Доклады АН СССР. – 1956. – Т. 108. – № 2. – С. 179- 182.

[Арнольд,1957.] В.И. Арнольд О функциях трех переменных.// Докл. АН СССР. – 1957. – Т. 114. – № 4. – С. 953- 956.

[Албert,1977] А.Албert Регрессия, псевдоинверсия, рекуррентное оценивание.– Пер.с англ.– М.: Наука.–1977.–305с.

[Кириченко,Лепеха,2002] Н.Ф. Кириченко, Н.П.Лепеха Применение псевдообратных и проекционных матриц к исследованию задач управления, наблюдения и идентификации.// Кибернетика и системный анализ. – 2002. – № 4. – с. 107-124.

[Кириченко, Крак, Полищук,2004] Н.Ф.Кириченко, Ю.В. Крак.А.А. Полищук Псевдообратные и проекционные матрицы в задачах синтеза функциональных преобразователей.// Кибернетика и системный анализ –2004.–№3.

[Donchenko, Kirichenko,2005.] V.S Donchenko, N.F. Kirichenko Generalized Inverse in Control with the constraints.// Cybernetics and System Analysis.– v.39 (6) November-December– 2003.–p.854-861(USA).

[Линник,1962] Ю.В.Линник Метод наименьших квадратов и основы математико-статистической теории обработки наблюдений. Изд. 2-е. – М.:Физматгиз.– 1962.– 349 с.

[Вапник,1979] В.Н. Вапник. Восстановление зависимостей по эмпирическим данным. – М.: Наука. – 1979. – 447 с.

[Ивахненко,1969] А.Г. Ивахненко Самоорганизующиеся системы распознавания и автоматического управления. – К.: Техника. – 1969. – 395 с.

Authors' Information

Volodymyr Donchenko – Kyiv National Taras Shevchenko University (Ukraine), Professor, e-mail: voldon@unicyb.kiev.ua

Mykola Kirichenko – Institute of the Cybernetics, National Academy of Sciences (Ukraine), Professor

Yuriy Krivonos – Institute of the Cybernetics, National Academy of Sciences (Ukraine), Member Correspondent of the National Academy of Sciences.

NEURAL NETWORK BASED OPTIMAL CONTROL WITH CONSTRAINTS

Daniela Toshkova, Georgi Toshkov, Todorka Kovacheva

Abstract: In the present paper the problems of the optimal control of systems when constraints are imposed on the control is considered. The optimality conditions are given in the form of Pontryagin's maximum principle. The obtained piecewise linear function is approximated by using feedforward neural network. A numerical example is given.

Keywords: optimal control, constraints, neural networks

ACM Classification Keywords: I.2.8 Problem Solving, Control Methods, and Search

Introduction

The optimal control problem with constraints is usually solved by applying Pontryagin's maximum principle. As it is known the optimal control solution can be obtained computationally. Even in the cases when it is possible an analytical expression for optimal control function to be found, the form of this function is quite complex. Because of that reason the possibilities of using neural networks for solving the optimal control problem are studied in the present paper.

The ability of neural networks to approximate nonlinear function is central to their use in control. Therefore it can be effectively utilized to represent the regulator nonlinearity. Other advantages are their robustness, parallel architecture.

Lately, different approaches are proposed in the literature treating the problem of constrained optimal control for using neural networks. In [Ahmed 1998] a multilayered feedforward neural network is employed as a controller. The training of the neural network is realized on the basis of the so called concept of Block Partial Derivatives. In [Lewis 2002] a closed form solution of the optimal control problem with constraints is obtained solving the associate Hamilton-Jacobi-Bellman (HJB) equation. The solution of the value function of HJB equation is approximated by using neural networks.

In the present paper the problem of finding the optimal control with constraints is considered. A numerical example is given.

Problem Statement

The control system, described by following differential equations is considered:

$$\frac{dx_i}{dt} = \sum_{j=1}^{n} a_{ij} x_i + b_i u \qquad (i = 1, 2, \ldots, n) \tag{1}$$

where x_j are phase coordinates of the system, function u describes the control action and a_{ij} are constant coefficients. The admissible control u belonging to the set U of piecewise linear functions is constrained by the condition

$$|u(t)| \le 1 \tag{2}$$

Following problem for finding the optimal control is formulated. To find such a control function $u(x_1,\ldots, x_n)$ for the system (1) among all the admissible controls that the corresponding trajectory $(x_1(t),\ldots, x_2(t))$ of the system (1) starting from any initial state $(x_1(0),\ldots, x_2(0))$ to tend to zero at $t \to \infty$ and the performance index

$$J = \int_0^{\infty} \left(\sum_{i=1}^{n} q_i x_i^2 + ru^2 \right) dt \tag{3}$$

to be converging and to take its smallest possible value. The coefficient qi and r are positive weight constants.

Optimality Conditions

The notation is introduced [Pontryagin 1983]:

$$f_0(x_1, \ldots, x_n, u) = \sum_{j=1}^{n} q_j x_j^2 + ru^2 \tag{4}$$

$$f_i(x_1, \ldots, x_n, u) = \sum_{j=1}^{n} a_{ij} x_j^2 + b_i u \qquad (i = 1,\ldots,n) \tag{5}$$

One more variable θ_0 is added to the state variables $(x_1,\ldots, x_n)$ of the system (1) [Chjan 1961]. It is a solution of the following equation

$$\frac{dx_0}{dt} = f_0(x_1, \ldots, x_n, u) \tag{6}$$

and initial condition $x_0(0) = 0$. Then the quantity J according to (9) becomes equal to the boundary of x(t) when $t \to \infty$. The system of differential equation, which are adjoint to the system (7) is composed with new variables $\Psi = \{\Psi_0, \Psi_1, \ldots, \Psi_n\}$:

$$\frac{d\Psi_0}{dt} = -\sum_{\alpha=0}^{n} \frac{\partial f_\alpha}{\partial x_0} \Psi_\alpha = 0 \tag{7}$$

$$\frac{d\Psi_i}{dt} = -\sum_{\alpha=0}^{n} \frac{\partial f_\alpha}{\partial x_0} \Psi_\alpha = -2q_i\theta_i - \sum_{j=1}^{n} a_{ij} \Psi_j \qquad (i = 1,\ldots,n) \tag{8}$$

After that the Hamilton function is composed:

$$H(\theta, \Psi, u) = \sum_{\alpha=0}^{n} \Psi_\alpha \frac{dx_\alpha}{dt} = \sum_{\alpha=0}^{n} \Psi_\alpha f_\alpha(x_1,\ldots x_n, u) = \Psi_0 \left(\sum_{i=1}^{n} q_i x_i^2 + ru^2 \right) + \sum_{i=1}^{n} \Psi_i \left(\sum_{j=1}^{n} a_{ij} x_j + b_i u \right) \tag{9}$$

In the right-hand side of Eq. (9) the quantity *u* is contained in the expression

$$H_1 = r\Psi_0(t)u^2(t) + u(t)\sum_{i=1}^{n} b_i\Psi_i(t) \tag{10}$$

Because of that the condition for maximum of H coincide with the condition

$$\begin{aligned}\max_{|u|\le 1} H_1 &= \max_{|u|\le 1}\left[r\Psi_0(t)u^2(t) + u(t)\sum_{i=1}^{n} b_i\Psi_i(t)\right] = \\ &= \max_{|u|\le 1}\left\{ r\Psi_0(t)\left[u(t) + \frac{1}{2r\Psi_0}\sum_{i=1}^{n} b_i\Psi_i(t)\right]^2 - \frac{1}{4r\Psi_0}\left[\sum_{i=1}^{n} b_i\Psi_i(t)\right]^2\right\}\end{aligned} \tag{11}$$

Having in mind condition (7) the quantity Ψ_0 is a constant. As its value can be any negative number it is set to $\Psi_0 = -1$.

After placing this value in Eq. (11) the maximum of the expression in the square brackets will be reached when the first negative addend becomes zero if it is possible or takes its minimal absolute value. The expression

$$\left[u(t) - \frac{1}{2r}\sum_{i=1}^{n} b_i\Psi_i(t)\right]^2 \tag{12}$$

will take its minimal absolute value if on condition $|u| \le 1$ a value of the following kind is chosen for u

$$u(t) = \begin{cases} \dfrac{1}{2r}\sum_{i=1}^{n} b_i\Psi_i & \text{at} \quad \left|\dfrac{1}{2r}\sum_{i=1}^{n} b_i\Psi_i\right| \le 1 \\ 1 & \text{at} \quad \left|\dfrac{1}{2r}\sum_{i=1}^{n} b_i\Psi_i\right| \ge 1 \\ -1 & \text{at} \quad \left|\dfrac{1}{2r}\sum_{i=1}^{n} b_i\Psi_i\right| \le -1 \end{cases} \tag{13}$$

The values of $\psi_c(t)$ can be determined if the adjoint equations (7), (8) are solved. This leads to the requirement the initial values of $\psi_0(0)$ to be found beforehand.

First u(t) is assumed not to reach its boundary values. Then after placing the upper expression from (13) instead of u(t) in Eqs. (1), (7) и (8) one obtains

$$\begin{aligned}\frac{dx_i}{dt} &= \sum_{j=1}^{n} a_{ij}x_j + \frac{b_i}{2r}\sum_{j=1}^{n} b_j\Psi_j \quad (i = 1,...,n) \\ \frac{d\Psi_i}{dt} &= 2q_i x_i - \sum_{j=1}^{n} a_{ji}\Psi_j\end{aligned} \tag{14}$$

This system of equations has to be solved with the initial conditions $x_1(0),\ldots, x_n(0)$ as well as with the final (boundary) conditions

$$\lim_{t\to\infty} x_1(t) = \lim_{t\to\infty} x_2(t) = ... = \lim_{t\to\infty} x_n(t) = 0 \tag{15}$$

It is necessary the appropriate initial conditions $\psi_1(0), \ldots, \psi_n(0)$ to be selected in such a way that the initial and the final conditions for $x_1(t),\ldots, x_n(t)$ to be satisfied.

The relationship between $x_i(0)$ and $\psi_i(0)$ has the following form [4]:

$$\Psi_i(0) = \sum_{j=1}^{n} \chi_{ij}x_i(0) \qquad (i = 1, \ldots, n) \tag{16}$$

These relationships have to be kept in any time, for which one can always assume to be the initial one. Therefore the optimal control *u* within the boundaries is determined and it has the following form:

$$u = \frac{1}{2r}\sum_{i=1}^{n} k_i x_i \qquad (17)$$

where $k_i = \sum_{j=1}^{n} b_j \chi_{ji}$

The expression (17) holds only in the cases when the absolute value of the sum $\frac{1}{2r}(k_1x_1 + ... + k_nx_n)$ is not greater than one. When $\left|\frac{1}{2r}(k_1x_1 + ... + k_nx_n)\right| > 1$ the optimal control passes on the boundary i.e. $|u| = 1$, if the right hand boundary conditions are satisfied i.e. the solution of the system (1), which became nonlinear in connection to the nonlinear relationship between u and $x_1,\ldots, x_n$, tends to zero. In other words the solution of the system has to be asymptotically stable. Thus the optimal control is defined by the expression

$$u(t) = \begin{cases} \frac{1}{2r}\sum_{i=1}^{n} k_i x_i & \text{at} \quad \left|\frac{1}{2r}\sum_{i=1}^{n} k_i x_i\right| \le 1 \\ 1 & \text{at} \quad \frac{1}{2r}\sum_{i=1}^{n} k_i x_i \ge 1 \\ -1 & \text{at} \quad \frac{1}{2r}\sum_{i=1}^{n} k_i x_i \le -1 \end{cases} \qquad (18)$$

Structure and Training of the Neural Network

For the control function realization a feed forward neural network with one hidden layer is used. Thus the necessity of solving a large number of equations for determining the coefficients k_i drops off.

The neural network consists of three layers – an input, output and hidden one. The input and hidden layers have five neurons and the output layer – one. The activation function of the output neuron is piecewise linear. The neural network output is

$$y = \begin{cases} +1 & \varphi(\nu) \ge 1 \\ \varphi(\nu) & |\varphi(\nu)| \le 1 \\ -1 & \varphi(\nu) \le 1 \end{cases} \qquad (19)$$

where $\nu = w^T z$. The neural network input is denoted z and w is the neural network weight. The neural network output represents the control u, x – the state vector and weights are the coefficient k.

The neural network is trained according to the back-propagation algorithm. Let the training sample $\{z(n), d(n)\}_{n=1}^{N}$ be given where z(n) are the system states and d(n) is the corresponding control, which are known preliminarily. The neural network is trained according to the back-propagation algorithm [Haykin 1999].

Simulation Results

In order to verify the suggested approach for solving the optimal control problem following system is considered:

$$\frac{dx_1}{dt} = x_2$$

$$\frac{dx_2}{dt} = -x_1 - 2x_2 + u$$

and the control is constrained by

$$|u| \le 1$$

The performance index to be minimized is of the form:

$$\int_0^{\infty} [x_1^2(t) + x_2^2(t) + u^2(t)dt$$

The problem is solved by using Pontryagins principle and neural networks. The results, which are obtained by both approaches, are compared. In Fig. 1 the optimal control, obtained by using neural networks is shown. Fig. 2 depicts the corresponding states trajectory. In Fig. 3 and Fig. 4 the optimal control, obtained by applying the maximum principle and the corresponding trajectory are given respectively. By 1 and 2 are denoted x_1 and x_2 respectively.

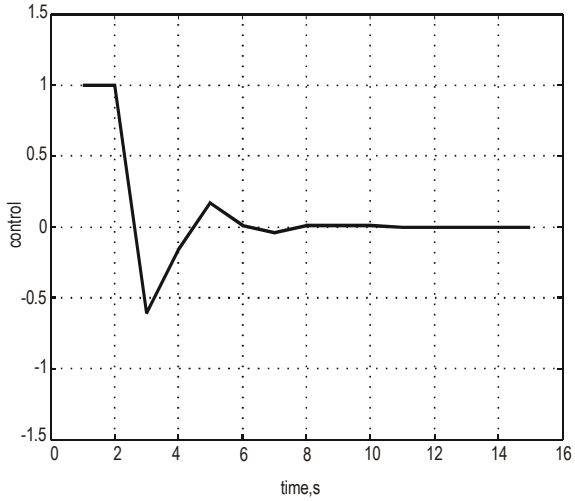


Fig. 1. Optimal control, obtained by using the suggested neural network based approach

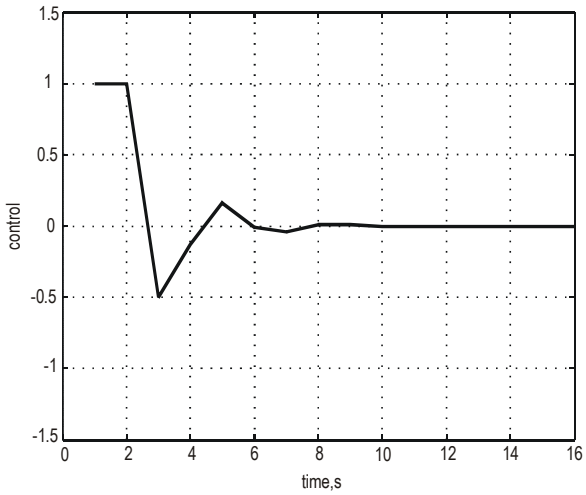


Fig. 2. Optimal control, obtained by applying the maximum principle

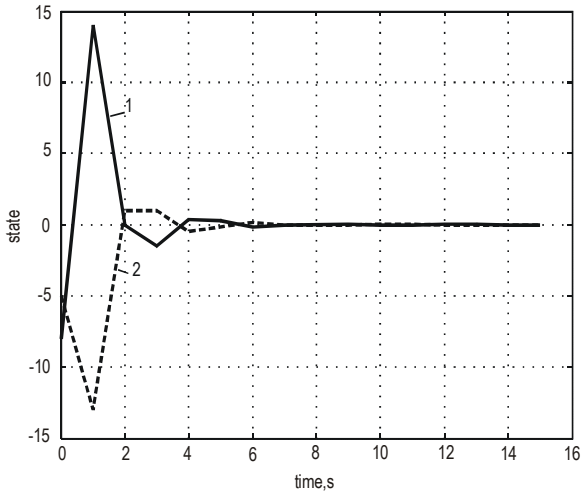


Fig.3 Optimal trajectory (neural network based approach)

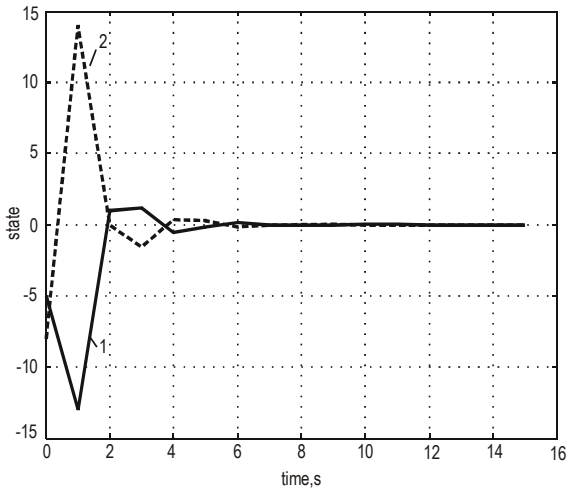


Fig.4 Optimal trajectory (Pontryagin's maximum principle)

Conclusion

In the present paper an approach for optimal constrained control based on using of neural networks is suggested.

