I T H E A

International Journal

INFORMATION CONTENT & PROCESSING

2014 Volume 1 Number 2

International Journal
INFORMATION CONTENT & PROCESSING
Volume 1 / 2014, Number 2

IJ ICP is official publisher of the scientific papers of the members of the ITHEA® International Scientific Society

IJ ICP rules for preparing the manuscripts are compulsory.
The **rules for the papers** for ITHEA International Journals as well as the **subscription fees** are given on www.ithea.org .
The papers should be submitted by ITHEA® Submission system http://ij.ithea.org .
Responsibility for papers published in IJ IMA belongs to authors.

International Journal "INFORMATION CONTENT AND PROCESSING" Volume 1, Number 2, 2014

Edited by the **Institute of Information Theories and Applications FOI ITHEA**, Bulgaria, in collaboration with
Institute of Mathematics and Informatics, BAS, Bulgaria,
V.M.Glushkov Institute of Cybernetics of NAS, Ukraine,
Universidad Politechnika de Madrid, Spain,
Hasselt University, Belgium
Institute of Informatics Problems of the RAS, Russia,
St. Petersburg Institute of Informatics, RAS, Russia
Institute for Informatics and Automation Problems, NAS of the Republic of Armenia.

Publisher: **ITHEA®**
Sofia, 1000, P.O.B. 775, Bulgaria. www.ithea.org, e-mail: info@foibg.com
Technical editor: Ina Markova

Printed in Bulgaria

ISSN 2367-5128 (printed) **ISSN 2367-5152 (online)**

Polynomial Regression using a Perceptron with Axo-axonic Connections

Nuria Gómez Blas, Luis F. de Mingo, Alberto Arteta

Abstract: *Social behavior is mainly based on swarm colonies, in which each individual shares its knowledge about the environment with other individuals to get optimal solutions. Such co-operative model differs from competitive models in the way that individuals die and are born by combining information of alive ones. This paper presents the particle swarm optimization with differential evolution algorithm in order to train a neural network instead the classic back propagation algorithm. The performance of a neural network for particular problems is critically dependant on the choice of the processing elements, the net architecture and the learning algorithm. This work is focused in the development of methods for the evolutionary design of artificial neural networks. This paper focuses in optimizing the topology and structure of connectivity for these networks.*

Keywords: *Social Intelligence, Neural Networks, Grammatical Swarm, Particle Swarm Optimization, Learning Algorithm.*

ACM Classification Keywords*: F.1.1 Theory of Computation - Models of Computation, I.2.6 Artificial Intelligence - Learning.*

Introduction

Neural networks are non-linear systems whose structure is based on principles observed in biological neuronal systems. A neural network could be seen as a system that can be able to answer a query or give an output as answer to a specific input. The in/out combination, i.e. the transfer function of the network is not programmed, but obtained through a training process on empiric datasets.

In practice the network learns the function that links input together with output by processing correct input/output couples. Actually, for each given input, within the learning process, the network gives a certain output which is not exactly the desired output, so the training algorithm modifies some parameters of the network in the desired direction. Hence, every time an example is input, the algorithm adjusts its network parameters to the optimal values for the given solution: in this way the algorithm tries to reach the best solution for all the examples. These parameters we are speaking about are essentially the weights or linking factors between each neuron that forms our network.

Neural Networks' application fields are tipically those where classic algorithms fail because of their unflexibility (they need precise input datasets). Usually problems with unprecise input datasets are those whose number of possible input datasets is so big that they can't be classified. For example in image recognition are used probabilistic algoithms whose efficency is lower than neural networks' and whose caratheristics are low flexibility and high development complexity. Another field where classic algorithms are in troubles is the analisys of those phenomena whose matematical rules are unknown.

There are indeed rather complex algorithms which can analyse these phenomena but, from comparisons on the results, it comes out that neural networks result far more efficient [Hu and Hwang 2001; Katagiri 2000]: these algorithms use *Fourier*'s trasform to decompose phenomena in frequential components and for this reason they result highly complex and they can only extract a limited number of harmonics generating a big number of approximations. A neural network trained with complex phenomena's data is able to estimate also frequential components, this means that it realizes in its inside a *Fourier*'s trasform even if it was not trained for that! One of the most important neural networks' applications is undoubtfully the estimation of complex phenomena such as meteorological, financial, socio-economical or urban events. Thanks to a neural network it's possible to predict, analyzing hystorical series of

datasets just as with these systems but there is no need to restrict the problem or use *Fourier*'s tranform. A defect common to all those methods it's to restrict the problem setting certain hypothesis that can turn out to be wrong. We just have to train the neural network with hystorical series of data given by the phenomenon we are studying.

Calibrating a neural network means to determinate the parameters of the connections (synapsis) through the training process. Once calibrated there is need to test the netowrk efficency with known datasets, which has not been used in the learning process. There is a great number of Neural Networks which are substantially distingushed by: type of use, learning model (supervised/non-supervised), learning algorithm, architecture, etc.

This paper focuses on supervised networks with the *backpropagation* learning algorithm and applied to signal analysis with a typical feedforward architecture.

Multilayer Perceptron

Multilayer perceptrons (*MLPs*) are layered feedforward networks [Cover and Tomas 1991] typically trained with static backpropagation. These networks have found their way into countless applications requiring static pattern classification. Their main advantage is that they are easy to use, and that they can approximate any input-output map.

In principle, backpropagation provides a way to train networks with any number of hidden units arranged in any number of layers [Hu 1996; Hu et al. 1994]. In fact, the network does not have to be organized in layers - any pattern of connectivity that permits a partial ordering of the nodes from input to output is allowed. In other words, there must be a way to order the units such that all connections go from ”earlier" (closer to the input) to "later" ones (closer to the output). This is equivalent to stating that their connection pattern must not contain any cycles. Networks that respect this constraint are called feedforward networks; their connection pattern forms a directed acyclic graph or dag.

Backpropagation algorithm can be expressed as equation (1). Note that in order to calculate the error for unit j, we must first know the error of all its posterior nodes (forming the set P_j). Again, as long as there are no cycles in the network, there is an ordering of nodes from the output back to the input that respects this condition. For example, we can simply use the reverse of the order in which activity was propagated forward.

$$\delta_j = f_j^{'}(net_j) \sum_{i \in P_j} \delta_j w_{ij} \tag{1}$$

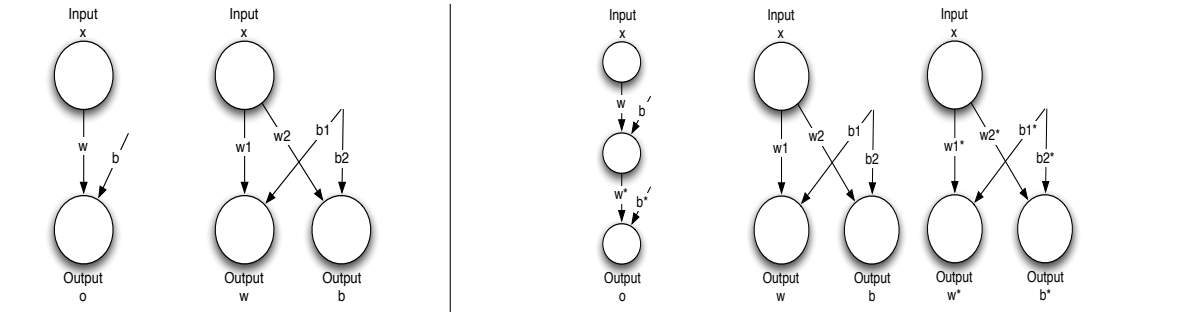
Axo-axonic Neural Networks

The most usual connection type in neural networks is the axo-dendritic connection. This connection is based on the fact that the axon of an afferent neuron is connected to another neuron via a synapse on a dendrite, and modelized in *ANN* model by a weighted activation transfer function. But, there exists many other connection types as: axo-somatic, axo-axonic and axo-synaptic [Delacour 1987]. This paper is focused on the second kind of connection type *axo-axonic*. Merely, the structure of the axo-axonic connection can be sketched by three neurons with a classical axo-dendritic connection and the synaptic axonal termination of N_3 connected to the synapse S_{12}. The principle consists on propagating the action of neuron N_3 as synapse S_{12}. In order to model previous connection type, two neural networks are required. The first (assistant) one will compute the weight matrix of the second (principal) one. And, the second network will output a response, using the previously computed weight matrix, this architecture is named Enhanced Neural Networks *ENN*.