On the basis of the simulation experiments one can say that the proposed approach for optimal control is accurate enough for the engineering practice. The suggested approach can be applied for optimal control in real time, where the control is constrained.

Bibliography

[Ahmed 1998] M.S. Ahmed and M.A. Al-Djani. Neural regulator design. Neural Networks, Vol. 11, pp. 1695-1709, 1998

[Chjan 1961] Zh.-V.Chjan. A problem of optimal system synthesis through the maximum principle. Automatization and telemechanics, Vol. XXII, No.1, 1961, pp. 8-12

[Haykin 1999] S. Haykin. Neural Networks: A Comprehensive Foundation., 1999, 2nd ed, Macmillan College Publishing Company, New York

[Lewis 2002] F.L. Lewis and M. Abu-Khalaf. A Hamilton-Jacobi setup for constrained neural network control. Proceedings of the 2003 IEEE International Symposium on Intelligent Control Houston, Texas *October 5-8. 2003

[Pontryagin 1983] L.S. Pontryagin, V.G. Boltyanskiy, R.V. Gamkrelidze and E.F.Mishtenko. Methematical theory of the optimal processes. Nauka, Moskow, 1983, in Russian

Authors' Information

Georgi Toshkov –e-mail: g_toshkov2006@abv.bg

Daniela Toshkova – e-mail: daniela_toshkova@abv.bg

Technical University of Varna, 1, Studentska Str, Varna, 9010, Bulgaria

Todorka Kovacheva – Economical University of Varna, Kniaz Boris Str, e-mail: todorka_kovacheva@yahoo.com

LINEAR CLASSIFIERS BASED ON BINARY DISTRIBUTED REPRESENTATIONS

Dmitri Rachkovskij

Abstract: Binary distributed representations of vector data (numerical, textual, visual) are investigated in classification tasks. A comparative analysis of results for various methods and tasks using artificial and real-world data is given.

Keywords: Distributed representations, binary representations, coarse coding, classifiers, perceptron, SVM, RSC

ACM Classification Keywords: C.1.3 Other Architecture Styles - Neural nets, I.2.6 Learning - Connectionism and neural nets, Induction, Parameter learning

Introduction

Classification tasks consist in assigning input data samples to one or more classes from a predefined set [1]. Classification in the inductive approach is realized on the basis of a training set containing labeled data samples. Usually, input data samples are represented as numeric vectors. Vector elements are real numbers (e.g., some measurements of object characteristics or their function) or binary values (indicators of some features in the input data).

This vector information often isn't explicitly relevant to the classification, therefore some kind of transformation is necessary. We have developed methods for transformation of input information of various kinds (such as numerical [2], textual [3], visual [4]) to binary distributed representations. These representations can then be

classified by linear classifiers – such as *SVM* [5] or more computationally effective and naturally handling multiple classes perceptron-like classifiers [4, 6]. The objective of this paper is to investigate efficiency of the proposed methods for distributed information representation and classification using real and artificial data of different modalities.

Numeric vector data classification

For an experimental research of the abovementioned methods on numeric data the following well-known test problems have been selected: Leonard-Kramer *LK*, *XOR*, *Double Spiral*; datasets generated by *DataGen* [6]; and sample data from the *Elena* database [7]. The dimensionality A of data vectors varied from 2 to 36, number of classes C varied from 2 to 11, and the number of samples in the training and test sets varied from 75 to 3218.

All selected problems have essentially non-linear class boundaries. Therefore, non-linear transformation of input numeric vectors has been used - i.e., *RSC* and *Prager* [2] methods of encoding. Those methods extract binary features – indicators of input A-dimensional vector presence in s-dimensional ($s<A$) hyperrectangle receptive fields with random position and size.

To investigate the impact of code parameters on the classification quality, we chose the following experimental scheme. Input vectors were converted to *RSC* and *Prager* codes. Those codes were then used as input data for training and testing linear classifiers. The number (or percent) of test errors was chosen as a classification quality criterion. We used *SVM* [5] and modifications of perceptron-like classifiers [4] as linear classifiers for the obtained distributed representations. Besides, classification experiments with (non-linear) kernel *SVM* using *Prager*, *RSC* [2] and standard (Gaussian and polynomial) kernels were conducted.

It is well known [5] that *SVM* doesn't support online learning, requires solving computationally expensive non-linear programming problems, and constructs optimal separating hyperplane for two-class problems only. In this work we have also used a perceptron with an enlarged margin and multi-class learning rule developed by I.Misuno in order to overcome *SVM* drawbacks. In the resulting perceptron outputs of neurons that correspond to classes are determined as $y_c = \Sigma_i x_i w_{ic}$, where w_{ic} are the weights of modifiable connections, x_i is the i-th element value of the vector input to the connections. (In the present context $\mathbf{x}$ is the binary vector obtained by input transformation to distributed representation, but the original data vector could be used for linear tasks as well). For the "true" class neuron $y_{c\text{-}true} = y_{c\text{-}true}(1-T)$, where $0<T<1$ is the "defense margin parameter". The classification output is the index c^* of neuron with the maximum activation: $c^* = \text{argmax}_c\ y_c$. In case of an error ($c^* \neq c_{true}$) connections are modified in the following way: $w_{ic} = w_{ic} + \Delta w$ for $c=c_{true}$ and $w_{ic} = w_{ic} - f(\Delta w)$ for c: $y_c > y_{c\text{-}true}$, where c_{true} is the index of the correct class. E.g., $f(\Delta w) = \Delta w/|c|$. Our previous version of the enlarged margin perceptron had single-class (not multi-class) learning rule: unlearning with single class $c^* = \text{argmax}_c\ y_c$ was performed in case of error, and $f(\Delta w) = \Delta w$. For T=0 and single-class learning rule one obtains usual percepton, while for T=0 and multi-class learning rule one obtains usual percepton with multi-class learning. Multi-class learning extracts and uses more information from a single error and so provides a potential for faster learning and better generalization for essentially multi-class tasks, especially at early learning iterations of the training set. This can be critical for the on-line learning tasks.

Experimental results for numerical data

Figure 1 demonstrates Leonard-Kramer problem results: dependencies of classification errors percent %*err*, elementary cell size *cell* (the smaller is the *cell,* the larger is resolution), and average fields dimensionality $E\{s\}$ vs the code density p (the fraction of 1s in the code). For *Prager* and *RSC* coding, the results of *SVM* and of the perceptron with single-class ("Perc0") and multi-class learning ("Perc1") with no margin ("T0") and enlarged margin ("T0.75") were averaged by 10 realizations of codes at N=100. Results for *SVM* with kernels (*Kernel*) are also shown. For all cases (as well as for large Ns Figure 2) classification error reaches its minimum near p=0.25, which corresponds to the minimum *cell* and $E\{s\} = 2$.

Figure 2 demonstrates %*error* and *cell* vs N at p=0.25. The results were averaged by 10 realizations of code generation trials. At N=500 the *SVM* results have already been close to the kernel results. For the enlarged margin perceptron (T=0.75) with multi-class learning the error for N>(300–1000) was lower than the *SVM* one. Training for perceptron was faster than that for *SVM* by 20 times, while testing was >100 times faster.

The experimental results for the *DataGen* data are given in Figure 3 (*A*=4, *S*=3, *C*=4, *R*=4, where *R* determines the complexity of the class regions [6]) and the number of samples per class is equal to 100. Averaging was conducted through 5 realizations of the *DataGen* samples and 5 realizations of codes. For these parameters the minimum cell value corresponds to *p*~0.3 (and close to it for *p*=0.125...0.5) and the error minimum for both *SVM* and the enlarged margin perceptron is also reached in this interval. For *N*=100 it is biased to the larger *p* values (which ensures a more stable number of 1s). For *N*=1000 the minimum is biased to the smaller *p* which corresponds to a larger mean dimensionality of receptive fields, while the number of 1s remains large enough and the *cell* is small enough. The training time for the perceptron is ~20 times less than for *SVM*, and the testing time is ~500 times less.

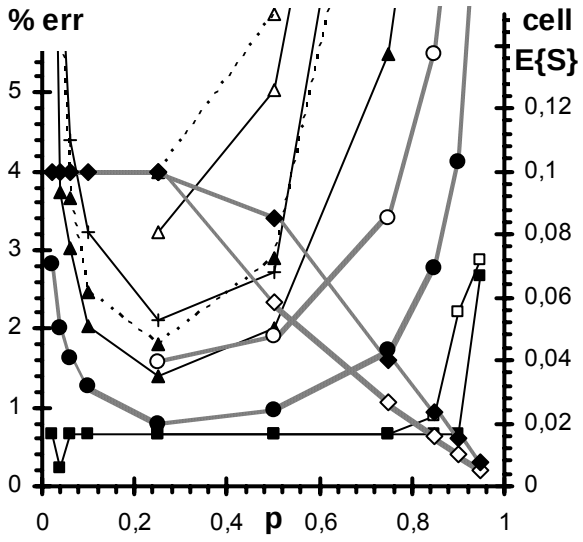


Figure 1

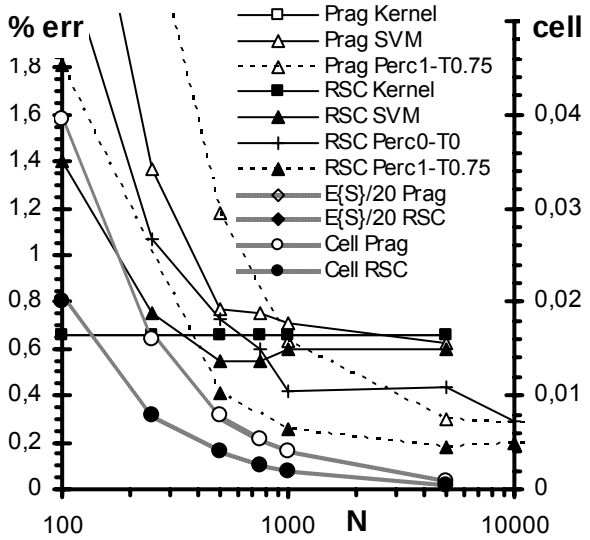


Figure 2

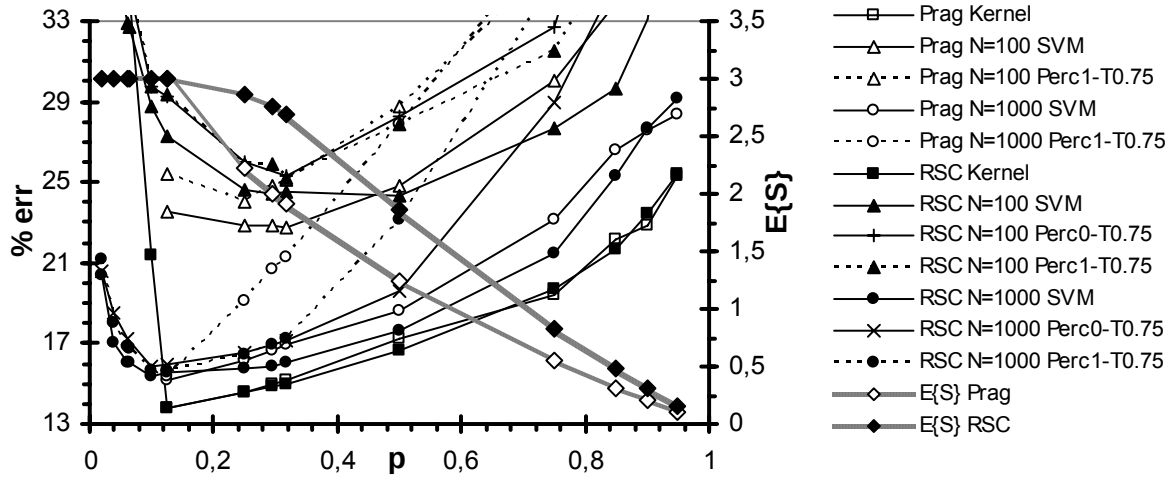


Figure 3

We have also obtained and compared experimental results for the multi-class and single class learning perceptrons. The error rate for multi-class learning perceptron was up to 1.5 times lower than for the usual one, whereas the error rate for multi-class perceptron with the enlarged margin was still lower and comparable with the error for single-class perceptron with the enlarged margin. Typically, the learning curves (test error vs training iteration number) were lower for multi-class learning than for single-class learning, and best results for multi-class

learning were higher than those for single-class learning. The results for ordinary perceptron (no margin and single-class learning) were typically lower than those for perceptrons with the enlarged margin in all tests.

Table 1

Database	RSC SVM	RSC kern.	RSC Perc0 T0	RSC Perc1 T0.75	Prager SVM	Prager kern.	kNN	MLP	IRVQ
Clouds	12.68	14.84	17.84	–	12.4	14.8	11.8	12.2	11.7
Concentric	1.36	1.2	1.58	–	1.17	1.04	1.7	2.8	1.5
Gaussian2 S=2	28.12	35.12	38.42	–	27.83	35.64	27.4	26.8	27.2
Gaussian7 S=2	14.35	15.68	20.49	–	14.36	15.76	15.9	15.3	11.5
Gaussian7 S=5	14.69	14.64	19.62	–	13.36	15.12	–	–	–
Iris S=2	6.53	6.67	6.13	5.33	5.59	6.67	4	4.3	6.7
Iris S=4	4.27	6.67	7.47	5.73	6.13	6.67	–	–	–
Phoneme S=2	14.12	11.51	16.43	13.7	15.79	14.47	12.3	16.3	16.4
Phoneme S=5	13.61	11.62	15.74	13.19	14.82	12.62	–	–	–
Satimage S=2	10.06	10.13	10.69	9.15	10.82	10.79	9.9	12.3	11.4
Satimage S=5	10.11	–	10.89	9.1	10.64	–	–	–	–
Texture S=2	0.82	0.76	1.44	1.13	0.82	0.80	1.9	2.0	3.1
Texture S=5	0.73	–	1.65	1.07	0.74	–	–	–	–

For the artificial data of the *Elena* database the code parameters were N=1000, A=S=2, p=0.25; for the real data (*Iris*, *Phoneme*, *Satimage*, *Texture*) N=10000, S=2,5(4), p=0.1 and 0.25. Table 1 demonstrates percentage of classification errors. For *SVM* and perceptron the results were obtained by averaging over 10 realizations of *RSC* and *Prager* codes. The best results of the known methods *kNN*, *MLP*, *IRVQ* are also given [7]. The comparison of results shows that *RSC* and *Prager* coding provided the best result to *Concentric*, *Phoneme*, *Texture* and the second best result for *Satimage* and *Gaussian 7D*. Perceptron training time is (on the average) several times less, and test time is dozens of times less than that for SVM.

Classification of texts and images

Traditional approaches to text classification use functions of word occurrence frequencies as elements of their vector representations. Methods for informative feature selection can be used to reduce vectors' dimensionality, [4]; however, even simplified methods that consider features as independent have quadratic computational complexity. We propose and investigate the use of distributed representations for dimensionality reduction of vector text representation. N-dimensional binary code with m 1s in random positions is used to represent each word. N-dimensional text representation is formed by adding of its word vectors, with the following mapping to binary space performed by a threshold operation, or by context-dependent thinning *CDT* (see [3]).

Testing in the classification task has been conducted using *Reuters*-21578 text collection [3] by means of *SVM*. For the *TOP*-10 categories *BEP* (break even point of recall/precision characteristic) for the initial vector representation of N^*=20000 was 0.920/0.863 (micro/macro averaging). Using of the distributed representations with N=1000, m=2 made it possible to obtain 0.861/0.775 (micro/macro averaging), and usage of *CDT* in some experiments increased it by several percents.

The analogously formed distributed representations were studied for classification of handwritten digit images of the *MNIST* database [4], where images were coded by the extracting binary features. The presence of each feature corresponded to the combination of white and black points in some positions of retina (*LIRA* features [4]). As a result, a "primary" binary code was obtained. Then it was transformed to the "secondary" representation using the same procedures as for text information.

Classification results with dimensionality reduction from N^* to N are shown in Table 2. Line "sel" contains the error percent obtained using selection of informative features [4]. Line "distr" contains classification results for the "secondary" binary distributed representations. Results for the distributed representations considerably exceed the results of initial representations for the same N and are similar to the results of feature selection methods [4].

We have also obtained and compared *MNIST* experimental results for multi-class and single class learning perceptrons. Here we used original *LIRA* features without transformation to secondary distributed representations, for N={1000, 10000, 50000}, and both with and without feature selection. We observed the same tendencies as for numerical data, however the advantage of multi-class learning was more pronounced for weaker classifiers (at N=1000) than for the better ones (at N=50000).

Table 2

N (err)	5000(667)	10000 (407)		50000 (195)			128000 (160)		
*N**	1000	1000	5000	1000	5000	10000	1000	5000	10000
sel	820	578	420	492	264	242	474	261	218
distr	904	727	**415**	632	274	**213**	826	264	**204**

Conclusions

The developed binary distributed representations of vector data (numeric, text, images) were investigated in the classification tasks. A comparative analysis of various method results for the tasks with artificial and real data was carried out. The study showed that analytical expressions for the characteristics of the *RSC-Prager* codes of the numerical vectors obtained in [2] make it possible to select code parameters that provide high results in the non-linear classification tasks using linear classifiers. Results obtained with the proposed perceptron with an enlarged margin are comparable to the results of the state-of-the-art *SVM* classifiers, however a significant decrease in training and recognition time has been observed. The results obtained with the *RSC-Prager* kernels also make it possible to reduce training and testing time for small S.

Application of distributed encoding for representation of binary features in texts and images also made it possible to obtain computationally effective solutions of classification tasks preserving classification quality. A promising direction of further studies could consist in developing computationally efficient *RSC* and *Prager* kernels, as well as developing distributed representations and kernels that provide a more adequate account for structural information in the input data.

Acknowledgements

Author expresses his gratitude to Ivan Misuno and Serge Slipchenko for the support and collaboration during preparation of this paper.

Bibliography

[1] R. Duda, P. Hart, D. Stork. Pattern Classification, 2nd ed. – New York: John Wiley & Sons, 2000.

[2] S.V. Slipchenko, I.S. Misuno, D.A. Rachkovskij. Properties of coarse coding with random hyperrectangle receptive fields. Mathematical machines and systems, N 4, pp. 15-29, 2005 (in Russian).

[3] I.S. Misuno. Distributed vector representation and classification of texts. USIM, N 1, pp. 85-91, 2006 (in Russian).

[4] I.S. Misuno, D.A. Rachkovskij, S.V. Slipchenko. The experimental research of handwritten digits classification. System technologies, N 4 (39), pp. 110–133, 2005 (in Russian).

[5] V.N. Vapnik. Statistical Learning Theory. – New York: John Wiley & Sons, 1998.

[6] I.S. Misuno, D.A. Rachkovskij, E.G. Revunova, S.V. Slipchenko, A.M. Sokolov, A.E. Teteryuk. Modular software neurocomputer SNC - implementation and applications. USiM, N 2, pp. 74–85, 2005 (in Russian).

[7] D. Zhora. Evaluating Performance of Random Subspace Classifier on ELENA Classification Database. Artificial Neural Networks: Biological Inspirations – ICANN 2005 – Springer–Verlag Berlin Heidelberg, pp. 343–349, 2005.

Authors' Information

Dmitri A. Rachkovskij – International Research and Training Center of Information Technologies and Systems; Pr. Acad. Glushkova, 40, Kiev, 03680, Ukraine; e-mail: dar@infrm.kiev.ua.

GROWING NEURAL NETWORKS USING NONCONVENTIONAL ACTIVATION FUNCTIONS

Yevgeniy Bodyanskiy, Iryna Pliss, Oleksandr Slipchenko

Abstract: In the paper, an ontogenic artificial neural network (ANNs) is proposed. The network uses orthogonal activation functions that allow significant reducing of computational complexity. Another advantage is numerical stability, because the system of activation functions is linearly independent by definition. A learning procedure for proposed ANN with guaranteed convergence to the global minimum of error function in the parameter space is developed. An algorithm for structure network structure adaptation is proposed. The algorithm allows adding or deleting a node in real-time without retraining of the network. Simulation results confirm the efficiency of the proposed approach.

Keywords: ontogenic artificial neural network, orthogonal activation functions, time-series forecasting.

ACM Classification Keywords: I.2.6 Learning – Connectionism and neural nets

Introduction

Artificial neural networks (ANNs) are widely applied to solving a variety of problems such as information processing, data analysis, system identification, control etc. under structural and parametric uncertainty [1, 2].

One of the most attractive properties of ANNs is the possibility to adapt their behavior to the changing characteristics of the modeled system. By adaptivity we understand not only the adjustment of parameters (synaptic weights), but also the possibility to adjust the architecture (the number of nodes). The goal of the present paper is the development of an algorithm for structural and synaptic adaptation of ANNs for nonlinear system modeling, capable of online operation, i.e. sequential information processing without re-training after structure modification.

The problem of optimization of neural network architecture has been studied for quite a long time. The algorithms that start their operation with simple architecture and gradually add new nodes during learning, are called 'constructive algorithms'. In contrast, destructive algorithms start their operation with an initially redundant network, and simplify it as learning proceeds. This process is called 'pruning'.

Radial basis function network (RBFN) is one of the most popular neural network architectures [3]. One of the first constructive algorithms for such networks was proposed by Platt and named 'resource allocation' [4]. By present time, a number of modifications of this procedure is known [5, 6]. One of the most known is the cascade-correlation architecture developed by Fahlman and Lebiere [7].

Among the destructive algorithms, the most popular are the 'optimal brain damage' [8] and 'optimal brain surgeon' [9]. In these methods, the significance of a node or a connection between nodes is determined by the change in error function that its deletion incurs. For this purpose, the matrix of second derivatives of the optimized function with respect to the tunable parameters is analyzed. Both procedures are quite complex computationally. Besides that, an essential disadvantage is the need for re-training after the deletion of non-significant nodes. This, in turn, makes the real-time operation of these algorithms impossible. Other algorithms such as [10] are heuristic and lack universality.

It should be noted that there is no universal and convenient algorithm, which could be used for the manipulation of the number of nodes and suitable for most problems and architectures. Many of the algorithms proposed so far lack theoretical justification as well as the predictability of the results of their application and the ability to operate in real time.

Network Architecture

Let's consider the network architecture, that implements the following nonlinear mapping

$$\hat{y}(k) = \hat{f}(x(k)) = \sum_{i=1}^{n} \sum_{j=1}^{h_i} w_{ji} \phi_{ji}(x_i(k)) \tag{4}$$

where $k = 1, 2, \ldots$ – discrete time or ordinal number of sample in training set, w_{ji} – tunable synaptic weights, $\phi_{ji}(\bullet)$ – j-th activation function for i-th input variable, h_i – number of activation functions for appropriate input variable, $x_i(k)$ – value of i-th input signal at time moment k (or for k-th training sample).

This architecture contains $h = \sum_{i=1}^{n} h_i$ tunable parameters and it can be readily seen that the this number is between the scatter-partitioned and grid-partitioned systems.

We propose the use of orthogonal polynomials of one variable for the activation functions. Particular system of functions can be chosen according to the specificity of the solved problem. If the input data are normalized on the hypercube $[-1, 1]^n$, the system of Legendre polynomials orthogonal on the interval $[-1, 1]$ with weight $\gamma(x) \equiv 1$ [17] can be used:

$$P_n(x) = 2^{-n} \sum_{m=0}^{[n/2]} (-1)^m \frac{(2n-2m)!}{m!(n-m)!(n-2m)!} x^{n-2m}, \tag{5}$$

where $[\bullet]$ is the integer part of a number.

System of Legendre polynomials is best suited for the case when we know exact interval of data changes before network construction. This is quite a common situation as well as an opposite one. For the latter case the following system of Hermite orthogonal polynomials can be used:

$$H_n(x) = n! \sum_{m=0}^{[n/2]} (-1)^m \frac{(2x)^{n-2m}}{m!(n-2m)!}. \tag{6}$$

This system is orthogonal on $(-\infty; +\infty)$ with weight function $h(x) = e^{-x^2}$ and gives us a possibility to decrease influence of the data lying far from the point of origin.

Normalized Hermite polynomials usually denoted by $\hat{H}_n(\bullet)$ (i.e. those with $\left\| \hat{H}_n(\bullet) \right\| = 1$) can be obtained from (6):

$$\hat{H}_n(x) = \sqrt{\frac{n!}{2^n \sqrt{\pi}}} \sum_{m=0}^{[n/2]} (-1)^m \frac{(2x)^{n-2m}}{m!(n-2m)!}. \tag{7}$$

Among other possible choices for activation functions we should mention Chebyshev [15, 16] and Hermite [18] functions as well as non-sinusoidal orthogonal systems proposed by Haar and Walsh.

Synaptic Adaptation

The sum of squared errors will be used as the learning criterion:

$$E(k) = \sum_{p=1}^{k} e^2(p) = \sum_{p=1}^{k} (y(p)) - \sum_{i=1}^{n} \sum_{j=1}^{h_i} w_{ji} \phi_{ji}(x_i(p)))^2, \tag{8}$$

where k is the ordinal number of an element in the learning sequence or the discrete time when the data is processed in the order of its arrival, $y(p)$ – value of learning signal at time moment p (or for p-th training sample).

For the convenience of further notation, let us re-write the expression for the output of the neural network (4) in the form

$$\hat{y}(k+1) = \phi^T(k+1)W(k), \tag{9}$$

where $\phi(k) = (\phi_{11}(x(k)), \phi_{21}(x(k)), ..., \phi_{h_n n}(x(k)))^T$ is a $(h \times 1)$ vector of the values of the basis functions for the *k*-th element of the training set (or at the instant *k* for sequential processing), $W(k) = (w_{11}(k), w_{21}(k), ..., w_{h_n n}(k))^T$ is a $(h \times 1)$ vector of synaptic weights estimates at the iteration k.

Since the output of the proposed neural network linearly depends on the tuned parameters, we can use the least squares procedure to estimate them. For sequential processing, e.g. in the case of online learning, we can use the recursive least squares method:

$$\begin{cases} W(k+1) = W(k) + \dfrac{P(k)(y(k+1) - W^T(k)\phi(k+1))\phi(k+1)}{1 + \phi^T(k+1)P(k)\phi(k+1)}, \\ P(k+1) = P(k) - \dfrac{P(k)\phi(k+1)\phi^T(k+1)P(k)}{1 + \phi^T(k+1)P(k)\phi(k+1)}. \end{cases} \tag{10}$$

Because of the orthogonality of the basis functions, the matrix $P(k)$ will tend to diagonal form as $k \to \infty$. If the activation functions are orthonormal, $P(k)$ will tend to the unity matrix. Due to this property, the learning procedure will retain numerical stability with the increase of the number of samples in the training sequence.

Structure Adaptation

Let's consider sequential learning that minimizes (8) and leads to the estimate

$$W_h(k) = R_h^{-1}(k)F_h(k), \tag{11}$$

$$R_h^{-1}(k) = R_h^{-1}(k-1) - \frac{R_h^{-1}(k-1)\phi(k)\phi(k)^T R_h^{-1}(k-1)}{1 + \phi(k)^T R_h^{-1}(k-1)\phi(k)}, \tag{12}$$

$$F_h(k) = F_h(k-1) + \phi(k)y(k). \tag{13}$$

The use of the recursive least squares (RLS) method and its modifications allows to obtain an accurate and well-interpretable measure of significance of each function in the mapping (4). This mapping can be considered as an expansion of an unknown reconstructed function in the basis $\{\phi_{ji}(.)\}$. Obviously, if the absolute value of any of the coefficients in this expansion is small, then the corresponding function can be excluded from the basis without significant loss of accuracy. The remaining synaptic weights does not need to be retrained if the weight of the excluded node is close to zero. Otherwise, the network should be retrained.

Assume that a vector of synaptic weights $W_h(k)$ of a network comprising h nodes was obtained at the instant k using the formula (11), where the index h determines the number of basis functions (the dimension of $\varphi(k)$). Also assume that the absolute value of the considered parameter $w_h(k)$ is small, and we want to exclude corresponding unit function from the expansion (4). The assumption about the insignificance of the activation h is not restrictive, because we always can re-number the basis functions. This will result only in the rearrangement of the rows and columns in the matrix $R_h(k)$ and in the change of ordering of the elements of the vector $F_h(k)$. However, the rearrangement of columns and/or rows of a matrix does not influence the subsequent matrix operations.

Taking into account the fact that the matrix $R_h(k)$ is symmetric, we obtain:

$$W_h(k) = R_h^{-1}(k)F_h(k) = \begin{pmatrix} R_{h-1}(k) & \beta_{h-1}(k) \\ \beta_{h-1}^T(k) & r_{hh}(k) \end{pmatrix}^{-1} \begin{pmatrix} F_{h-1}(k) \\ f_h(k) \end{pmatrix}, \tag{14}$$

where $r_{ij}(k)$ is the element of the i-th row and j-th column of the matrix $R_h(k)$,

$\beta_{h-1}(k) = (r_{1h}(k),...,r_{h-1h}(k))^T = (r_{h1}(k),...,r_{hh-1}(k))^T$, $f_i(k)$ is the i-th element of vector $F_h(k)$.

After simple transformations of (14) we obtain the expression

$$W_h(k) = \begin{pmatrix} W_{h-1}(k) - R_{h-1}^{-1}(k)\beta_{h-1}(k)w_h(k) \\ w_h(k) \end{pmatrix} \tag{15}$$

that enables us to exclude the function from (4) and obtain the corrected estimates of the remaining parameters of the ANN. For this operation, we use only the information accumulated in the matrix $R_h(k)$ and vector $F_h(k)$.

Using the same technique as above, it is possible to write a procedure that can be used to add a new function to the existing basis. Direct application of the Frobenius formula [12] leads to the algorithm

$$W_{h+1}(k) = R_{h+1}^{-1}(k)F_{h+1}(k) = \begin{pmatrix} R_h(k) & \beta_h(k) \\ \beta_h^T(k) & r_{h+1,h+1}(k) \end{pmatrix}^{-1} \begin{pmatrix} F_h(k) \\ f_{h+1}(k) \end{pmatrix} =$$

$$\begin{pmatrix} W_h(k) + R_h^{-1}(k)\beta_h(k)\dfrac{\beta_h^T(k)W_h(k) - f_{h+1}(k)}{r_{h+1h+1}(k) - \beta_h^T(k)R_h^{-1}(k)\beta_h(k)} \\ \dfrac{-\beta^T(k)W_h(k) + f_{h+1}(k)}{r_{h+1,h+1}(k) - \beta_h^T(k)R_h^{-1}(k)\beta_h(k)} \end{pmatrix} \tag{16}$$

where $\beta_h(k) = (r_{1h+1}(k),...,r_{hh+1}(k))^T = (r_{h+11}(k),...,r_{h+1h}(k))^T$.