ENN as Taylor series approximators.

It is well known that a function can be approximated with a given error using a polynomial $P(x) = \hat{f}(x)$ with a degree n. The error $f(x) - P(x)$ is measure in such a way that in order to find a suitable approximation (error lower than a known threshold) it is only needed to compute sucessive derivates of function $f(x)$ until a certain degree n.

Enhanced Neural Networks behave as n-degree polynomial approximators depending on the number of hidden layer in the architecture. In order to obtain such behavior all activation functions of the net must be lineal function $f(x) = ax + b$.



$$o = wx + b = (w_1x + b_1)x + w_2x + b_2 = Ax^2 + Bx + C$$

$$o = w^*(wx + b) + b^* = (w_1^*x + b_1^*)[(w_1x + b_1)x + w_2x + b_2] + w_2^*x + b_2^* = Ax^3 + Bx^2 + Cx + D$$

Figure 1: ENN architectures and output expressions

As shown in figure 1 and output equations, the number of hidden layers can be increased in order to increase the degree of the output polynomial, that is, the number n of hidden layers control, in some sense, the degree $n + 2$ of output polynomial of the net.

Table 1 shows how the degree of the output polynomial increases according to the number of hidden layers in the net.

Table 1: Number hidden layers vs. degree of output polynomial

Hidden Layers	Degree $P(x)$	Output Polynomial
0	2	$o = a_2x^2 + a_1x + a_0$
1	3	$o = a_3x^3 + a_2x^2 + a_1x + a_0$
...	...	...
n	$n + 2$	$o = \sum_{i=0}^{n+2} a_ix^i$

The only condition that the learning algorithm must verified is that weights must be adjusted to values related with the sucesive derivates of function $f(x)$ that pattern set represents. Usually such function is unkown therefore, if the network converges with a low mean squared error then all weights of the net have converged to the derivates of function $f(x)$ (the pattern set unkown function), and such weights will gather some information about the function and its derivates that the pattern set represents.

Data sets

Previously defined neural network architecture has been used to approximate different data sets generated using a 2-degree polynomial expression of n variables, $\mathcal{P}_n$

Table 2: Sample of randomly generated values using an uniform distribution.

```
    Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
-0.97600 -0.50130  0.01855 -0.04053  0.41540  0.98880

SAMPLE DATA --   Standard deviation:  0.5708282 , Variance:  0.3258449

  [1] -0.05100 -0.26798 -0.77103 -0.05086 -0.26436 -0.32416 -0.96345 -0.82661  0.89400  0.86154  0.29179  0.86964  0.72917
 [14] -0.38158 -0.26924  0.07974 -0.42825  0.98878 -0.87016 -0.41886  0.08352  0.25384 -0.39868 -0.97603 -0.79159  0.59038
 [27] -0.96696 -0.69930 -0.95632 -0.42189  0.59004  0.65942  0.40932 -0.77566  0.16400  0.21629  0.45231 -0.39479  0.92717
 [40]  0.36792 -0.49954  0.90206  0.51239 -0.72945 -0.94883 -0.97396  0.54177  0.06722  0.00008  0.59842 -0.69638  0.07971
 [53] -0.50657  0.15915  0.35297 -0.73656  0.05649  0.48023  0.97079  0.17456  0.07224  0.92373 -0.64668  0.63300 -0.30577
 [66] -0.16878 -0.35884 -0.15355 -0.20623 -0.97129 -0.80947  0.44842 -0.61685  0.03132 -0.41215 -0.82795  0.00578  0.98312
 [79] -0.20718 -0.70245  0.27336 -0.92647  0.69454 -0.55756  0.21941 -0.03392  0.43355  0.31659 -0.21409  0.13297  0.65948
 [92] -0.74168 -0.61549  0.10033  0.12152  0.06672  0.68932 -0.04284  0.04052  0.11754  0.52441  0.70085 -0.08563  0.23654
```

$$\mathcal{P}_n = (\bar{C} \times \bar{X})^2 + c_0, \tag{2}$$

where $\bar{C} = \{c_0, c_1, c_2, \cdots, c_n\}$ are the coefficients and $\bar{X} = \{x_1, x_2, \cdots, x_n\}$ are the variables.

A neural network with no hidden layer is able to approximate such data sets with no error at all (or at least with a lower bound), according to theoretical results. Next samples presents different empirical results to justify such theoretical proposition.

All testing data, coefficients and variable values, are generated using an uniform distribution in interval $(-1, 1)$. Table 2 show some generated values and the mean, median, max, min and quartiles of such data just to check the uniform properties of generated values. Figure 2 shows a correlation matrix among 10 variables. The table and figure only show part of the data not all.

Polynomial regression: 2 variables

$$\begin{aligned} f(x,y) &= (Ax + By + C)^2 \\ &= A^2x^2 + B^2y^2 + C^2 + 2ACx + 2BCy + 2ABxy \end{aligned} \tag{3}$$

$$\begin{pmatrix} A^2 & i_1 & j_1 \\ i_2 & B^2 & k_1 \\ j_2 & k_2 & C^2 \end{pmatrix}, \text{where } i_1 + i_2 = 2AB, j_1 + j_2 = -2AC, k_1 + k_2 = -2BC \tag{4}$$

A neural network has been trained using a random data set with an uniform distribution and describing the polynomial function:

$$f(x,y) = ((x, y, 1) \times (-0.7909866, -0.7742045, 0.726699))^2 \tag{5}$$

Mean squared error of the network must be equal to 0, according to the theoretical results. Next listing shows obtained results with the proposed neural network architecture. Note that MSE in the training and cross validation data sets is really low (near 0).

```
Number of variables:  2
Coefficients (A, B, C): -0.7909866 -0.7742045 0.726699
Squared coefficients (A*A, B*B, C*C): 0.6256597 0.5993926 0.5280914
Number of patterns:  2000 , Iterations:  102 , Learning rate:  0.05 , Cross val.:  20 %
Mean Squared Error (TRAINING):
 Standard deviation  1.037659e-14 , Variance  1.076736e-28
      Min.    1st Qu.     Median       Mean    3rd Qu.       Max.
-2.442e-14 -1.499e-14 -8.105e-15 -6.531e-15  9.159e-16  2.565e-14
Mean Squared Error (CROSS VALIDATION):
```