Thus, with the help of equation (16) we can add a new function (neuron) to the model (4), and exclude an existing function using the formula (15) without retraining remaining weights. In order to perform these operations in real time, it is necessary to accumulate the information about a larger number of basis functions than currently being used. E.g., we can initially introduce a redundant number of basis functions H and accumulate information in the matrix $R_H(k)$ and vector $F_H(k)$ as new data arrive, with only $h < H$ basis functions being used for the description of the unknown mapping. The complexity of the model can be either reduced or increased as required.

The analysis of equations (11), (15), and (16) shows that the efficiency of the proposed learning algorithm is directly related to the condition number of the matrix $R_h(k)$. This matrix will be non-singular if the functions $\{\varphi_i(.)\}_{i=1}^h$ used in the expansion (4) are linear-independent. The best situation is when the function system $\{\varphi_i(.)\}_{i=1}^h$ is orthogonal. In this case, the matrix $R_h(k)$ becomes diagonal, the formulas (11), (15), and (16) being greatly simplified because

$$diag(a_1,..,a_n)^{-1} = diag\left(\frac{1}{a_1},...,\frac{1}{a_n}\right), \tag{17}$$

where $diag(a_1,..,a_n)$ is an $(n \times n)$ matrix with non-zero elements $a_1,..,a_n$ only on the main diagonal.

Simulation Results

We have applied the proposed ontogenic network with orthogonal activation functions to online identification of a rat's (*Ratus Norvegius Vistar*) brain activity during sleeping phase.

The signal was measured with frequency of 64 Hz. We took a fragment of signal containing 3200 points (50 second of measuring), that was typical for sleeping phase of rat's life activity. Two neural networks of type (4) were trained in real-time. Each network had 10 inputs – delayed signal values ($y(k)$, $y(k-1)$,..., $y(k-9)$) and was trained to output one-step ahead value of the process – $y(k+1)$. First network utilized synaptic adaptation algorithm (11) while second one also involved the structure adaptation technique (15), (16). Initially both ANNs had 5 activation functions per input, the one with synaptic adaptation only retained all 50 tunable

parameters during it's work while ANN with structure adaptation mechanism had only 25 fired functions (the most significant ones chosen in real-time). For the results comparing purpose we also trained multilayer perceptron (further referred as MLP) with the same structure of inputs and training signal, having 5 units in the 1st and 4 in the 2nd hidden layers (that totals to 74 tunable parameters). As MLP is not capable of real-time data processing, all samples are used as training set and test criteria are calculated on the same data points. MLP was trained during 250 epochs with Levenberg-Marquardt algorithm. Our research showed that this is enough to achieve precision comparable to proposed ontogenic neural network with orthogonal activation functions.

For visual presentation of processed data see Fig. 1 which shows the results of identification using proposed neural network together with original time series.

Results of identification can be found in table 1. We used some different measures of identification quality. First, we analyse normalized root mean squared error, which is closely related to the learning criterion. Two other criteria used: "Wegstrecke" [19] characterizes the quality of the model for prediction/identification (+1 means perfect one), "Trefferquote" [20] is percent value of correctly predicted direction changes.

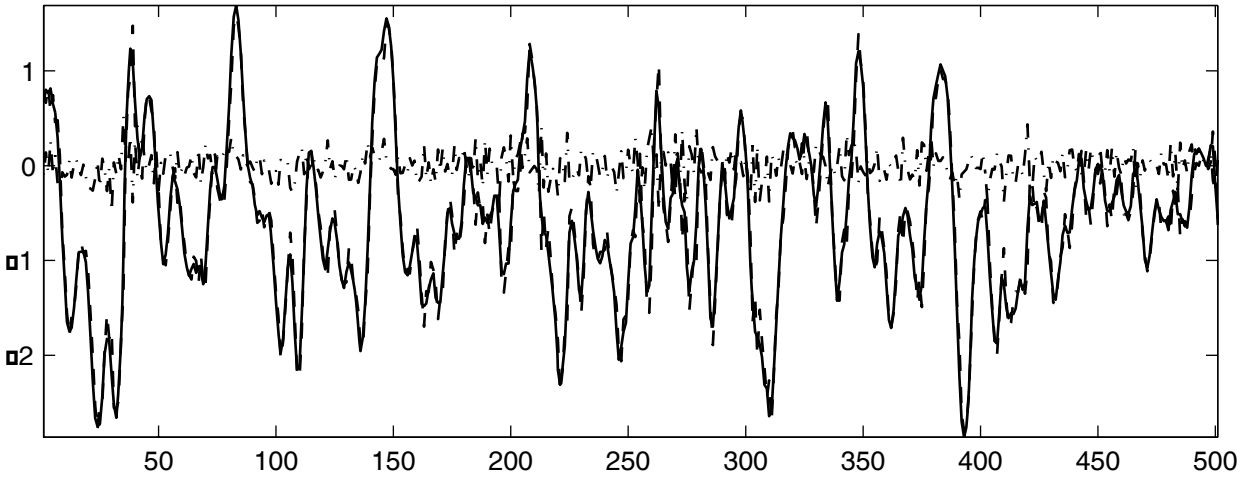


Figure 1. Identification of a rat's brain activity during sleeping phase using proposed neural network with orthogonal activation functions – brain activity signal (*solid line*), network output (*dashed line*), and identification error (*dash-dot line*)

Table 1 – Identification results for different architectures

Decription	*NRMSE*	*Trefferquote*	*Wegstrecke*
OrthoNN, real-time processing	*0.1852*	*82.2847*	*0.85312*
OrthoNN, real-time processing, variable number of nodes	*0.2175*	*77.6357*	*0.74625*
MLP, offline learning (250 epochs), error on the training set	*0.1685*	*83.9533*	*0.87192*

We can see that utilizing structure adaptation technique leads to somewhat worth results. This is the tradeoff for having less tunable parameters and possibility to process non-stationary signals.

Adaptation of neural network in real time benefits us in a number of ways. First, as noted earlier, it can reduce computational complexity. Second and perhaps more important benefit is in using adapting neuromodel as a basis for some higher level system of data processing (e.g. time-series classification, diagnostics system etc.).

After obtaining promising results of online identification we used proposed neural network architecture to monitor rat's state in real time. Second level of monitoring system was built with the help of expert which initially provided recorded activity of rat's brain together with animal state for each moment of time. We processed the data and analyzed neural network's set of states. The analysis showed that states are dividable in a space of synaptic weights. A slightly modified Bayes estimator for the state of observed object was synthesized and trained. Simulation showed that developed automated monitoring system is capable of online data processing and gives correct state in 94,5% of cases. The response of the systems in form of object's state was later verified by the expert and found reliable enough to be used for data preprocessing in day to day activity.

Conclusion

A new computationally efficient neural network with orthogonal activation functions was proposed. It has a simple and compact architecture not affected by the curse of dimensionality, and provides high precision of nonlinear dynamic system identification. An apparent advantage is much easier implementation and lower computational load as compared to the conventional neural network architectures.

The approach presented in the paper can be used for nonlinear system modeling, control, and time series prediction. An interesting direction of further work is the use of the network with orthogonal activation functions as a part of hybrid multilayer architecture. Another possible application of proposed ontogenic neural network is its use as a basis for diagnostic systems.

References

1. Handbook of Neural Computation. IOP Publishing and Oxford University Press, 1997.
2. Nelles O. Nonlinear System Identification. Berlin, Springer, 2001.
3. Poggio T. and Girosi F. A Theory of Networks for Approximation and Learning. A.I. Memo No. 1140, Artificial Intelligence Laboratory, Massachusetts Institute of Technology, 1989.
4. Platt J. A resource allocating network for function interpolation. Neural Computation, 3, 1991, p. 213-225.
5. Nag A. and Ghosh J. Flexible resource allocating network for noisy data. In: Proc. SPIE Conf. on Applications and Science of Computational Intelligence, SPIE Proc. Vol. 3390, Orlando, Fl., April 1998, p. 551-559.
6. Yingwei L., Sundararajan N. and Saratchandran P. Performance evaluation of a sequential minimal radial basis function (RBF) neural network learning algorithm. IEEE Trans. on Neural Networks, 9, 1998, p. 308-318.
7. Fahlman S. E. and Lebiere C. The cascade-correlation learning architecture. Technical Report CMU-CS-90-100, School of Computer Science, Carnegie Mellon University, Pittsburgh, PA, 1990.
8. Cun Y. L., Denker J. S., Solla S. A. Optimal Brain Damage. Advances in Neural Information Processing Systems, 2, 1990, p. 598-605.
9. Hassibi B. and Stork D. G. Second-order derivatives for network pruning: Optimal brain surgeon. In: Advances in Neural Information Processing Systems, Hanson et al. (Eds), 1993, p. 164-171.
10. Prechelt L. Connection pruning with static and adaptive pruning schedules. Neurocomputing, 16, 1997, p. 49-61.
11. Takagi T. and Sugeno M. Fuzzy identification of systems and its application to modeling and control. IEEE Trans. on System, Man and Cybernetics. 15, 1985, p. 116-132.
12. Gantmacher F. R. The Theory of Matrices. Chelsea Publ. Comp., New York, 1977
13. Narendra K. S. and Parthasarathy K. Identification and control of dynamic systems using neural networks. IEEE Trans. on Neural Networks, 1, 1990, p. 4-26.
14. Scott I. and Mulgrew B. "Orthonormal function neural network for nonlinear system modeling". In: Proceedings of the International Conference on Neural Networks (ICNN-96), June, 1996.
15. Patra J.C. and Kot A.C. Nonlinear dynamic system identification using Chebyshev functional link artificial neural network. IEEE Trans. on System, Man and Cybernetics – Part B, 32, 2002, p. 505-511.
16. Bodyanskiy Ye.V., Kolodyazhniy V.V., and Slipchenko O.M. "Forecasting neural network with orthogonal activation functions" In: Proc. of 1st Int. conf. "Intelligent decision-making systems and information technologies", Chernivtsi, Ukraine, 2004, p. 57. (in Russian)
17. Bateman, H., Erdelyi, A.: Higher Transcendental Functions. Vol.2. McGraw-Hill (1953)

18. Liying M., Khorasani K. Constructive Feedforward Neural Network Using Hermite Polinomial Activation Functions. IEEE Trans. On Neural Networks, 16, No. 4, 2005, p.821–833.
19. Baumann M. Nutzung neuronale Netze zur Prognose von Aktionkursen. – Report Nr. 2/96, TU Ilmenau, 1996.
20. Fueser K. Neuronale Neteze in der Finanzwirtschaft. – Wiesbanden: Gabler, 1995.

Authors' Information

Yevgeniy Bodyanskiy - Dr. Sc., Prof., Head of Control Systems Research Laboratory, Kharkiv National University of Radio Electronics, Lenin Av., 14, Kharkiv, 61166, Ukraine, e-mail: bodya@kture.kharkov.ua

Iryna Pliss - Ph.D., Senior research scientist, Control Systems Research Laboratory, Kharkiv National University of Radio Electronics, Lenin Av., 14, Kharkiv, 61166, Ukraine, e-mail: pliss@kture.kharkov.ua

Oleksandr Slipchenko - Ph.D., Senior research scientist, Control Systems Research Laboratory, Kharkiv National University of Radio Electronics, Lenin Av., 14, Kharkiv, 61166, Ukraine, e-mail: aslipchenko@kture.kharkov.ua

DOUBLE-WAVELET NEURON BASED ON ANALYTICAL ACTIVATION FUNCTIONS

Yevgeniy Bodyanskiy, Nataliya Lamonova, Olena Vynokurova

Abstract: In this paper a new double-wavelet neuron architecture obtained by modification of standard wavelet neuron, and its learning algorithm are proposed. The offered architecture allows to improve the approximation properties of wavelet neuron. Double-wavelet neuron and its learning algorithm are examined for forecasting non-stationary chaotic time series.

Keywords: wavelet, double-wavelet neuron, recurrent learning algorithm, forecasting, emulation, analytical activation function.

ACM Classification Keywords: I.2.6 Learning – Connectionism and neural nets

Introduction

Recently, in the analysis tasks and the non-stationary series processing under the uncertainty conditions computational intelligence techniques particularly hybrid neural networks are widely used. The most important tasks related to signal processing are forecasting and emulation of dynamic non-stationary states of systems in the future.

For solving such kind of forecasting problems a variety of neural network architectures including hybrid architectures are used. However they are either bulky because of their architecture (for instance multilayer perceptron) or poorly adjusted to learning process in real time. In most cases the activation functions for these neural networks are sigmoidal functions, splines, polynomials and radial basis functions.

In addition wavelet theory is widespread [1-3] and allows to recognize the local characteristics of the non-stationary signals with high accuracy. At the confluence of the two approaches, hybrid neural networks and wavelet theory, have evolved the so-called wavelet neural networks [4-18] that have good approximating properties and sensitivity to the characteristics changes of the analyzed processes.

Previous studies have proposed and described [19-21] attractive features of wavelet neuron such as technical realization, ensured accuracy and learning simplicity. At the same time the wavelet functions are incarnated either at the level of synaptic weights or the neuron output, and as a learning algorithm the gradient learning algorithm with constant step is used. For the improvement of approximation abilities and the acceleration of the learning

process the present work introduces a new structure called double-wavelet neuron and learning algorithm with smoothing and approximation properties.

Wavelet Analytical Activation Functions

Various kinds of analytical wavelets can be used as the activation functions of double-wavelet neuron. Among them we propose to use the triangular wavelet [8] and analytic wavelets generator [9], that have large spectrum of properties.

Fig. 1 shows the proposed triangular wavelet. To fulfil the standard condition $\int_{-\infty}^{\infty}\varphi(x)dx=0$ (the main condition of wavelet existence), it is necessary to perform the condition $h_2=\frac{(b-d)+h_1(c-a)}{(c-e)}$. After that we can write the mathematical expression for this function in the following form

$$\varphi(x,[a,b,c,d,e,h_1])=\begin{cases}0,\ if\ \ x<a\ \&\ x>e;\\ -h_1\dfrac{(x-a)}{(b-a)},\ if\ \ a\le x\le b;\\ \dfrac{(h_1+1)(x-b)}{(c-b)}-h_1,\quad if\ \ b\le x\le c;\\ -\dfrac{(h_2+1)(x-c)}{(d-c)}+1,\quad if\ \ c\le x\le d;\\ h_2\dfrac{(x-d)}{(e-d)}-h_2,\quad if\ \ d\le x\le e.\end{cases} \tag{1}$$

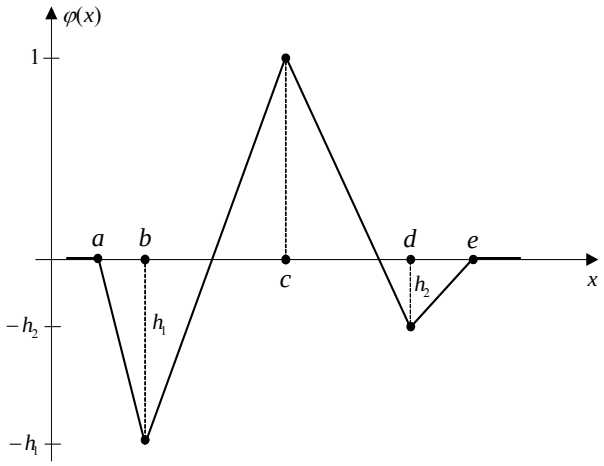


Fig. 1 – Triangular wavelet

Distinctive feature of the proposed triangular wavelet is that this function can be both even and odd, according to the values of function parameters. Such function can be implemented in chip as an activation function of wavelet neuron.

In some situation instead of triangular wavelet using a universal wavelet activation function based on analytic wavelet generator [9] is useful. As is known most of wavelets can be divided into the even and odd functions. Therefore because of the processing signal type it is necessary to choose even analytic wavelets generator

$$y_{even}(x(k))=\sum_{i=1}^{n}a_i\cos(i\,x(k))=a^T\varphi_{even}(x(k)) \tag{2}$$

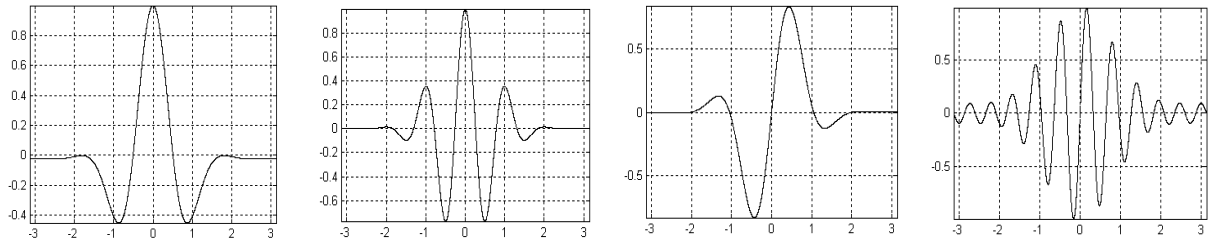
(here $\varphi_{even}(x(k)) = (\cos x(k), \cos 2x(k), \ldots, \cos nx(k))$; a_i are the spectral decomposition coefficients of even wavelet) or odd analytic wavelets generator

$$y_{odd}(x(k)) = \sum_{i=1}^{n} b_i \sin(i\, x(k)) = b^T \varphi_{odd}(x(k)), \tag{3}$$

where $\varphi_{odd}(x(k)) = (\sin x(k), \sin 2x(k), \ldots, \sin nx(k))$; b_i are the spectral decomposition coefficients of even wavelet.

These analytic wavelets generators allow to get the large number of wavelet functions and to tune their parameters during the wavelet neuron training.

Fig. 2 a, b show the representatives of wavelets obtained using even analytical wavelets generator (2), and fig. 2 c, d show the representatives of wavelets obtained using odd analytical wavelets generator (3).



a) b) c) d)

Fig. 2 – Representatives of wavelets that obtained using analytical wavelets generator

It is easy to see, that the situation shown on fig. 2 a is the most similar to the Mexican hat wavelet, on fig. 2 b – the Morlet wavelet, on fig. 2 c – the POLYWOG 2 wavelet, on fig. 2 d – the RASP 2 wavelet [4].

Structure of Double-Wavelet Neuron

Fig. 3 introduces the structure of double-wavelet neuron that consists of nonlinear wavelet synapses which use analytical activation functions.

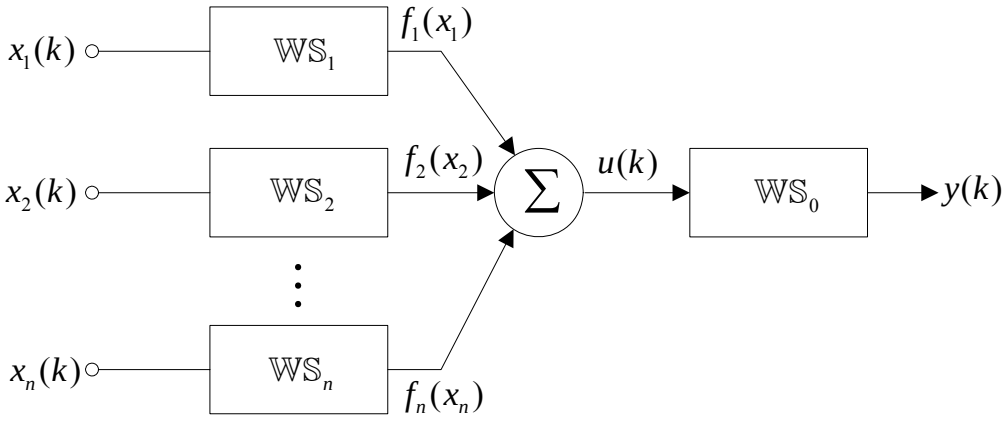


Fig. 3 – Generalized structure of double-wavelet neuron

If a vector signal $x(k) = (x_1(k),\ x_2(k),\ \ldots,\ x_n(k))^T$ (here $k = 0,\ 1,\ 2,\ \ldots$ is the number of sample in the training set or current discrete time) is fed to the input of the double-wavelet neuron shown in Fig. 4 then the output is described by the expression

$$y(k) = f_0\left(\sum_{i=1}^{n} f_i(x_i(k))\right) = f_0(u(k)) =$$
$$= \sum_{l=0}^{h_2} \varphi_{l0}\left(\sum_{i=1}^{n}\sum_{j=0}^{h_1} \varphi_{ji}(x_i(k)) w_{ji}(k)\right) w_{j0} = \sum_{l=0}^{h_2} \varphi_{l0}(u(k)) w_{l0}(k), \quad (4)$$

and depends from the synaptic weights $w_{ji}(k)$, w_{l0} as well as by the values of the used wavelet functions $\varphi_{ji}(x_i(k))$, $\varphi_{l0}(u(k))$, assuming that $\varphi_{00}(\bullet) = \varphi_{0i}(\bullet) \equiv 1$.

The double-wavelet neuron is composed of two sublayers: hidden layer that contains *n*-wavelet synapses with h_1 wavelet-functions in each and output layer that contains one wavelet-synapse with h_2 wavelet-functions.

In each wavelet-synapse, the wavelets that differ between each other by dilation, translation and bias factors are realized.

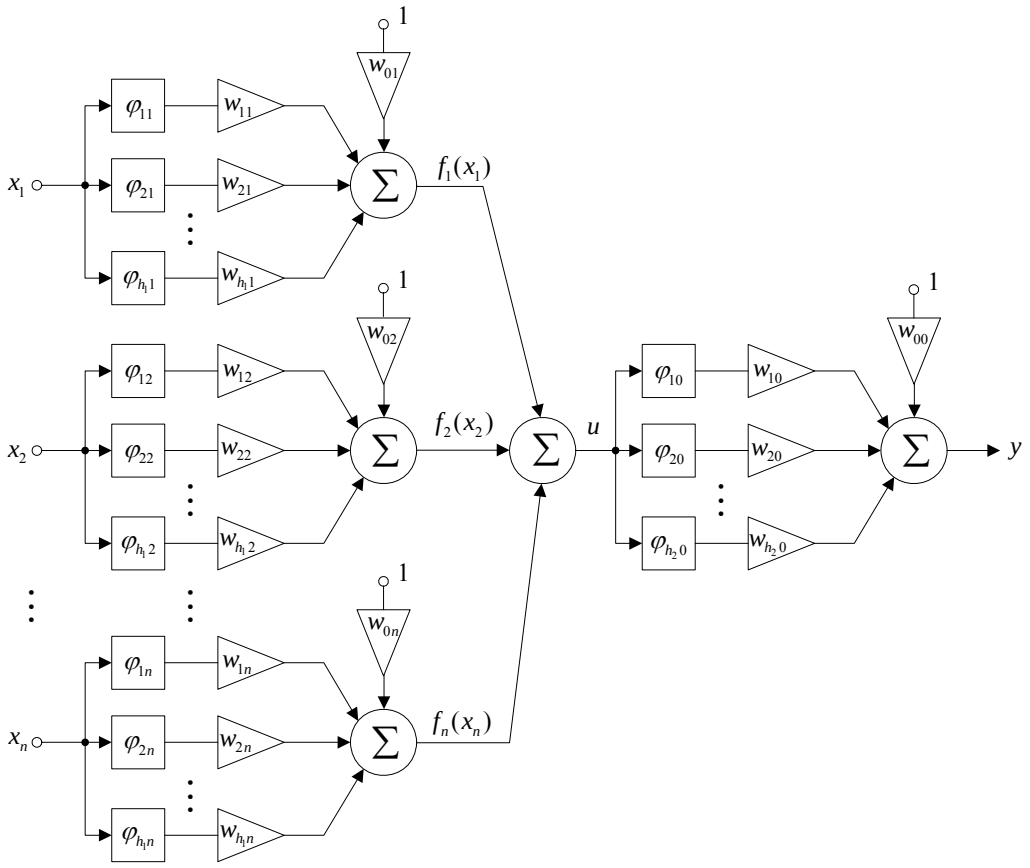


Fig. 4 - Architecture of double-wavelet neuron with nonlinear wavelet-synapses

Synthesis of Double-Wavelet Neuron Learning Algorithm

For the tuning of the output layer of double-wavelet neuron we shall use the one-step criterion

$$E(k) = \frac{1}{2}(d(k) - y(k))^2 = \frac{1}{2}e^2(k), \quad (5)$$

where $d(k)$ is the external training signal.

The learning algorithm for the output layer of double-wavelet neuron on the basis of gradient approach can be written as

$$w_{j0}(k+1) = w_{j0}(k) + \eta_0(k)e(k)\varphi_{j0}(u(k)), \quad (6)$$

or in the vector form

$$w_0(k+1) = w_0(k) + \eta_0(k)e(k)\varphi_0(u(k)), \quad (7)$$

where $w_0(k) = (w_{00}(k), w_{10}(k), w_{20}(k), \ldots, w_{h_2 0}(k))^T$ is $(h_2+1)\times 1$ vector of synaptic weights, $\varphi_0(u(k)) = (1, \varphi_{10}(k), \varphi_{20}(k), \ldots, \varphi_{h_2 0}(k))^T$ is a vector of activation functions, $e(k)$ is a learning error, $\eta_0(k)$ is a learning rate parameter which is subject to determination.

To increase the rate of convergence of the training process it is necessary to turn from gradient procedures to the second-order procedures.

We propose to use the following learning algorithm

$$\begin{cases} w_0(k+1) = w_0(k) + \dfrac{e(k)\varphi_0(u(k))}{\gamma_i^{w_0}(k)}, \\ \gamma_i^{w_0}(k+1) = \alpha\gamma_i^{w_0}(k) + \left\|\varphi_0(u(k+1))\right\|^2, \end{cases} \quad (8)$$

where α is the forgetting factor of out-dated information ($0 \le \alpha \le 1$).

This algorithm has both the smoothing and approximating properties. The tuning of hidden layer is carried out in the same way on the basis of error backpropagation approach by using the same criterion written in the form

$$E(k) = \frac{1}{2}(d(k) - f_0(u(k)))^2 = \frac{1}{2}\left(d(k) - f_0\left(\sum_{i=1}^{n}\sum_{j=0}^{h_1}\varphi_{ji}(x_i(k))w_{ji}(k)\right)\right)^2. \quad (9)$$

The learning algorithm for the hidden layer of double-wavelet neuron on the basis of gradient optimization has the form

$$w_{ji}(k+1) = w_{ji}(k) + \eta(k)e(k)\, f_0'(u(k))\varphi_{ji}(x_i(k)), \quad (10)$$

or in the vector form

$$w_i(k+1) = w_i(k) + \eta(k)e(k)\, f_0'(u(k))\varphi_i(x_i(k)), \quad (11)$$

where $w_i(k) = (w_{0i}(k), w_{1i}(k), w_{2i}(k), \ldots, w_{h_1 i}(k))^T$ is the vector of synaptic weights, $\varphi_i(x_i(k)) = (1, \varphi_{1i}(k), \varphi_{2i}(k), \ldots, \varphi_{h_1 i}(k))^T$ is the vector of wavelet-activation functions, $e(k)$ is the learning error, $\eta(k)$ is the learning rate.

By analogy with (8) one can introduce the procedure

$$\begin{cases} w_i(k+1) = w_i(k) + \dfrac{e(k)\, f_0'(u(k))\,\varphi_i(x_i(k))}{\gamma_i^{w_1}(k)}, \\ \gamma_i^{w_1}(k+1) = \alpha\gamma_i^{w_1}(k) + \left\|\varphi_i(x_i(k+1))\right\|^2, \end{cases} \quad (12)$$

where $0 \le \alpha \le 1$.

Besides the quadratic goal function (5) for the learning algorithm synthesis the goal function based on Trefferquote criterion, Wegstrecke criterion and hybrid criterions can be used [23-25].

Simulation Results

Effectiveness of performance of the proposed double-wavelet neuron and its learning algorithm (8), (12) were investigated in the process of solving forecasting problem and chaotic behaviour emulation of nonlinear dynamic system described by the equation

$$x_{n+1} = \frac{5x_n}{1+x_n^2} - 0.5x_n - 0.5x_{n-1} + 0.5x_{n-2} \quad (13)$$

with initial values $x_0 = 0.2,\ x_1 = 0.3,\ x_2 = 1.0$.

The training set contained 10000 samples, and checking set – 500 samples. Double-wavelet neuron had 5 synapses in the hidden layer corresponding to 5 inputs $x(k-4),\ x(k-3), x(k-2), x(k-1),\ x(k),\ (n=5)$ with 20 wavelets in each synapse ($h_i = 20,\quad i = 1\ldots 5$). Output layer consists of 5 wavelets in synapse WS_0. Initial values of synaptic weights were generated in a random way from -0.1 to $+0.1$.

Several criteria were used for the quality rating of forecast:

- mean-square error (RMSE)

$$RMSE = \frac{1}{N}\sum_{k=1}^{N}(x(k) - \hat{x}(k))^2\ ; \quad (14)$$

- mean absolute percentage error (MAPE)

$$MAPE = \frac{1}{N}\sum_{k=1}^{N}\frac{\left|x(k) - \hat{x}(k)\right|}{x(k)}100\%; \quad (15)$$

- Trefferquote [24, 25] representing percentage ratio of correctly predicted directions to actual direction of the signal

$$Trefferquote = \frac{N - \frac{1}{2}\sum_{k=1}^{N}\left|sign(\hat{x}(k) - x(k-1)) - sign(x(k) - x(k-1))\right|}{N}\cdot 100\%; \quad (16)$$

- Wegstrecke [24, 25], representing quality rating of the predicted model (value +1 corresponds to the optimal predictive model, and -1 – to the incorrect forecast) and described by the equation

$$Wegstreke = \frac{\sum_{k=1}^{N} signal(k)(x(k) - x(k-1))}{\sum_{k=1}^{N}\left|x(k) - x(k-1)\right|}, \quad (17)$$

where $signal(k)$ is a sign-function in the form

$$signal(k) = \begin{cases} 1, & if \quad \hat{x}(k) - x(k) > 0, \\ -1, & if \quad \hat{x}(k) - x(k) < 0, \\ 0, & in \ \ other \ cases, \end{cases}$$

where $x(k)$ is the actual value of forecasting process, $\hat{x}(k)$ is the forecast, N is the length of training set.

Fig. 5 shows the results of forecasting process on the basis of data from test set after 10 training epoch with the parameter $\alpha = 0.99$.

Table 1 shows the results of forecasting process on the basis of the double-wavelet neuron compared the results of forecasting process on the basis of standard wavelet-neuron with the gradient learning algorithm, radial basis neural network and multilayer perceptron.

Thus as it can be seen from experimental results the proposed double-wavelet neuron with the learning algorithm (8), (12) having the same number of adjustable parameters ensures the best quality of forecast and high learning rate in comparison with conventional architectures. The experimental results of multilayer perceptron and radial basis function network are worse because the multilayer perceptron had not time to be trained at such a small number of epochs and radial basis function network suffers from the curse of dimensionality.