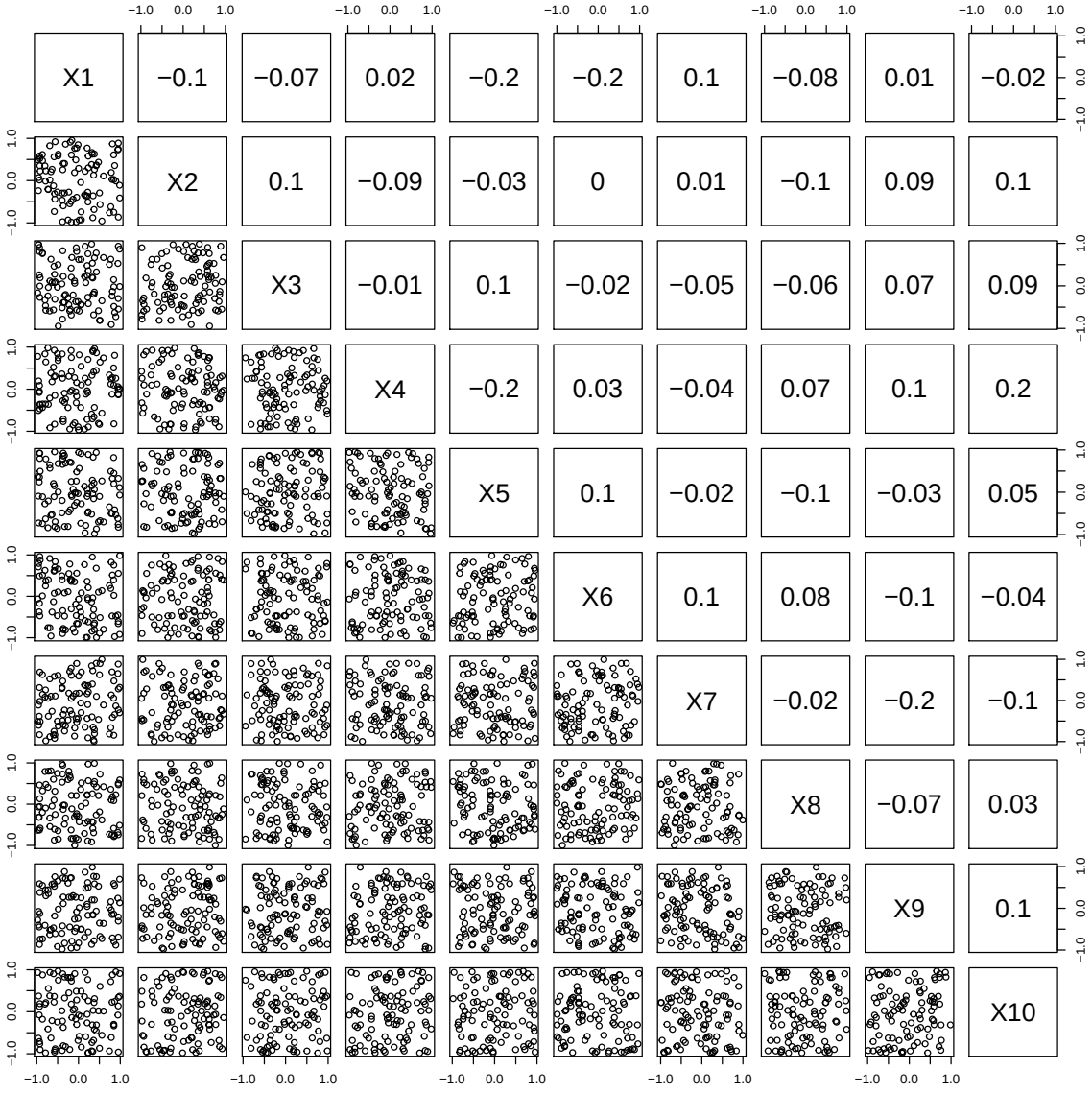


Figure 2: Correlation matrix among 10 variables of the generated data set.

```
 Standard deviation  1.06709e-14 , Variance  1.138682e-28
      Min.    1st Qu.     Median       Mean    3rd Qu.       Max.
-2.465e-14 -1.367e-14 -6.852e-15 -5.291e-15  2.276e-15  2.476e-14
MATRIX Network coefficients:
          [,1]      [,2]      [,3]
[1,] 0.6256597 0.4474168 0.4502775
[2,] 0.7773538 0.5993926 0.5274104
[3,] 0.6993407 0.5978168 0.5280914
A*A = 0.6256597 , B*B = 0.5993926 , C*C = 0.5280914
2*A*B = 1.224771 , -2*A*C = 1.149618 , -2*B*C = 1.125227
```

Coefficients of regression polynomial can be obtained using the weights matrix of trained neural network. Previous matrix shows final weights with a cuasi-null MSE, in our case the coefficients are the following ones (according to equation 4):

$$(A^2, B^2, C^2) = (0.6256597, 0.5993926, 0.5280914) \tag{6}$$

$$(2AB, -2AC, -2BC) = (1.224771, 1.149618, 1.125227) \tag{7}$$

Such results are tottally coherent with equation 5, that is, proposed neural network is able to approximate the data set and generate the polynomial function that describes the data set.

High dimension properties

Previous neural network model behaves with the same *MSE* in case using high dimension data sets. Following listing shows the learning result of a polynomial with 50 variables. Table 3 shows a summary using different number of variables.

```
Number of variables:  50
Coefficients (A, B, ...): -0.4198864 0.6027695 0.6039954 -0.1065414 0.9356629 0.9834093 -0.8226473
0.8743879 -0.7042015 -0.3393832 0.7709589 -0.5013933 -0.8844903 -0.699804 -0.6985472 0.5526986
0.5012762 -0.07739273 0.1085922 -0.914291 0.4816206 -0.7921206 -0.3315833 0.1244581 -0.5412277
-0.5013642 -0.9528081 0.2618788 0.2249716 -0.9169567 0.1750446 0.7072091 -0.6856862 0.3719996
-0.1491458 -0.5018547 0.9422977 0.691155 0.9959811 0.2117917 -0.1234416 0.1298864 -0.8055485
0.2700599 0.9574739 0.8468997 0.3655415 0.6998148 -0.2625522 -0.6242939 -0.9633645
Squared coefficients (A*A, B*B, ...): 0.1763046 0.3633311 0.3648104 0.01135107 0.8754651 0.9670938
0.6767485 0.7645542 0.4958998 0.115181 0.5943776 0.2513952 0.7823231 0.4897256 0.4879682
0.3054758 0.2512778 0.005989635 0.01179226 0.835928 0.2319584 0.627455 0.1099475 0.01548981
0.2929274 0.2513661 0.9078433 0.0685805 0.05061223 0.8408095 0.03064062 0.5001447 0.4701656
0.1383837 0.02224446 0.2518581 0.8879249 0.4776952 0.9919784 0.04485574 0.01523784 0.01687047
0.6489085 0.07293237 0.9167562 0.7172391 0.1336206 0.4897407 0.06893364 0.3897428 0.9280712
Number of patterns:  1850 , Iterations:  150 , Learning rate:  0.05 , Cross val.:  20 %
Mean Squared Error (TRAINING):
 Standard deviation  0.08624005 , Variance  0.007437346
    Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
-0.31370 -0.03923  0.01557  0.01329  0.06721  0.28350
Mean Squared Error (CROSS VALIDATION):
 Standard deviation  0.8489767 , Variance  0.7207615
    Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
-2.22700 -0.52780  0.02572  0.02860  0.57930  2.79500
MATRIX Network coefficients:
0.139645 0.2698724 0.4126177 0.03386078 0.8716183 0.8480131 0.795495 0.8117691 0.4728284 0.202625
0.4350294 0.344523 0.9615263 0.4889943 0.5821863 0.4702517 0.3315289 -0.01390491 -0.01994969
0.9191157 0.1723037 0.6412748 0.06657735 0.1184948 0.3114974 0.2422406 0.9148941 0.1689978
-0.03863459 0.9131449 -0.06600995 0.4795042 0.4226703 0.1084583 0.09037618 0.2399482 0.8859146
0.3590639 0.8691869 -0.01292396 0.04054493 0.1078918 0.6508386 0.144514 0.7362382 0.7127247
0.1572646 0.4372879 0.122485 0.4098477 0.8552449
```

Table 3: Results with different number of variables (see figure 3)

```
Number of variables:  3
Number of patterns:  1200 , Iterations:  20 , Learning rate:  0.05 , Cross validation set:  20 %
Real coefficients:  0.885 0.814 0.114 -0.513
Squared Real coefficients:  0.783 0.663 0.013 0.263
Mean Squared Error (TRAINING): Standard deviation 0.005430024 , Variance 2.948516e-05
Min.  1st Qu.  Median Mean 3rd Qu.  Max.  -0.0172000 -0.0030180 0.0009090 0.0006146 0.0048900 0.0106200
Mean Squared Error (CROSS VALIDATION): Stan ard deviation 0.005166362 , Variance 2.669129e-05
Min.  1st Qu.  Median Mean 3rd Qu.  Max.  -0.0136000 -0.0026350 0.0009429 0.0006865 0.0046690 0.0107400
Network coefficients:  0.77 0.65 0.008 0.274
```

```
Number of variables:  5
Number of patterns:  2000 , Iterations:  20 , Learning rate:  0.05 , Cross validation set:  20 %
Real coefficients:  -0.633 0.07 -0.428 -0.441 -0.858 -0.946
Squared Real coefficients:  0.401 0.005 0.183 0.194 0.736 0.895
Mean Squared Error (TRAINING): Standard deviation 0.001910566 , Variance 3.650264e-06
Min.  1st Qu.  Median Mean 3rd Qu.  Max.  -0.0047460 -0.0015840 -0.0002670 -0.0002031 0.0010550 0.0059510
Mean Squared Error (CROSS VALIDATION): Stan ard deviation 0.001839322 , Variance 3.383107e-06
Min.  1st Qu.  Median Mean 3rd Qu.  Max.  -0.0046280 -0.0013950 -0.0003133 -0.0002007 0.0008592 0.0055970
Network coefficients:  0.404 0.007 0.187 0.197 0.739 0.89
```

```
Number of variables:  7
Number of patterns:  2800 , Iterations:  20 , Learning rate:  0.05 , Cross validation set:  20 %
Real coefficients:  -0.085 0.723 -0.362 -0.662 0.049 -0.006 -0.272 -0.725
Squared Real coefficients:  0.007 0.523 0.131 0.438 0.002 0 0.074 0.525
Mean Squared Error (TRAINING): Standard deviation 0.0006043065 , Variance 3.651864e-07
Min.  1st Qu.  Median Mean 3rd Qu.  Max.  -0.0017650 -0.0005786 -0.0001700 -0.0001428 0.0002686 0.0020250
Mean Squared Error (CROSS VALIDATION): Stan ard deviation 0.0006502922 , Variance 4.228799e-07
Min.  1st Qu.  Median Mean 3rd Qu.  Max.  -1.614e-03 -5.753e-04 -9.671e-05 -8.572e-05 3.291e-04 2.020e-03
Network coefficients:  0.008 0.524 0.131 0.439 0.003 0.001 0.075 0.523
```

```
Number of variables:  9
Number of patterns:  3600 , Iterations:  20 , Learning rate:  0.05 , Cross validation set:  20 %
Real coefficients:  0.601 -0.022 0.753 0.023 0.764 0.589 -0.06 -0.772 0.03 0.777
Squared Real coefficients:  0.361 0 0.567 0.001 0.584 0.347 0.004 0.596 0.001 0.604
Mean Squared Error (TRAINING): Standard deviation 0.0001982064 , Variance 3.928577e-08
Min.  1st Qu.  Median Mean 3rd Qu.  Max.  -5.585e-04 -1.581e-04 -2.589e-05 -1.777e-05 1.126e-04 7.886e-04
Mean Squared Error (CROSS VALIDATION): Stan ard deviation 0.0002006681 , Variance 4.026771e-08
Min.  1st Qu.  Median Mean 3rd Qu.  Max.  -5.901e-04 -1.647e-04 -3.507e-05 -2.336e-05 9.689e-05 7.056e-04
Network coefficients:  0.361 0.001 0.567 0.001 0.584 0.347 0.004 0.596 0.001 0.603
```

```
Number of variables:  11
Number of patterns:  4400 , Iterations:  20 , Learning rate:  0.05 , Cross validation set:  20 %
Real coefficients:  -0.915 -0.737 -0.706 -0.744 0.656 -0.332 0.981 -0.112 0.744 0.59 0.026 0.797
Squared Real coefficients:  0.837 0.543 0.498 0.553 0.43 0.11 0.962 0.013 0.554 0.348 0.001 0.636
Mean Squared Error (TRAINING): Standard deviation 4.617442e-05 , Variance 2.132077e-09
Min.  1st Qu.  Median Mean 3rd Qu.  Max.  -1.404e-04 -4.247e-05 -1.064e-05 -9.864e-06 2.066e-05 1.620e-04
Mean Squared Error (CROSS VALIDATION): Stan ard deviation 4.979714e-05 , Variance 2.479755e-09
Min.  1st Qu.  Median Mean 3rd Qu.  Max.  -1.413e-04 -4.654e-05 -1.265e-05 -1.095e-05 2.230e-05 1.527e-04
Network coefficients:  0.837 0.543 0.498 0.554 0.43 0.11 0.962 0.013 0.554 0.348 0.001 0.635
```

```
Number of variables:  13
Number of patterns:  5200 , Iterations:  20 , Learning rate:  0.05 , Cross validation set:  20 %
Real coefficients:  -0.101 0.877 -0.38 -0.081 0.022 0.13 0.035 -0.428 0.076 0.364 -0.156 -0.52 0.225 -0.893
Squared Real coefficients:  0.01 0.769 0.145 0.007 0 0.017 0.001 0.183 0.006 0.133 0.024 0.271 0.05 0.798
Mean Squared Error (TRAINING): Standard deviation 0.0001415502 , Variance 2.003647e-08
Min.  1st Qu.  Median Mean 3rd Qu.  Max.  -4.066e-04 -1.160e-04 -1.987e-05 -1.510e-05 7.840e-05 5.004e-04
Mean Squared Error (CROSS VALIDATION): Stan ard deviation 0.0001463287 , Variance 2.141208e-08
Min.  1st Qu.  Median Mean 3rd Qu.  Max.  -4.326e-04 -1.208e-04 -2.816e-05 -2.235e-05 7.465e-05 4.004e-04
Network coefficients:  0.01 0.769 0.145 0.007 0.001 0.017 0.001 0.183 0.006 0.133 0.025 0.271 0.051 0.798
```

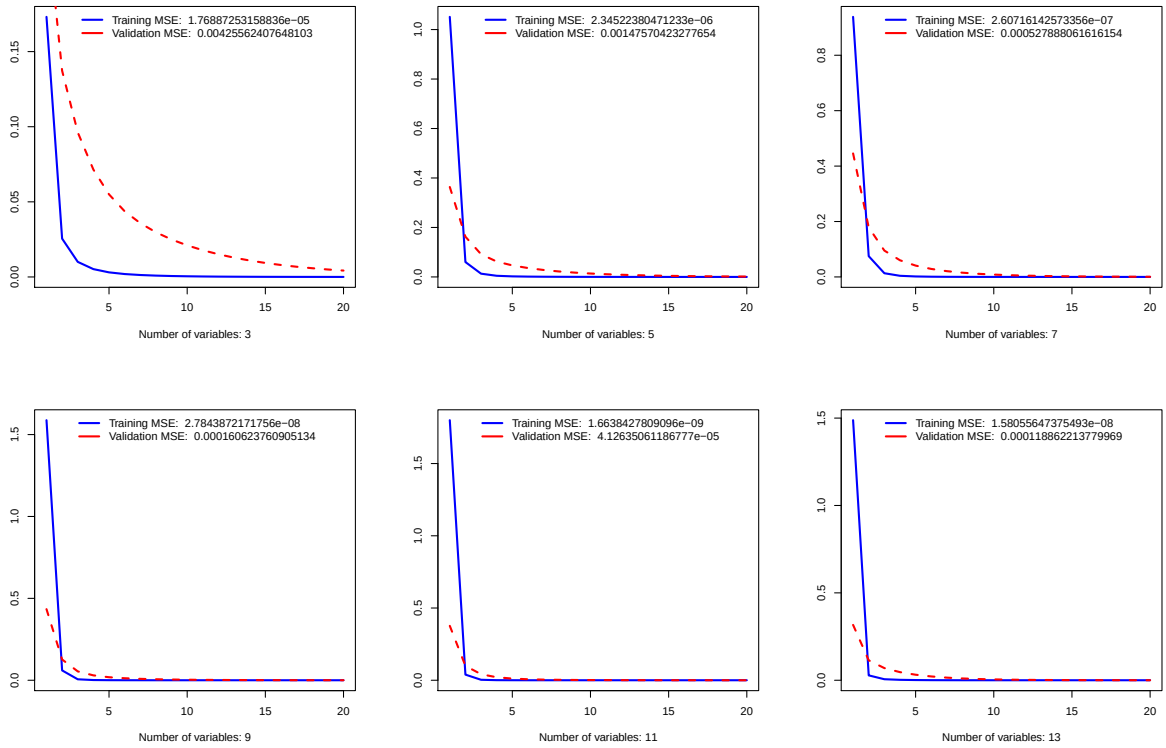


Figure 3: Training and cross-validation MSE with different number of variables (see table 3)

Particle Swarm Optimization and Neural Networks

Particle swarm optimization can be applied to solve many problems. One of them could be the training of a neural network architecture: Given a neural architecture, the problem is to find weights that minimize the mean squared error of the net. Individuals code weights of the neural network, and the fitness function corresponds to the mean squared error. According to *Kolmogorov* a multilayer perceptron can approximate any function even when the number of hidden neurons is unknown.