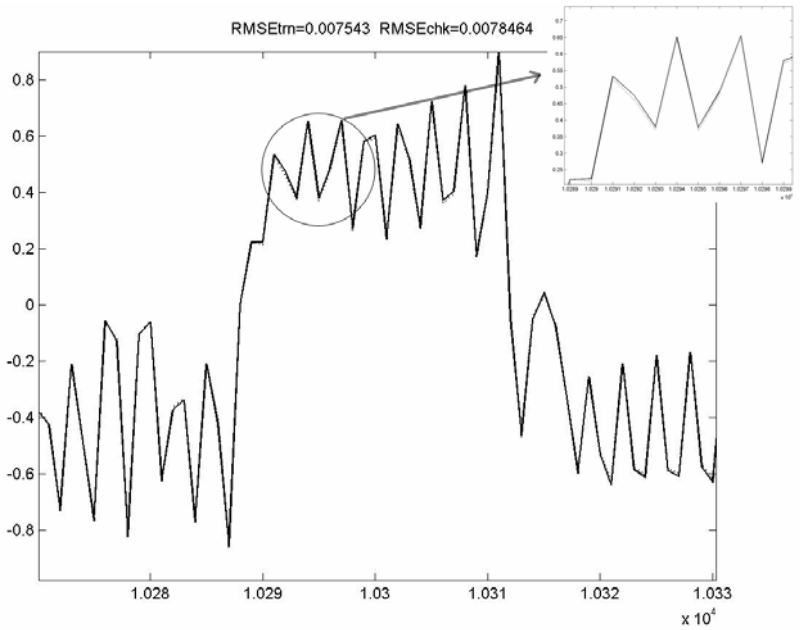


Fig. 5 – Forecasting of chaotic dynamic system behaviour using the double-wavelet neuron

Table 1 - The results of time series forecasting

Neural network/ Learning algorithm	Number of adjustable parameters	Criteria			
		RMSE	MAPE	Wegstrecke	Trefferquote
Double-wavelet neuron/ Proposed learning algorithm of wavelet-synapses parameters (8), (12)	105	0.0076	1.9%	1	99.8%
Wavelet-neuron/ Gradient learning algorithm of parameters of wavelet-synapses with constant step	105	0.0101	3.1%	0.98	98.8%
Radial basis function network / RLSE	105	0.3774	7%	0.4883	55,2%
Multilayer perceptron / Gradient learning algorithm	115	0.5132	9%	0.5882	75,5 %

Conclusions

The double-wavelet neuron architecture and learning algorithm which allows to adjust its parameters are proposed. This algorithm is very simple in the way of its numerical implementation, possesses high rate of convergence and additional smoothing and approximation properties.

Bibliography

[1]. Chui C. K. An Introduction to Wavelets. New York: Academic. 1992, 264 p.

[2]. Daubechies I. Ten Lectures on Wavelets. Philadelphia, PA: SIAM. 1992, 228 p.

[3]. Meyer Y. Wavelets: Algorithms and Applications. Philadelphia, PA: SIAM. 1993, 133 p.

[4]. Lekutai G., van Landingham H.F. Self-tuning control of nonlinear systems using neural network adaptive frame wavelets. Proc. IEEE Int. Conf. on Systems, Man and Cybernetics. Piscataway, N.J. 2, 1997, P. 1017-1022.

[5]. Bodyanskiy Ye., Lamonova N., Pliss I., Vynokurova O. An adaptive learning algorithm for a wavelet neural network. Blackwell Synergy: Expert Systems. 22 (5), 2005, P. 235-240.

[6]. Bodyanskiy Ye., Kolodyazhniy V. Pliss I., Vynokurova O. Learning wavelet neuron based on the RASP-function. Radio Electronics. Computer Science. Control. 1., 2004, P. 118-122.

[7]. Vynokurova O., Bodyanskiy Ye., Lamonova N. Adaptive hybrid wavelet neural network for the solving problem of forecasting and emulation // Proc. of 12th International Conference on Automatic Control "Automatics-2005", Kharkiv: NTU "KPI". - 2005. – Vol.3. – P.40-41. (in Russian)

[8]. Bodyanskiy Ye., Vynokurova O. Triangular wavelet and the formal neuron based on it // Proc. of 3-th Int. Scient. Conf. «Mathematical support and software for artificial intelligence» (MPZIS-2005) – Dnepropetrovsk: DNU. - 2005. – P. 14-15 (in Russian)

[9]. Vynokurova O. Lamonova N., Pliss I. Analytic wavelets generator // Problemy Bioniki. – 2004. - N 60. - P. 104 – 109 (in Russian)

[10]. Billings S. A., Wei H.-L. A new class of wavelet networks for nonlinear system identification. IEEE Trans. on Neural Networks. 16 (4), 2005, P. 862-874.

[11]. Szu H. H., Telfer B., Kadambe S. Neural network adaptive wavelets for signal representation and classification. Opt. Eng. 31, 1992, P. 1907–1916.

[12]. Zhang Q. H., Benveniste A. Wavelet networks. IEEE Trans. on Neural Networks. 3 (6), 1992, P. 889–898.

[13]. Dickhaus H., Heinrich H. Classifying biosignals with wavelet networks. IEEE Eng. Med. Biol. Mag. 15(5), 1996, P. 103–111.

[14]. Cao L. Y., Hong Y. G., Fang H. P., He G. W. Predicting chaotic time series with wavelet networks. Phys. D. 85, 1995, P. 225–238.

[15]. Oussar Y., Dreyfus G. Initialization by selection for wavelet network training. Neurocomputing. 34, 2000, P. 131–143.

[16]. Zhang J., Walter G. G., Miao Y., Lee W. N. W. Wavelet neural networks for function learning. IEEE Trans. on Signal Process. 43(6), 1995, P. 1485–1497.

[17]. Zhang Q. H. Using wavelet network in nonparametric estimation. IEEE Trans. on Neural Networks. 8(2), 1997, P. 227–236.

[18]. Casdagli M. Nonlinear prediction of chaotic time series. Phys. D. 35, 1989, P. 335–356.

[19]. Soltani S. On the use of wavelet decomposition for time series prediction. Neurocomputing. 48, 2002, P. 267–277.

[20]. Yamakawa T., Uchino E., Samatu T. Wavelet neural networks employing over-complete number of compactly supported non-orthogonal wavelets and their applications. IEEE Int. Conf. on Neural Networks, Orlando, USA., 1994, P. 1391-1396.

[21]. Yamakawa T., Uchino E., Samatu T. The wavelet network using convex wavelets and its application to modelling dynamical systems. The Trans. on the IEICE. J79-A. 12, 1996, P. 2046-2053.

[22]. Yamakawa T. A novel nonlinear synapse neuron model guaranteeing a global minimum – Wavelet neuron. Proc. 28th IEEE Int. Symp. On Multiple-Valued Logic. Fukuoka, Japan: IEEE Comp. Soc., 1998, P.335-336

[23]. Bodyanskiy Ye., Lamonova N., Pliss I., Vynokurova O. Learning algorithm wavelet neuron based on combined criterion Radio Electronics. Computer Science. Control. 2., 2005, P. 83-89.

[24]. Baumann M. Nutzung neuronaler Netze zur Prognose von Aktienkursen. Report Nr. 2/96, TU Ilmenau., 1996, 113 S.

[25]. Fueser K. Neuronale Netze in der Finanzwirtschaft. – Wiesbaden: Gabler, 1995. – 437 S.

Authors' Information

Bodyanskiy Yevgeniy – Doctor of Technical Sciences, Professor of Artificial Intelligence Department and Scientific Head of the Control Systems Research Laboratory, Kharkiv National University of Radio Electronics, Lenina av. 14, Kharkiv, Ukraine 61166, Tel +380577021890, e-mail: bodya@kture.kharkov.ua

Lamonova Nataliya – Candidate of Technical Sciences (equivalent Ph.D.), Senior Research Scientist of Control Systems Research Laboratory, Kharkiv National University of Radio Electronics, Lenina av. 14, Kharkiv, Ukraine 61166, Tel +380577021890, e-mail: webmaster@natashka.de

Vynokurova Olena – Candidate of Technical Sciences (equivalent Ph.D.), Senior Research Scientist of Control Systems Research Laboratory Kharkiv National University of Radio Electronics, Lenina av. 14, Kharkiv, Ukraine, 61166, Tel +380577021890, e-mail: vinokurova@kture.kharkov.ua

SELECTING CLASSIFIERS TECHNIQUES FOR OUTCOME PREDICTION USING NEURAL NETWORKS APPROACH

Tatiana Shatovskaya

Abstract: This paper presents an analysis of different techniques that is designed to aid a researcher in determining which of the classification techniques would be most appropriate to choose the ridge, robust and linear regression methods for predicting outcomes for specific quasi-stationary process.

Keywords: classification techniques, neural network, composite classifier

ACM Classification Keywords: F.2.1 Numerical Algorithms and Problems

1. Introduction

There are a lot of approaches to building mathematical models for quasi-stationary process with multicollinearity and noisiness. For example, ridge regression is a linear-regression variant that is used for highly correlated independent variables, as is often the case for a set of predictors that are designed to approximate the same function [1]. Ridge regression adds a constraint that the sum of the squares of the regression coefficients be equal to a constant λ. Varying this parameter produces a set of predictors. Robust methods estimation parameters of mathematical model have stability in relation to infringement of requests normality the rests of model. They are insensitive not only to mistakes in a dependent variable, but also take into account a degree of influence of points of factorial space, that is reveal emissions in independent variables that allows to receive effective estimations of the coefficients regression models. For all methods a necessary condition of a solvency of their estimations is symmetry of allocating of mistakes of regression model.

But the main problem for the researcher is how to select an appropriate method for given task. In some cases using only one classification method for choosing the estimation method could not the solve problem. A multitude of techniques exists for modeling process outcomes. But the selection of modeling techniques to use for a given class of process is a nontrivial problem because there are many techniques from which to choose. It could be that the modeling technique used is not the most appropriate for the task and that accuracy can be increased through the use of a more appropriate model. There are many reasons why a model may have low predictive value.

This paper presents an analysis of different techniques that is designed to aid a researcher in determining which of the classification techniques would be most appropriate to choose the ridge, robust and linear regression methods for predicting outcomes for specific quasi-stationary process. We shall try to see that success can be attained with particular architectures on commonly used data for such process.

2. Model Class Combinations

There are many techniques to construct classifiers that will be able to chosen the ridge, robust and linear regression estimation methods. As usual such classifiers build from the same model class, for example using only neural models, decision trees or discriminant function. According to our goal we suggested another approach to building a diverse set of classifiers from different model classes, such as decision trees, nearest neighbor algorithms, linear discriminant function, neural network [2-5]. It is opening question whether classifiers from similar or dissimilar model classes are combined most effectively.

There are many architectures for combination of classifier [2]. One of them is a modular architecture. Modularity is a very important concept in nature. Modularity can be defined as subdivision of a complex object into simpler objects. The subdivision is determined either by the structure or function of the object and its subparts. Modularity can be found everywhere: in living creatures as well as in inanimate objects. Replication and decomposition are the two main concepts for modularity. These concepts are found in concrete objects as well as in thinking. It is often difficult to discriminate sharply between them: replication and decomposition often occur in combination.

Replication is a way of reusing knowledge. Decomposition is often found when dealing with a complex task. It is a sign of intelligent behavior to solve a complex problem by decomposing it into simpler tasks which are easier to manage and then reassemble the solution from the results of the subtasks.

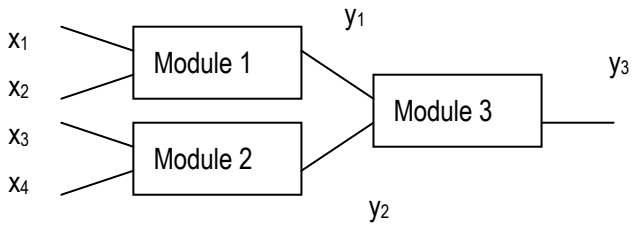


Fig. 1. A Modular Solution

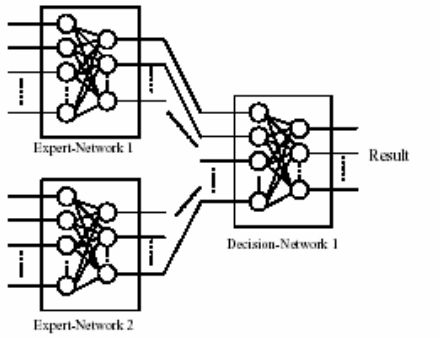


Fig. 2. A Multiple Neural Networks

For example if we choose a neural network as a modular solution we may construct a "building" of neural networks (fig.2).

The term Multiple Neural Networks is used for strongly separated architectures. Each of the networks works independently on its own domain. The single networks are built and trained for their specific task. The final decision is made on the results of the individual networks, often called expert networks or agents. The decision system can be implemented in many different ways: depending on the problem a simple logical majority vote function, another neural network, or a rule based expert system may be employed [6]. The outputs of the expert networks are the input data of the decision network which is trained after the expert networks have been trained. The decision is made according to the outputs of the experts, not directly from the input data. The term Modular Neural Networks (MNN) is very fuzzy. It is used for many different structures [6]. One idea of modular neural network architecture is to build a bigger network by using modules as building blocks. All modules are neural networks. The architecture of a single module is simpler and the sub-networks are smaller than a monolithic network. For this modular approach it is always necessary to have a control system to enable the modules to work together in a useful way. Another idea of modularity is a not-fully connected network. There are many articles and papers published in the field of neural computing [3-6]. An interesting investigation of the relation between structure and function of modular neural networks is given in [7]. The article [7] examines the structural evidence for a modular architecture in the human brain which is given by different psychologists, biologists, and neurologists. Several levels of modularity in the brain are described. Human multitasking abilities and disabilities are explained with the modular and parallel structure of the brain. Individual functions are broken up into sub-processes that can be executed in separate modules without mutual interference. They suggest building more modular artificial neural networks which are similar to the modular structure of the brain [7]. These new architectures may then increase the ability of the network to solve more complex real world problems. Following this motivation for a modular architecture, a new network structure is introduced. The basic building block in this network is the CALM (Categorization and Learning Module) which works on a competitive and unsupervised basis and has the ability to differentiate input patterns in different categories. For a very detailed description of the CALM see [7].

But no meta-generalization scheme is guaranteed to yield neural networks with a minimal generalization error. We concentrates on a recursive modular layered framework for classifier combination or neural networks combination in which the layer of classifiers at each level is used to combine the predictions of the classifiers at the level immediately below.

According to goal of our researching it is suggesting to create two-layer architecture in which the classifiers to be combined are called level-0 classifiers, and the combining classifier is the level-1 classifier. The layering may be iterated to create level-2 classifiers, and so on. Such architecture is a framework for classifier combination in which each layer of classifiers is used to combine the predictions of the classifiers at the immediately preceding

layer. A single classifier at the top-most level outputs the ultimate prediction. The classifier at each layer receives as input a vector of predictions of the classifiers in the layer immediately below. While the information passed from layer to layer may take the form of vectors of predictions, confidence values, or other data, we will limit our attention to systems in which only predictions of estimation methods class are passed from layer to layer. We will also limit ourselves to two-layer generalizes, consisting of a set of component classifiers and a single combining classifier that combines the predictions of the component classifiers.

In effect, such combining classifiers are an attempt to minimize generalization error by using the classifiers in higher numbered layers to learn the types of errors made by the classifiers immediately below. The task of the level-1 (and higher) classifiers is to learn to use the contestant predictions to predict more accurately.

Such combining classifiers framework diagram looks like a multilayer neural network diagram (Fig. 1).

There are certainly analogous aspects to the two frameworks. The distinction between them appears to lie partially in the type of information that is passed from the input layer to the succeeding layer and in the granularity of the classifier nodes themselves. In a neural network, an activation value is passed to forward layers, which may or may not be an ultimate prediction or even have some recognizable interpretation. Generally, in the stacked generalization framework, a "full-fledged" class prediction is passed to the combining classifier, and not just a scalar that somehow contributes to a prediction. Also, in other implementations of such classifiers, the classifiers to be stacked are complex, and may be neural networks themselves.