Obviously, a neural network with i input neurons, h hidden neurons and o output neurons it has $(i+1)h+(h+1)o$ weights and therefore, individuals of the PSO have $(i+1)h+(h+1)o$ dimensions. By considering such number, any real application with neural networks has at least 20 weights. A classical particle swarm algorithm could be applied however individuals have a high dimension and then convergence depends on the random initialization.

Figure 4 shows the learning curve of the PSO algorithm applied to a XOR neural network. This network has a $2-2-1$ architecture. It can be seen that the random initialization of individuals affect the convergence process (columns of figure). And the number of iterations (100 or 1000, at each row) achieves a lower fitness (mean squared error). Anyway, this simple example is solved with 10 individuals in the population, with dimension 9.

Another example is a binary coding neural network. An exclusive 8-bit vector coded it in a 3-bit vector. This classical problem can be solved by using a multilayer perceptron with 3 hidden neurons. Table below shows the input/output patterns of the neural network and the final weights found applying the PSO algorithm. In this case the dimension of individuals is 39 with a population of 15 individuals.

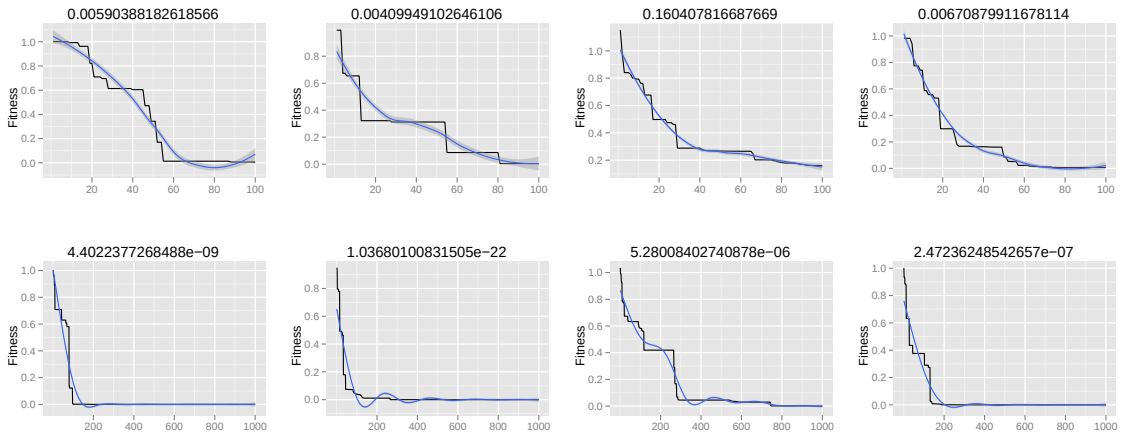


Figure 4: XOR multilayer perceptron with 2 hidden neurons and a particle swarm optimization learning using 10 individuals (individuals have 9 dimensions). Each column represents a different random initialization and each row a number of iterations (100 and 1000).

Input								Output		
1	1	1	1	1	1	1	-1	1	1	1
1	1	1	1	1	1	-1	1	1	1	-1
1	1	1	1	1	-1	1	1	1	-1	1
1	1	1	1	-1	1	1	1	1	-1	-1
1	1	1	-1	1	1	1	1	-1	1	1
1	1	-1	1	1	1	1	1	-1	1	-1
1	-1	1	1	1	1	1	1	-1	-1	1
-1	1	1	1	1	1	1	1	-1	-1	-1

Best fitness value: 5.067e−05

Best neural network weights with a $8-3-3$ architecture:

Input layer $\rightarrow$ Hidden layer

	0.1156528	1.097272	-0.946379977
	-0.3683220	25.945492	-1.703378035
	1.5325933	-9.765752	-0.636187430
	0.4830886	26.536611	0.002121948
	-1.5133460	0.790667	-0.131921926
	-1.7465932	-3.369892	0.987214704
	1.0519552	-4.920479	0.005404300
	0.3397246	-1.924014	3.273439670
Bias:	0.7092471	-5.304714	-0.331283300

Hidden layer $\rightarrow$ Output layer

	1.261584	-4.759320	0.9853185
	148.541038	-1.054461	-3.1161444
	-162.388447	-2.017318	-3.4691962
Bias:	0.090192	1.207254	1.9260548

These two neural examples have shown that the PSO can be successfully applied to the particle swarm algorithm in order to solve, in some way, the convergence of the algorithm when dealing with high dimension individuals.

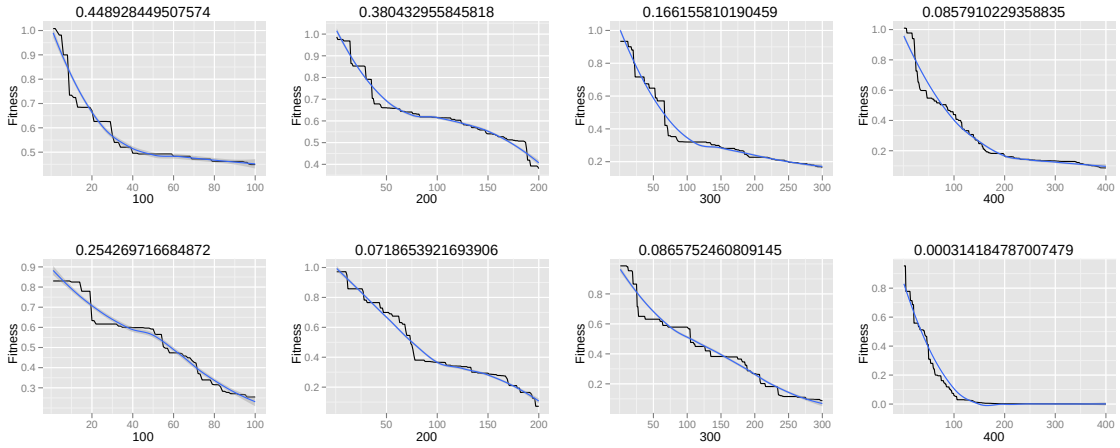


Figure 5: Binary coding neural network form 8 inputs to 3 outputs. First row is a neural network with 3 hidden neurons and second row a neural network with 5 hidden neurons. Mean squared error (fitness of the PSO) decreases as the number of iterations (100 to 400) increases.

The XOR example with dimension 9 and the binary-coding example with dimension 39 are a good starting point to combined classical neural networks with swarm intelligence.

Conclusion

The problem of nonlinear constrained optimization arises frequently in engineering. In general it does not have a deterministic solution. In the past, nonlinear optimization methods were developed and now it is a challenge to work with differentiable functions. Before gradient methods were used successfully for solving some problems. Evolutionary methods provide a new possibility for solving such problems. The PSO technique has been used successfully in optimizing real functions without restrictions, but it has been little used for problems with restrictions. This has happened mainly because there are no mechanism to incorporate restrictions on the *fitness* function. Evolutionary Computation has tried to solve the constrained optimization problem, either by bypassing nonfeasible solutions sequences, or by using a penalty function for nonfeasible sequences. Some researchers suggest to use two subfunctions of *fitness*. One helps to evaluate feasible elements and the other one evaluates the unfeasible one. In this regard, there are many criteria. Moreover, some special self adaptive functions have been designed to implement the penalty technique.

Hu and Eberhart [Hu and Eberhart 2002] presented a PSO algorithm. This algorithm bypasses nonfeasible sequences. it also creates a random initial population, in which nonfeasible sequences are bypassed until the entire population has only feasible particles. By upgrading the positions of the particles nonfeasible sequences are bypassed automatically. The cost of the technique that creates the initial populations is high; especially when it comes to problems with nonlinear constraints because then it must create an entire population of feasible individuals. In his work, Cagnina et al [Cagnina et al. 2008] proposed the following strategies for implementing the PSO into problems with restrictions: **a)** If two particles are feasible, select the one with the best *fitness*. **b)** When a particle is feasible and the other is not, the feasible one is chosen. **c)** If two particles are nonfeasible, the one with the lowest degree of nonfeasibility is selected. These strategies are applied when the particles *gbest* and *lbest* are selected.