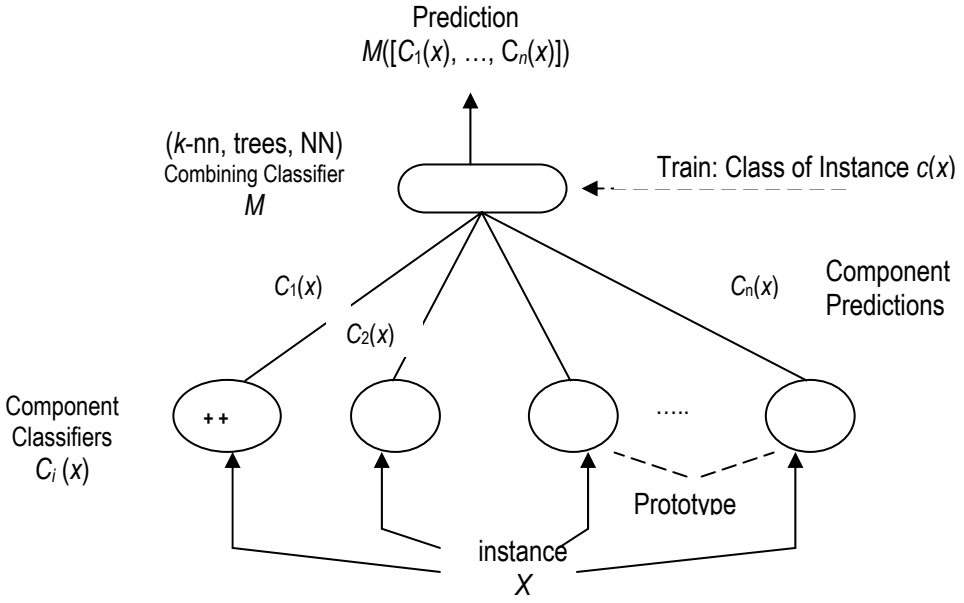


Fig. 3. Classifier Architecture

3. Architecture and Algorithm

We have been given a set of n level-0 (component) learning algorithms, a level-1 learning (combining) algorithm, and a training set of classified instances, T_0. The *n* level-0 learning algorithms should be distinct, so that diverse level-0 classifiers are obtained. Otherwise, no synergy will result from their combination. How to create diverse component classifiers is a fundamental problem for composite classifier construction. Our algorithm has the two phases, training and application.

Training Phase:

1. Train the component classifiers as follows. For each instance in the data set, train each of the *n* level-0 classifiers using the remaining instances. After training, classify the held-out instance using each of the trained level-0 classifiers. Form a vector from the predictions of each of the level-0 classifiers and the actual class of that

instance. These vectors have length n + 1, since they have as components the predictions of each of the n level-0 component classifiers and a class label.

2. Train the level-1 classifier, using as the level-1 training set the collection of vectors of the level-0 classifier predictions and the actual classes. This collection has cardinality $|T_0|$, since there is one level-1 training instance corresponding to each level-0 training instance.

3. Since the level-0 classifiers have not been trained on the entire training set, re-train the level-0 classifiers on the entire training set.

Application Phase:

When presented with a new instance whose class is unknown, classify the instance using each of the level-0 classifiers, deriving an input vector for the level-1 classifier. The derived vector is then classified by the level-1 classifier, which outputs a prediction for the new instance. Leave-one-out cross validation is applied in the training phase to ensure that the level-1 algorithm is trained on the generalizations made for unseen data by the level-0 classifiers. Since "generalization" refers to data outside the training set, this observation is memorialized in the name "composite generalization", as opposed to "stacked classification".

In an experiment with combining linear, ridge, robust regression function showed that using 10-fold cross validation to create the level-1 training data yielded slightly more accurate stacked generalizes than when we applied only leave-one-out cross validation. Also in our experiment has been used decision-tree to generate classifiers that make diverse prediction. We combines a set of trees that have been pruned to the *k*-node trees that displayed the smallest training set error, for various choices of *k*. Investigation of the effect of the combination of neural networks with different numbers of units have been performed too. The accuracies of a given model will vary for the different prediction, so have opportunity to compare it on commonly used data.

In our study we used a commonly used data and compare prediction as follow:

- Maximal accuracy prediction: predicted value must lie within a narrow range of actual value.
- Minimal level prediction: actual value is no less than 5 point below predicted value.
- Significant assistance prediction.

Table 1. Accuracy prediction

Model	Accuracy
Combination of Decision trees	55.7%
Combination of Linear discriminant function	68.9%
Combination of Neural network	76.5%
Linear regression	45.8%

The accuracy for each model for the minimal level prediction is higher than those for the same model for the maximal accuracy prediction. Obtained results shows that combined classifier of neural network have the best accuracy prediction. Does this suggest that artificial neural network models should be used for all outcome predictions in class of quasi-stationary process?

For check-up such situation the experiment was designed to test "whether such composite classifier of combination of neural network can be used to separate ridge and robust estimation methods for incomplete input information" using a set of neural network.

As income information from quasi-stationary process with multicolinearity and noisiness for level-0 classifiers used: volume of sample, number of independent variables, degree of multicollinearity, dispersion of a mistake in a dependent variable, ratio of scales of "littering" and basic distributions of the "polluted" distribution of mistakes of model, degree of pollution of independent variables, the form of emissions in independent variables, length of a tail of the "polluted" distribution of independent variables. As a level-0 classifier we used a Probabilistic neural network, Multiple Perceptron Layers, Radial Basis Function for prediction a class or subclass of methods. When an input task is given, the allocator determines which module (neural network) should be used to fulfill this task. Generally, many modules might be selected to fulfill the task together. Each of these selected modules outputs a result based on local computation. The coordinator then gives the final result based on outputs of the modules. If

the allocator is so strong that a single module can always be correctly selected to perform a given task, the coordinator can be removed. If, on the other hand, the allocator is so weak that all modules must be used to fulfill a task, a strong coordinator would be useful to make the final judgment. Interesting enough, most existing nets are different from each other simply because their allocators or coordinators are stronger or weaker.

For a level-1 classifier as income information has been used a set of criteria of estimation method accuracy. In the table 4 shows the error rate of prediction the most effective method estimation on every level of classifier.

Table 2. Error rate

Model	Error rate
Probabilistic neural network	0.13 %
Multiple Perceptron Layers	0.15 %
Radial Basis Function	0.2 %

Working within this combined classifier on a difficult incoming data from the quasi-stationary process with multicolinearity and noisiness, composite classifier using a probabilistic classifier and a neural network attained accuracy not achieved by any other learning algorithm or modelling techniques. Does this even suggest that the NN-models should use for all outcome of all quasi-stationary process with multicolinearity and noisiness?

It is necessary to note that in choosing a modeling technique we must weigh the costs of the techniques against the accuracies of the techniques. While it may be cost effective for the minimal level prediction to use an NN-model to gain an additional 5-6% in accuracy, it may not be cost effective to use an NN-model or decision trees model for the maximal assistance prediction. In creating a neural network model there are a large number of decisions that must be made, including: Which learning algorithm should be used? Which architecture? How many layers? Which activation functions? What learning rate? How long to train? And so on. The large number of decisions means that there is a very large space of possible neural networks for a given data set. In creating a neural network model the goal is to find the best network by searching through this large space. On our data set the Probabilistic neural network and Counter propagation neural network has the maximal accuracy, but on the other data set it's not necessary. But the idea to composite such type of classifier or to composite classifier that belongs to different type of model by using the recursive-layered framework allows to minimize the error rate of classification.

Bibliography

1. Лесная Н.С., Репка В.Б., Шатовская Т.Б. Метод выбора эффективных процедур оценивания параметров моделей квазистационарных процессов в нейросетевой экспертной системе // Радиотехника. Всеукраинский межведомственный научно-технический сборник. - Харьков. 2001. - № 119. - С. 195-198.
2. Skalak D.B. Prototype selection for composite nearest neighbour classifiers. Neurological Research 2001; 20. P. 116-328.
3. Breiman L., Friedman J.H., Olshen R.A., Stone C.J. Classification and regression tress. Belmont, CA: Wadsworth. 1984.
4. Lang E.W., Pitts L.H., Damron S.L., Rutledge R. Outcome after severe head injury: Analysis of prediction based upon comparison of neural network versus logistic regression analysis. Neurological Research 1997; 19. P. 274-280.
5. Grisby J., Kooken R., Hershberger J. Simulated neural network to predict outcomes, cost and length of stay among orthopaedic rehabilitation patients. Arch. Phys. Med. Rehabil. 1994. Vol. 75. P. 1077-1082.
6. Manchester Metropolitan University. Department of Computing. Report, September1997. A Modular Neural Network Architecture with Additional Generalization Abilities for High Dimensional Input Vectors.
7. Happle H. and Murre. Design and Evolution of Modular Neural Network Architectures. 2000. Vol. 75. P. 256-277

Authors' Information

Tatiana Shatovskaya - Department of Software Engineering, Kharkiv National University of Radioelectronics, Computer Science Faculty, 61166, Kharkiv, Lenin avenue 14, mywork@kture.kharkov.ua

SEARCHING FOR NEAREST STRINGS WITH NEURAL-LIKE STRING EMBEDDING

Artem Sokolov

Abstract: We analyze an approach to a similarity preserving coding of symbol sequences based on neural distributed representations and show that it can be viewed as a metric embedding process.

Keywords sequence similarity, edit distance, metric embeddings, distributed representations, neural networks

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

Introduction

Edit distance (Levenshtein distance) [Levenshtein, 1966] is used in a large number of research areas from genetics and web-search to anomaly detection in network traffic and voice recognition. Taking into account the contemporary data sequences' lengths (millions and billions of symbols) that have to be dealt with in the mentioned areas, the classic $O(n^2)$ edit distance calculation [Vintsyuk, 1968; Wagner, 1974] is not applicable in practice.

These circumstances gave birth to a branch of information theory concerned with the acceleration of edit distance calculation or its approximation (see survey [Navarro, 2001]). An exponential increase of the characteristic lengths of sequences, which are subject to comparison (the genome assembly, the need to compare data flows in information systems, etc.) urged interest to applications of the metric embedding theory (see survey [Indyk, 2004]). This theory is concerned with space mappings that simplify distance calculation [Indyk, 2001]. Levenshtein edit distance embedding to a vector space is known to be an actual open problem [Matoušek, 2002].

Independently, within the framework of the neural network paradigm of AI several approaches were proposed to the task of distributed representation and comparison of strings and other structured objects [Kussul, 1991; Rachkovskij, 2001]. Some approaches aimed at finding similarity of strings were presented in [Sokolov, 2005]. Here we develop one of them, namely, the approach based on the position-dependent thinning of vector representations, giving a theoretical grounding to the obtained scheme with the aid of probabilistic embedding of the edit metrics into the Manhattan space.

Task Description

We seek for a way to effectively calculate Levenshtein edit distance with the help of vector representations or, more specifically, by embedding edit metrics to a vector space. Our method belongs to the group of the so-called q-gram edit distance approximation methods (q-gram is a substring of length q), started by [Ukkonen, 1992]. We observed that the approach based on the distributed representations [Sokolov, 2005] resembles edit distance embedding or sketching methods [Cormode, 2000; Bar-Yossef, 2004; Batu, 2004] and therefore we attempted to combine both presenting the neural coding approach as an edit distance embedding into Manhattan space l_1. In order to show that the proposed method realizes one of the possible embedding definitions [Indyk, 2001], we will give the proofs of:

1) «upper bound», i.e. statements like $ed(x,y) \le k_1 \Rightarrow P[d(v(x),v(y) \le d_1] \ge p_1$ (1a)
2) «lower bound», .i.e. statements like $ed(x,y) > k_2 \Rightarrow P[d(v(x),v(y) > d_2] \ge p_2$. (1b)

This definition (with appropriate values of the parameters envolved) covers embeddings that make discrimination between "near" and "far" strings (see subsection "LSH").

Mapping Description

For the two input strings *x,y* of length *n*, we independently and equiprobably select a sampling window of width *w* in both strings: *x[i,i+w-1]* and *y[i,i+w-1]*. Using fixed parameters of *q*-gram length q_1 and q_2, a *q*-gram vector $v_{w,q}$ is composed for each window for each $q=q_1,\ldots,q_2$ (vector of quantities of each *q*-gram appeared within a string). The obtained vectors corresponding to strings *x* and *y* are concatenated into vectors $v_q(x)$, $v_q(y)$. The Manhattan distance d^{Σ} between them would be the sum of Manhattan distances (i.e., $d_q(s,t) = ||v_q(s)-v_q(t)||_1$) between *q*-gram vectors of the windows:

$$d^{\Sigma}(x[i,i+w-1], y[i,i+w-1]) = \sum_{q=q_1}^{q_2} d_q(x[i,i+w-1],[i,i+w-1]). \quad (2)$$

In the following, using the defined distance, we show that necessary embedding properties (1a) and (1b) hold. Some details and proofs omitted in this short paper will be presented elsewhere.

Lower Bound

De Bruijn graphs. For a string *x* and some parameter *q* de Bruijn graph [Bruijn, 1946] *B[x;q], q ≥ 3* is a graph, whose vertices are all *(q-1)*-grams (*(q-1)*-spectrum) of the string *x.* An edge $a^1a^2\ldots a^q$ connects vertices labeled $a^1a^2\ldots a^{q-1}$ and $a^2a^3\ldots a^q$. Such graphs are widely used in genetics [Pevzner, 1989] and in cryptographic stream ciphers' analysis. Let a de Bruijn graph built using the union of *(q-1)*-spectra of two strings *x,y* be *B[x,y;q],* and the path corresponding to a string *x* be π_x.

Let us consider possible local configurations of paths π_x *and* π_y on *B[x,y;q]*. Let us call the right and left branching points the vertices where the paths, correspondingly, diverge or converge. Let a "*half-loop*" be a subpath of either of the two paths stretching from a right branching point till the following left branching point in the path direction. The situation when there are no half-loops, i.e. in *B[x,y;q]* there is only one left or right branch point, or a left one and a right point following it, is called a "*fork*". In such a configuration, the left folk compulsorily contains as least one of the starting arcs of at least one of the paths, and a right folk contains terminating arcs of at least one of the ways. Let a "*shift*" be a special case of the fork, when there is a non-empty subpath $\pi_c, w \geq |\pi_c| \geq 0$, that $\pi_x = \pi'_x \pi_c$ and $\pi_y = \pi_c \pi'_y$, where π'_x, π'_y are some, possibly empty, subpaths.

A concept of "rotation" will be used to designate a way of obtaining identical spectra from different strings [Ukkonen, 1992, Pevzner, 1995]. A rotation is a situation when *(q-1)*-grams on the edges of a string are identical. It this case a corresponding path on the de Bruijn graph is a cycle, and starting from any of its vertices one can get different strings with the same spectrum.

Lemma 1 *Let* $x,y \in \Sigma^w$ *and there exist substrings* $x' \in x, y' \in y$, $x' \neq y'$, *such that x' is a rotation of y', then*

$$ed(x,y) \leq w + d_q(x,y)/2 - q + 1. \quad (3)$$

In the following we denote $\Delta q = q_2 - q_1$, and $Q = (\Delta q+1)(\Delta q+2)$.

Our aim is to determine such a distance measure between two *w*-wide windows and such a threshold that strings with a distances less than this threshold would represent a shift and can be aligned in a fixed number of operations, by simply editing them at the beginning and the end of the window, namely, by eliminating forks at the edges of the windows. The usual *q*-gram distance (with a fixed *q*) cannot provide the desired result since for any *q* there can be found two strings *x,y,* where on the graph *B[x,y;q]* there will be half-loops.

Therefore we propose distance (2) and the following lemma states that with its aid it is possible to determine, whether the windows can be aligned with a small number of edit operations provided they do not include rotations of substrings.

Lemma 2 *Let* $x,y \in \Sigma^w$, *and x,y do not include substrings that are rotations of each other,* $q_2 > q_1 \geq 3$, $Q \leq 4(w - q_2 + 1)$,

$$d^{\Sigma}(x,y) < Q, \quad (4)$$

then $ed(x,y) < 2(\Delta q+1)$.

The next lemma unifies lemmas 1 and 2 by imposing conditions that allow applying both lemmas, correspondingly, in the case of presence of rotations and their absence.

Lemma 3 *Let* $q_2 > q_1 > ((w-3)^{1/2}+9)/4$, $w \geq 7$, $Q \leq 4(w - q_2+1)$, $Q<w$,

$$w-q_1 \leq \Delta q^2+5\Delta q /2 \quad \text{and} \quad d^{\Sigma}(x,y) < Q, \tag{5}$$

then $ed(x,y) < Q$.

We checked lemma 3 and 4 experimentally for those values of parameter w that still allowed for brute force string comparison. For a binary alphabet, all pairs of 2^w strings were compared for $w = 8,\ldots,17$ (experiment ran for 4 days). For a ternary alphabet all pairs of 3^w strings were compared for $w = 8,\ldots,10$ (2 days). None of the experiments has found a pair of strings violating the lemma.

Let there be two types of pairs of windows $x[i,i+w-1],y[i,i+w-1]$: "good" and "bad" – correspondingly those for which condition (4) holds or not. The next lemma says that it is possible to simultaneously align successive "good" windows.

Lemma 4 *Let conditions of lemma 3 hold,* $w=7,\ldots,n$, *if for all* $i=1,\ldots,n-w+1$ *it holds that* $d^{\Sigma}(x[i,i+w-1],y[i,i+w-1])<Q$, *then* $ed(x,y)<2Q$.

The next lemma defines the minimal possible distance between two "good" pairs of windows, with "bad" pair between them. Denoting $t=w-q_2-\Delta q$ we show that the maximum distance between two "good" pairs of windows without a possibility to contain a "bad" one between them is $t+2$.

Lemma 5 *Let* $x,y \in \Sigma^m$, $t > 2$ *and the conditions of lemma 3 hold, if for some* $i,j=1,\ldots,m-w+1$, $j >i$, $j-i < t+2$

$$d^{\Sigma}(x[i,i+w-1],y[i,i+w-1]) < Q, \quad d^{\Sigma}(x[j,j+w-1],y[j,j+w-1]) < Q, \tag{6}$$

then for all $i'=i,..,j$ $d^{\Sigma}(x[i',i'+w-1],y[i',i'+w-1]) < Q$.

Let N be the number of "bad" windows. Let us find the upper limit (lemma 6) on the edit cost for all possible arrangements of N windows, using the following string edit algorithm. Assume we have aligned strings up to position $j-1$. If all consecutive pairs of windows, beginning from position j and to $j+r$, are "good", then we align them with not more than $2Q$ operations, using the result of lemma 4, and continue from position $j+r+w-1$. If the next pair of windows in position j is bad, we use one edit operation to replace symbol $x[j]$ with symbol $y[j]$, thus aligning one symbol and continuing to the next pair of windows in position $j+1$.

Lemma 6 *Let conditions of lemma 3 hold. Let* $T=\lfloor (n-1)/w \rfloor$. *The cost of aligning* strings *y and x with the help of the above algorithm is upper bounded with*

$$\max(Q+N, (2Q-1)\min(T,N)+\min(N,n-1-(w-1)\min(T,N))+2Q) \leq 2Q(\lceil N/t \rceil+1) \tag{7}$$

Finally, for the independent and equiprobable window sampling we get from (7) the following lemma that specifies property (1b):

Lemma 7 *For* $x,y \in \Sigma^n$ *and holding conditions of lemma 3, if* $ed(x,y) > k_2$, *then*

$$P[d^{\Sigma}(x,y) \geq Q] > (t\, k_2/2Q-2) / (n-w+1). \tag{8}$$

Nearest Neighbor Search

In this part we consider a possible procedure for the nearest string search (NNS) with the help of edit distance approximation described above. Let $P=\{p_1,\ldots,p_P | p_i \in P\}$ be a collection of strings and p_0 be the input probe string, to which it is necessary to find the nearest one (by the edit distance) from P. Searching for the exact nearest neighbor is often a laborious task. Namely, for large dimensionalities of the input space (in our case it is $d=|\Sigma|^n$) the existing NNS algorithms are reduced to the linear search on P. On the other hand, the "approximately" nearest neighbor is often sufficient in applications and it is often much easier to find it.

First we transform vectors v_q into hash-values distributed around $||v_q||$ with the help of a *p*-stable distribution, and a modified scheme from [Datar, 2004] (see subsection "Modification"). Then we apply the well-known scheme of locality-sensitive hashing (LSH) [Indyk, 1998].

LSH. Let us describe the original [Indyk, 1998] LSH scheme applied to strings with the classic edit metrics. Define a ball with radius *r* containing points distanced from its center not farther then *r* : $S(t,r)=\{q \mid ed(q,s) \le r\}$.

Definition of locality-sensitive functions. A family of hash-functions $H=\{h: \Sigma \to X\}$ is called (r_1,r_2,p_1,p_2)-sensitive, if for any $x,y \in \Sigma^n$ and any independently and equiprobably chosen $h \in H$ holds the following:

$$t \in S(s,r_1) \Rightarrow P[h(t)=h(s)] \ge p_1 \quad \text{and} \quad t \notin S(s,r_2) \Rightarrow P[h(t)=h(s)] \le p_2, \quad r_1<r_2 \quad \text{and} \quad p_1<p_2 \tag{9}$$

Compose random hash-vectors $g_j=(h_1,\ldots,h_K)$, $j=1,\ldots,L$ from functions *h*. Additionally, we create cells where we put a string $p_i \in P$ based on the value of the hash-vector $g(p_i)$: a string p_i is put into a cell with an identifier equal to the hash-vector value. The aim is to get high collision probability between nearby strings, and low probability between distant ones. Then, applying the same hash to the probe we check whether it equals one of the previously stored hashes of vectors from *P*: for probe p_0 we calculate all hash-vectors $g_i(p_0)$, $i=1,\ldots,L$ and examine corresponding cells. If some cell contains a string $p^*_i \in S(p,r_2)$, the algorithm returns YES and p^*_i and NO otherwise (thus representing a solution to the so called (r_1,r_2)-PLEB task [Indyk, 1998]). The algorithm terminates after checking *2l* cells. For such a procedure, the following theorem holds:

Theorem 1 [Indyk, 1998] *Let H be a (r_1,r_2,p_1,p_2)-sensitive family of functions, $K=-\ln|P|/\ln(p_2)$, $L=|P|^{\rho}$, where $\rho=\ln(p_1/p_2)$. Then the above algorithm solves (r_1,r_2)-PLEB task and takes $O(|\Sigma|^q|P|+|P|^{1+\rho})$ space, $O(|P|^{\rho})$ distance calculations, and $O(|P|^{\rho}K)$ calculations of hash functions.*

LSH with a 1-stable distribution. In [Datar, 2004], it is proposed to use a particular property of stable distributions, that linear combinations of theirs random values ϕ_i are distributed as one such random variable multiplied by the norm of linear combination's coefficients. Due to linearity of scalar product, $(v_1,\phi) - (v_2,\phi) \sim ||v_1-v_2||_p\phi$. Hash-functions are defined as:

$$h(v)=\lfloor((v_q,\boldsymbol{\phi})+b)/r\rfloor, \tag{10}$$

where *b* is an equiprobably distributed random variable on $[0,r]$, $\boldsymbol{\phi}$ is a vector with elements taken from Cauchy distribution. If one divides a real axis into equal intervals, then, intuitively, vectors with the similar norm will likely fall into the same interval. It is possible to show [Datar, 2004] that for two fixed vectors the hash-function (10) is

$$p(c) = \int_0^r \frac{1}{c} f(c)\left(1-\frac{t}{c}\right)dt = \frac{1}{\pi}\left(2\arctan\left(\frac{r}{c}\right) - \frac{c}{r}\ln\left(1+\left(\frac{r}{c}\right)^2\right)\right) \tag{11}$$

where *f(.)* is the probability density function of the absolute value of ϕ, and $c=||v_1-v_2||$ is the distance between the hashed vectors. As *p(c)* is a monotonically decreasing function, the family of such functions is locality-sensitive (see (9)). So, hash-functions (10) can be used in the LSH scheme.

Modification. We will use the hash-functions (10) in a different way from that described above, to pursue the analogy to the distributed approaches [Sokolov, 2005]. Instead of multiplying a fixed vector v_q by a number of random vectors ϕ to form hash-vectors g_j, we take *K* random vectors $v^i_q(s)$ obtained by random and independent sampling with a window of width *w* from string *s* (see "Mapping description"). For each of them, we generate a separate random vector ϕ_i whose elements are taken from the Cauchy distribution. Let us designate the scalar product of these two vectors $h'_i=(v^i_q(s),\phi_i)$ and fix the hash-functions of the form $h'(v)=\lfloor(h'_i+b)/r\rfloor$. Taking into account lemmas 1 and 7, the following lemma holds indicating that the family of such functions is also locality-sensitive.

Lemma 8 Let *function p(.) be defined as in (11). For the collision probability of hash functions h' it holds*

$$P[h'(x)=h'(y) \mid ed(x,y) \le k_1] \le p(2k_1(\Delta q+1))(1-k_1w/(n-w+1)),$$
$$P[h'(x)=h'(y) \mid ed(x,y) \ge k_2] \le 1-(tk_2/2Q-2)(1-p(Q)). \tag{12}$$

With this method of hash-function formation, we get output vectors without matrix multiplying of intermediate *q*-gram representations by random vectors ϕ, thus obtaining a scheme consistent with the neural network approach [Sokolov, 2005].

In the original setting parameter *r* in (10) can be chosen, e.g., to minimize p_1/p_2 (see [Datar, 2004]) and to speed up the NNS procedure (theorem 1). However, here we will use *r* to fulfill requirements (9) on r_1, r_2 и p_1, p_2, that, together with (12), leads to the conclusion that we should have

$$k_2 > 2Q\left(2+\frac{n-w+1}{t\left(1-p(d_2)\right)}\left(1-p(d_1)+\frac{wk_1p(d_1)}{n-w+1}\right)\right). \quad (13)$$

This should be at most $\Theta(n)$ for the property (1b) not to become trivial. We can achieve this by letting *r* be some function of the string length *n,* e.g. $r=n^{\mu}$. We also set $w=n^{\gamma}$. Taking into account asymptotic behavior of $t=\Theta(w)$, $\Delta q=\Theta(w^{1/2})$, $Q=\Theta(w)$, analysis shows, that the optimal values for the parameters are $\mu = \gamma = 2/3$. And from (13), the lower bound on the k_2 growth rate is

$$k_2 = \Omega(k_1 n^{2/3} \ln n) \quad (14)$$

Parameter ρ (see theorem 1) determines search efficiency and resource requirements for the described LSH scheme. If , for $\varepsilon > 1$, k_2 is chosen as follows:

$$k_2 = 2Q\left(2+\frac{\varepsilon(n-w+1)}{t\left(1-p(d_2)\right)}\left(1-p(d_1)+\frac{wk_1p(d_1)}{n-w+1}\right)\right),$$

it can be shown, using the same method as for locality-sensitive family of functions for the Hamming distance [Indyk, 1998], that $\rho = O(1/(1+\varepsilon))$.

Ternarization. It is rather attractive to have either binary or ternary vectors at the output, because they are more beneficial than integer-valued ones because of a more efficient implementation. Moreover, (sparse) binary or ternary vectors are widely used in distributed processing models [Rachkovskij, 2001]. Hash-function (10) will take ternary values $\{-1,0,1\}$ if $1 < h(v) < 2$ and so $-r < (\phi,v) < r$. Integrating $(\pi(1+x^2))^{-1}$ with limits from $-r/||v||$ to $r/||v||$ we get for the percentage of ternary elements in the output: $2\arctan(r/||v||)/\pi$. Density of zero elements is an important parameter in distributed representations, and for ternary vectors it is given by $\arctan(r/||v||-||v|| \ln(1+(r/||v||)^2)/2r)/\pi$ and is increasing with the growth of $r=n^{2/3}$.

Conclusion

We analyzed the concept of the distributed representations of sequences [Sokolov, 2005] from the point of view of metric embeddings, presented a new *q*-gram approximation method of the edit distance, and proved the possibility of constructing locality-sensitive functions. Thus we showed that the distributed representations used for the comparison of sequential data in the neural network paradigm could be justified with the aid of the methods from the embedding theory. This approach can also be considered as the substantiation of the Broder approach [Broder, 1995] who takes for the document similarity measure the degree of the coincidence of the sets of their *q*-grams and also other bag-of-grams methods. We also gave conditions for obtaining binary and ternary vectors at the output that can be useful for a unified approach to representation and processing of various data types and modalities [Rachkovskij, 2001].

A prospective direction of further work may be to check if lemma 3 can could be strengthened to guarantee $ed(x,y) < 2(\Delta q+1)$ with modified conditions on q_1 and Δq. If the available preliminary experimental indications of this are proved, it would lead to a considerable improvement of the lower bound (14) in the modified LSH scheme and in a deterministic variant of the mapping.

Bibliography

[Bar-Yossef, 2004] Z. Bar-Yossef, T.S. Jayram, R. Krauthgamer, R. Kumar: Approximating Edit Distance Efficiently. *FOCS*, pp. 550-559, 2004

[Batu, 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, 1998] A. Broder. On the resemblance and containment of documents. In *SEQS: Sequences '97*, 1998.

[Bruijn, 1946] N. G. de Bruijn. A combinatorial problem. In *Koninklijke Nederlandsche Akademie van Wetenschappen*, volume 49, 1946.

[Cormode, 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

[Datar, 2004] M. Datar, N. Immorlica, P. Indyk, V. Mirrokni. Locality-sensitive hashing scheme based on p-stable distributions. In *20-th annual symposium on Computational geometry*, pages 253–262, Brooklyn, New York, USA, 2004.

[Indyk, 1998] P. Indyk, R. Motwani. Approximate nearest neighbors: towards removing the curse of dimensionality. In *Proc. of 30th STOC*, pp. 604–613, 1998.

[Indyk, 2001] P. Indyk. Algorithmic aspects of geometric embeddings. In *FOCS*, 2001.

[Indyk, 2004] P. Indyk. Embedded stringology. Talk at *15-th Annual Combinatorial Pattern Matching Symposium*, July 2004.

[Kussul, 1991] Kussul, E. M., & Rachkovskij, D. A. (1991). Multilevel assembly neural architecture and processing of sequences. In A .V. Holden & V. I. Kryukov (Eds.), Neurocomputers and Attention: Vol. II. Connectionism and neurocomputers (pp. 577-590). Manchester and New York: Manchester University Press.

[Levenshtein, 1966] V. I. Levenshtein. Binary codes capable of correcting deletions, insertions, and reversals. *Soviet Physics - Doklady*, 10(8):707–710, February 1966.

[Matoušek, 2002] Open problems. In J. Matoušek, editor, *Workshop on Discrete Metric Spaces and their Algorithmic Applications*, Haifa, March 2002.

[Navarro, 2001] G. Navarro. A guided tour to approximate string matching. *ACM Computing Surveys*, 33(1):31–88, 2001.

[Pevzner, 1989] P. A. Pevzner,P. L-tuple DNA sequencing: computer analysis. *J. Biomol. Struct. Dyn.*, 7, 63–73,1989

[Pevzner, 1995] P. A. Pevzner. DNA physical mapping and alternating eulerian cycles in colored graphs. *Algorithmica*, 13(1/2):77–105, 1995.

[Rachkovskij, 2001] D. A. Rachkovskij, E. M. Kussul. Binding and Normalization of Binary Sparse Distributed Representations by Context-Dependent Thinning, *Neural Comp*. 13: 411-452, 2001.

[Shamir, 2004] R. Shamir. Lecture notes in Analysis of Gene Expression Data, DNA chips and Gene Networks: Sequencing by hybridization. www.cs.tau.ac.il/~rshamir/ge/04/scribes/lec02.pdf, 2004.

[Sokolov, 2005] A. Sokolov, D. Rachkovskij. Some approaches to distributed encoding of sequences. In *Proc. of XI-th International Conference Knowledge-Dialogue-Solution*, volume 2, pp. 522–528, Varna, Bulgaria, June 2005.

[Ukkonen, 1992] E. Ukkonen. Approximate string-matching with q-grams and maximal matches. *Theor. Comput. Sci.*, 92(1):191–211, 1992.

[Vintsyuk, 1968] T. K. Vintsyuk. Speech discrimination by dynamic programming. *Kibernetika (Cybernetics)*, (4):81–88, 1968.

[Wagner, 1974] R. A. Wagner, M. J. Fischer. The string-to-string correction problem. *J. of the ACM*, 21(1):158–173, 1974.

Authors' Information

Artem M. Sokolov – International Research and Training Center of Information Technologies and Systems; Pr. Acad. Glushkova, 40, Kyiv, 03680, Ukraine; e-mail: sokolov (at) ukr.net

TABLE OF CONTENTS OF VOLUME 14, NUMBER 3