We propose to analyze the penalty methods under E.A. perspective (Evolutionary Algorithms). The penalty methods use functions (penalty functions) that degrade the quality of the nonfeasible solution. In this way the constrained problem becomes a problem without constraints by using a modified evaluation function:

$$eval(x) = \begin{cases} \text{f(x)} & x \in \mathcal{F} \\ \text{f(x) + } \textit{penalty}\text{(x)} & eoc \end{cases} \tag{8}$$

$\mathcal{F}$ is the set created by the intersection of all sets that are the restrictions of the problem (Feasible region). The penalty is zero if no violation occurs and it is positive otherwise. The penalty function is based on the distance between a nonfeasible sequence and the feasible region $\mathcal{F}$, It also works for repairing solutions outside of the feasible region $\mathcal{F}$.

Acknowledgements

The research was supported by the Spanish Research Agency projects TRA2010-15645 and TEC2010-21303-C04-02.

Appendix A: Implementation in R

```
train <- function (iter, alpha, patrones_in, patrones_out, test, verbose) {
  patrones_in <- as.matrix(patrones_in)
  patrones_out <- as.matrix(patrones_out)
  entradas <- ncol(patrones_in)
  salidas <- length(patrones_out[1,])
  num_patrones <- round(nrow(patrones_in)*(1-test),digits=0)
  num_patrones_test <- nrow(patrones_in)
  num_pesos <- (entradas+1)*salidas
  matriz_pesos_auxiliar <- matrix(runif((entradas+1)*num_pesos),nrow=(entradas+1), ncol=num_pesos)
  matriz_pesos_principal <- matrix(runif((entradas+1)*salidas),nrow=(entradas+1),ncol=salidas)
  mse <- (1:iter)
  mse_test <- (1:iter)
  for ( i in 1:iter) {
    mse[i] <- 0.0
    for (id_patron in 1:num_patrones) {
        patron_in <- c((patrones_in[id_patron,]),-1)
        patron_out <- c((patrones_out[id_patron,]))
        salida_red_auxiliar <- patron_in %*% matriz_pesos_auxiliar
        matriz_pesos_principal <- (matrix(salida_red_auxiliar,nrow=(entradas+1),ncol=salidas))
        salida_red_principal <- patron_in %*% matriz_pesos_principal
        error <- (salida_red_principal - patron_out)
        mse[i] <- mse[i] + sum(error*error*0.5)
        variacion_pesos <-  -alpha * (error)
        matriz_pesos_principal <- matriz_pesos_principal + t(matrix(variacion_pesos,nrow=(salidas),
          ncol=(entradas+1))) * (matrix(patron_in,entradas+1,salidas))
        vector_salida_red_auxiliar <- c((matriz_pesos_principal))
        error_auxiliar <- (salida_red_auxiliar - vector_salida_red_auxiliar)
        variacion_pesos_auxiliar <-  -alpha * error_auxiliar
        matriz_pesos_auxiliar <- matriz_pesos_auxiliar + t(matrix(variacion_pesos_auxiliar,nrow=(num_pesos),
          ncol=(entradas+1))) * (matrix(patron_in,entradas+1,num_pesos))
    }
    salida_test <- test(matriz_pesos_auxiliar, patrones_in[(num_patrones+1):num_patrones_test, ],
      (patrones_out[(num_patrones+1):num_patrones_test, ]))
    mse_test[i] <- (sum(abs(salida_test - (patrones_out[(num_patrones+1):num_patrones_test, ])))/
      (num_patrones_test-num_patrones))/salidas
    if ((verbose>0) && (i%%verbose)==0) {
        cat("Iteration␣", i, "␣-->␣MSE",(mse[i]/num_patrones)/salidas, "\t\t␣-->␣CV␣", mse_test[i], "\n")
    }
    mse[i] <- (mse[i]/num_patrones)/salidas
  }
  train <- list(mpa=matriz_pesos_auxiliar,mse=mse,mse_test=mse_test)
}

test <- function  (matriz_pesos_auxiliar,patrones_in, patrones_out) {
    patrones_in <- as.matrix(patrones_in)
    patrones_out <- as.matrix(patrones_out)
    entradas <- ncol(patrones_in)
    salidas <- length(patrones_out[1,])
    num_patrones <- nrow(patrones_in)
    num_pesos <- (entradas+1)*salidas
    salida_red <- patrones_out
    for (id_patron in 1:num_patrones) {
        patron_in <- c((patrones_in[id_patron,]),-1)
        salida_red_auxiliar <- patron_in %*% matriz_pesos_auxiliar
        matriz_pesos_principal <- (matrix(salida_red_auxiliar,nrow=(entradas+1),ncol=salidas))
        salida_red_principal <- patron_in %*% matriz_pesos_principal
        salida_red[id_patron,] <- (salida_red_principal)
    }
    test <- salida_red
}

panel.cor <- function(x, y, ...)
```

```
{
    par(usr = c(0, 1, 0, 1))
    txt <- as.character(format(cor(x, y), digits=2))
    text(0.5, 0.5, txt, cex = 1.5 * ( abs(cor(x, y))) + 2 )
}

plot_correlation <- function(patrones_in, patrones_out, network_output) {
    pairs(data.frame(patrones_in, patrones_out, network_output),upper.panel=panel.cor,
      main="Relationships between characteristics of data", col="gray", cex=0.5)
}

plot_mse <- function(mpa, title="", xlabel="", ylabel="") {
  iteraciones <- length(mpa$mse)
  plot(c(1:iteraciones),mpa$mse,lty=1,type="l",main=c(title),xlab=xlabel,ylab=ylabel, col="blue", lwd=2)
  lines(c(1:iteraciones),mpa$mse_test,lty=2, col="red", lwd=2)
  legend("top", bty="n", cex=1, legend=c(paste("Training MSE: ", mpa$mse[iteraciones]),
    paste("Validation MSE: ", mpa$mse_test[iteraciones])), col=c("blue","red"), lty=1,lwd=2)
}
```

Bibliography

Cagnina, L. C., Esquivel, S. C., and Coello, C. (2008). Solving engineering optimization problems with the simpel constrained particle swarm optimizer. *Informatica*, 32:319–326.

Cover, T. M. and Tomas, J. A. (1991). *Elements of information theory*. John Willey Sons.

Delacour, J. (1987). *Apprentissage et memoire: Une approache neurobiologique*. Masson.

Hu, X. and Eberhart, R. (2002). Solving constrained nonlinear optimization problems with particle swarm optimization. In *6th world Multiconference on Systemics, Cybernetics and Informatics.*, volume 5.

Hu, Y. H. (1996). Pattern classification with multiple classifiers. Technical report, Department of Electrical Computing Engineering. University of Wisconsin, Madison.

Hu, Y. H. and Hwang, J.-N. (2001). *Handbook of neural network signal processing VE profiling*. CRC Press.

Hu, Y. H., Tompkins, W. J., Urrusti, J. L., and Afonso, V. X. (1994). Applications of artificial neural networks for ecg. *Journal of electrocardiology*, 26:66–73.

Katagiri, S. (2000). *Handbook of neural networks for speech processing*. Artech House.

Authors' Information

Nuria Gómez Blas *- Dept. Organización y Estructura de la Información, Escuela Univesitaria de Informática, Universidad Politécnica de Madrid, Crta. de Valencia km. 7, 28031 Madrid, Spain; e-mail: ngomez@eui.upm.es*

Major Fields of Scientific Research: Bio-inspired Algorithms, Natural Computing

Luis Fernando de Mingo López *- Dept. Organización y Estructura de la Información, Escuela Univesitaria de Informática, Universidad Politécnica de Madrid, Crta. de Valencia km. 7, 28031 Madrid, Spain; e-mail: lfmingo@eui.upm.es*

Major Fields of Scientific Research: Artificial Intelligence, Social Intelligence

Alberto Arteta *- Dept. Tecnología Fotónica, ETSI Telecomunicación, Universidad Politécnica de Madrid, Avenida Complutense 30, Ciudad Universitaria, 28040 Madrid, Spain; e-mail: m.muriel@upm.es*

Major Fields of Scientific Research: Theoretical Computer Science, Microwave Photonics

SPREADING THE MOORE - PENROSE PSEUDO INVERSE ON MATRICES EUCLIDEAN SPACES: THEORY AND APPLICATIONS

Volodymyr Donchenko, Fedir Skotarenko

Abstract*: In the paper the development of operating technique for matrices Euclidean spaces is represented. Particularly, within such development transfer of linear operators' technique with preserving properties of closely correspondence to natural subspaces is represented. Also - spectral results, SVD – and Moore – Penrose Pseudo Inverse – technique, theory of orthogonal projectors and grouping operators. Besides, solution of the linear discrimination problem in Euclidean spaces of matrices is represented in the paper. Realization the program of empowering the operating technique in matrices Euclidean space made possible on the basis of putting in circulation of so-called "cortege operators" and, correspondingly - "cortege operations".*

Keywords*: Feature vectors, matrix corteges operators, Single Valued Decomposition for cortege linear operators, linear discrimination.*

ACM Classification Keywords*: G.2.m. Discrete mathematics: miscellaneous, G.2.1 Combinatorics; G.3 Probability and statistics, G.1.6. Numerical analysis, I.5.1.Pattern Recognition; H.1.m. Models and Principles: miscellaneous.*

Introduction

Grouping information problem (GIP), which is fundamental one in applications, appears in two main forms namely: the problem of recovering the function, represented by their observations, and the problem of clustering, classification and pattern recognition. Examples of approaches in the field are represented perfectly in [Kohonen, 2001], [Vapnik, 1998], [Haykin, 2001], [Friedman, Kandel, 2000], [Berry, 2004]. It is opportune to notice, that math modeling is the representation of an object structure by the means of mathematical structuring. A math structure after Georg Cantor is a set plus "ties" between its elements. Only four fundamental types of "ties" (with its combination as fifth one) exist: relations, operations, functions and collections of subsets. Thus, the mathematical description of the object (mathematical modeling) cannot be anything other than representing the object structure by the means of mathematical structuring. It refers fully to so call "complex system". A "complex system" should be understanding and, correspondingly, determined, as an objects with complex structure (complex "ties"). When reading attentively manuals by the theme (see, for example, [Yeates, Wakefield, 2004], [Forster, Hölzl, 2004]) one could find correspondent allusions. "Structure" understanding of the object is reasonable in determining of a "complex systems" instead of defining it as the "objects, consisting of numerous parts, functioning as an organic whole".

In the essence, math modeling is representing by math "parts plus ties" "parts plus ties" of the object in applied field.

It is commonly used approach for designing objects - representative to construct them as an finite ordered collection of characteristics: quantitative (numerical) or qualitative (non numerical). Such ordered collection of characteristics is determined by term cortege in math. Cortege is called vector when its components are numerical. In the function recovering problem objects - representatives are vectors and functions are used as a rule to design correspond mathematical "ties". In clustering and classification problem the collection may be both

qualitative and quantitative. In last case correspond collection is called feature vector. It is reasonable to note that term “vector” means more, than simply ordered numerical collection. It means that curtain standard math “ties” are applicable to them. These “ties” are adjectives of the math structure called Euclidean space denoted by R^n.

Euclidean spaces, namely R^n, are the first among math structures rich on ties. Already in the very definition Euclidean spaces offer a range of structural links: from operations to scalar product, norms and limit transitions in various form. Besides, these ones possess highly developed technique of linear operators: with spectral theory, Singular Valued Decomposition (SVD) and Moore – Penrose technique in Euclidean spaces of real valued vectors (in R^n). Regarding uses of R^n we recall linear regression and classification or clasterization problem with necessity of designing an appropriate feature vectors. But there is an urgent need in the application to expand the range of "representatives" of real objects with preserving the wealth structural relationships inherent Euclidean spaces. Matrices Euclidean spaces are a natural extension of the class of Euclidean spaces. Speech recognition with the spectrograms as the representatives and the images in the problem of image processing and recognition are the natural object areas with “matrices representatives”. So, of utmost importance is developing mathematical modeling tools and, in particular, the problems of grouping information problem is the transfer on matrices Euclidean spaces the wealth of technical capabilities of R^n. As to technique designing for the spreading of Euclidean spaces as “environmental” math structure first steps have been made for example, by [Donchenko, 2011], [Donchenko, Zinko, Skotarenko, 2012].

Just the belonging to the base math structure (Euclidean space) determines advantages of the “vectors” against “corteges” as ordered finite collection of elements. It is noteworthy to say, that real-valued vectors as a variant of Euclidean space is not unique. A space of all matrixes of a fixed dimension is alternative example. As it was mentioned above, the choice of R^n as “environmental” space is determined by perfect technique developed for manipulation with vectors. These include classical matrix methods and classical linear algebra methods. SVD-technique and methods of Generalized or Pseudo Inverse according Moore – Penrose are comparatively new elements of linear matrix algebra technique [Nashed, 1978] (see, also, [Albert, 1972], [Ben-Israel, Greville, 2002]). Outstanding impacts and achievements in this area are due to N. F. Kirichenko (especially, [Kirichenko, 1997], [Kirichenko, 1997], see also [Kirichenko, Lepeha, 2002]). Greville’s formulas: forward and inverse - for pseudo inverse matrixes, formulas of analytical representation for disturbances of pseudo inverse, - are among them. Additional results in the theme as to further development of the technique and correspondent applications one can find in [Kirichenko, Lepeha, 2001], [Donchenko, Kirichenko, Serbaev, 2004], [Kirichenko, Crak, Polishuk, 2004], [Kirichenko, Donchenko, Serbaev,2005], [Kirichenko, Donchenko, 2005] [Donchenko, Kirichenko, Krivonos, 2007], [Kirichenko, Donchenko, 2007], [Kirichenko, Krivonos, Lepeha, 2007], [Kirichenko, Donchenko, Krivonos, Crak, Kulyas, 2009].

As to the choice of the collection (design of cortege or vector) it is necessary to note, that good “feature” selection (components for feature vector or cortege or an arguments for correspond functions) determines largely the efficiency of the problem solution.

In the paper the development of operating technique for matrices of Euclidean spaces is represented. Particularly, transfer on these spaces linear operators technique with preserving of close correspondence with “natural subspaces”; spectral results; SVD – and Moore – Penrose Pseudo Inverse – technique M-P PdI); the theory of orthogonal projection and grouping operators. Also, solution of the linear discrimination problem in Euclidean spaces of matrices is represented below in the paper. Realization the program of empowering the operating technique in matrices Euclidean space made possible on the basis of putting in circulation of so-called "cortege operators” and, correspondingly - "cortege operations”.

Matrixes spaces and cortege operators

Theorem 1. (Enhanced spectral theorem) For an arbitrary linear operator between a pair of Euclidean spaces $(E_i,(,)_i),i=1,2:\wp_E:E_1\to E_2$, the collection of singularities $(v_i,\lambda_1^2),\ (u_i,\lambda_1^2)i=\overline{1,r},\ r=rank\wp_E$ exists for the operators $\wp_E^*\wp:E_1\to E_1,\ \wp\,\wp_E^*:E_2\to E_2$ correspondingly, with a common for both operators $\wp_E^*\wp,\ \wp\,\wp_E^*$ set of Eigen values $\lambda_1^2,i=\overline{1,r}:\ \lambda_{i-1}\geq\lambda_i>0,\ \overline{i=2,r}$ such that $\wp_E x=\sum_{i=1}^{r}\lambda_i u_i(v_i,x)_1,\quad \wp_E^* y=\sum_{i=1}^{r}\lambda_i v_i(u_i,y)_2$.

Besides, the following relations take place:

$$u_i=\lambda_1^{-1}\wp v_i,i=\overline{1,r},\ v_i=\lambda_1^{-1}\wp_E^* u_i,i=\overline{1,r}.$$

Matrixes spaces and SVD – technique for cortege operator

We denote by $R^{(m\times n),K}$ - Euclidean space of all matrixes K-corteges from $m\times n$ matrixes: $\alpha=(A_1\vdots...\vdots A_K)\in R^{(m\times n),K}$ with a "natural" component wise trace inner product:

$$(\alpha,\beta)_{cort}=\sum_{k=1}^{K}(A_k,B_k)_{tr}=\sum_{k=1}^{K}trA_k^TB_k,\ \alpha=(A_1\vdots...\vdots A_K),\beta=(B_1\vdots...\vdots B_K)\in R^{(m\times n),K}.$$

1. We also denote by $\wp_\alpha:R^K\to R^{m\times n}$ a linear operator between the Euclidean space determined by the relation:

$$\wp_\alpha y=\sum_{k=1}^{K}y_kA_k,\alpha=(A_1\vdots...\vdots A_K)\in R^{(m\times n),K},y=\begin{pmatrix}y_1\\ \cdots\\ y_K\end{pmatrix}\in R^K \tag{1}$$

2. **Theorem 2.** Range $\Re(\wp_\alpha)=L_{\wp_\alpha}$, which is linear subspace of R^{m*n}, is the subspace spanned on the components of cortege $\alpha=(A_1\vdots...\vdots A_K)\in R^{(m\times n),K}$, that determines $\wp_\alpha$: $\Re(\wp_\alpha)=L_{\wp_\alpha}=L(A_1,\ldots,A_K)$.

3. **Theorem 3.** Conjugate for the operator, determined by (1) is a linear operator, which, obviously, acts in the opposite direction: $\wp_\alpha^*:R^{m\times n}\to R^K$, and defined as:

$$\wp_\alpha^*X=\begin{pmatrix}(A_1,X)\\ \cdots\\ (A_K,X)\end{pmatrix}=\begin{pmatrix}trX^TA_1\\ \cdots\\ trX^TA_K\end{pmatrix} \tag{2}$$

4. **Theorem 4.** A product of two operators $\wp_\alpha^*\wp_\alpha:R^K\to R^K$ is a linear operator, defined in the standard way by the matrix from the next equality:

$$\wp_\alpha^*\wp_\alpha=\begin{pmatrix}trA_1^TA_1,...,trA_1^TA_K\\ \cdots\\ trA_K^TA_1,...,trA_K^TA_K\end{pmatrix} \tag{3}$$

Remark. Matrix defined by (3) is the Gram' matrix for the matrixes - cortege components $A_k, k = \overline{1,K}$ of Euclidean space $R^{m\times n}$.

5. Singular value decomposition (SVD) for a matrix (3) is obvious. That matrix is a classical $K\times K$ matrix: symmetric and positively semi-definite. This SVD is defined by a collection of singularities $(v_i, \lambda_i^2), i, j = \overline{1,r}$:

$$|| v_i ||= 1, v_i \perp v_j, i \neq j; i, j = \overline{1,r}; \lambda_1 > \lambda_2 > ... > \lambda_r > 0,\ \wp_\alpha^* \wp_\alpha v_i = \lambda_i^2 v_i, i = \overline{1,r}.$$

The operator $\wp_\alpha^* \wp_\alpha$ by itself is determined by the relation $\wp_\alpha^* \wp_\alpha = \sum_{i=1}^{r} \lambda_i^2 v_i v_i^T = \sum_{i=1}^{r} \lambda_i^2 v_i (v_i, \cdot)$.

Each of the row - vectors $v_i^T, i = \overline{1,r}$ will be written by their components:

$$v_i^T = (v_{i1}, ..., v_{iK}), i = \overline{1,r},\ r \text{ - is the rank of } A_k, k = \overline{1,K} \text{ in linear space is } R^{m\times n}.$$

6. **Theorem 5.** Matrices $U_i \in R^{m\times n} : U_i = \frac{1}{\lambda_i} \wp_\alpha v_i = \frac{1}{\lambda_i} \sum_{k=1}^{K} A_k v_{ik}, i = \overline{1,r}$, defined by the singularities $(v_i, \lambda_i^2), i = \overline{1,r}$ of the operator $\wp_\alpha^* \wp_\alpha$ are elements of a complete collection of singularities $(U_i, \lambda_i^2), i = \overline{1,r}$ of the operator. $\wp_\alpha^* : R^K \to R^{m\times n}$.

Proof. The result directly follows from the Theorem 1 and standard relationships between singularities of the $\wp_\alpha^* \wp_\alpha$, $\wp_\alpha \wp_\alpha^*$ operators.

7. **Theorem 6**. (Singular Value Decomposition (SVD) for cortege operator) Singularity of two operators $\wp_\alpha^* \wp_\alpha, \wp_\alpha \wp_\alpha^*$, obviously determine the singular value decomposition of operators $\wp_\alpha$, $\wp_\alpha^*$:

$$\wp_\alpha y = \sum_{i=1}^{r} \lambda_i U_i v_i^T y = \sum_{i=1}^{r} \lambda_i U_i (v_i, y) \in R^K, \wp_\alpha^* X = \sum_{i=1}^{r} \lambda_i v_i (U_i, X)_{tr}, X \in R^{m\times n}.$$

8. **Corollary**. A variant is a SVD for the operator $\wp_\alpha$ is represented by the next relation:

$$\wp_\alpha = \sum_{k=1}^{r} \lambda_k U_k v_k^T = \sum_{k=1}^{r} (\wp_\alpha v_k) v_k^T.$$

Pseudo Inverse Technique for matrixes Euclidean spaces

Fundamental operator of Moore - Penrose Pseudo Inverse (M-P PdI) theory is PdI operator: for in the case under consideration. This operator is proposed to be determined by SVD-representation.

Theorem 7. The PdI operators for $\wp_\alpha$, $\wp_\alpha^*$ are determined, correspondingly, by the relations

$$\wp_\alpha^+ X = \sum_{k=1}^{r} \lambda^{-1} v_k (U_k, X)_{tr} = \sum_{k=1}^{r} \lambda^{-2} v_k (\wp_\alpha v_k, X)_{tr}, \forall X \in R^{m\times n}, (\wp_\alpha^*)^+ y = \sum_{i=1}^{r} \lambda^{-1} U_i v_i^T y = \sum_{i=1}^{r} \lambda^{-1} U_i (v_i y), \forall y \in R^K.$$

9. **Principal operators PdI theory for a cortege operators: basic OP-operators**.

Two pairs of operators are principal importance in classical M-P PdI theory namely, these are: a) two pares of orthogonal projector operators(OP-operators): on the ranges of operators $\wp_\alpha^*, \wp_\alpha$ correspondingly and on the kernels of these operators; b) grouping operators(G-operators) [Kirichenko, Lepeha, 2002].

As it was mentioned above OP – operators project on the subspaces $\Re(\wp_\alpha)=L_{\wp_\alpha},\Re(\wp_\alpha^*)=L_{\wp_\alpha^*},\ \mathrm{Ker}\,\wp_\alpha=\Re^\perp(\wp_\alpha^*)=L^\perp_{\wp_\alpha^*},\mathrm{Ker}(\wp_\alpha^*)=\Re^\perp(\wp_\alpha)$ - are determined, in the essence, by the orthogonal projections on two ranges: $\wp_\alpha,\wp_\alpha^*$ correspondingly. These orthogonal projections will be designated in one of two equivalent ways:

$$P(\wp_\alpha^*)\equiv P_{L_{\wp_\alpha}}=P_{L(A_1,\ldots,A_K)},L_{L_{\wp_\alpha}}\subseteq R^{m\times n},P(\wp_\alpha)\equiv P_{L_{\wp_\alpha^*}},L_{\wp_\alpha^*}\subseteq R^K.$$

Two OP- operators: on the kernels of $\wp_\alpha,\wp_\alpha^*$ – being orthogonal complements to the correspond ranges are the compliments to identity operators of ranges OP- operators: as $Ker\wp_\alpha^*=L^\perp_{\wp_\alpha}\subseteq R^{m\times n},Ker\wp_\alpha=L^\perp_{\wp_\alpha^*}\subseteq R^K$.

These OP –operators on correspondent kernels we denote by $Z(\wp_\alpha),Z(\wp_\alpha^*)$ correspondingly:

$$Z(\wp_\alpha)\equiv P_{L^\perp_{\wp_\alpha^*}},Z(\wp_\alpha^*)\equiv P_{L^\perp_{\wp_\alpha}},$$

Obviously:

$$Z(\wp_\alpha)\equiv E_K-P(\wp_\alpha),Z(\wp_\alpha^*)\equiv E_{m\times n}-P(\wp_\alpha^*) \tag{4}$$

$E_K,E_{m\times n}$ - identity operators in correspondent spaces.

In accordance with the general properties of PdI, the next properties are valid:

$$P(\wp_\alpha)=\wp_\alpha^+\cdot\wp_\alpha,P(\wp_\alpha^*)=\left(\wp_\alpha^*\right)^+\cdot\wp_\alpha^*=\wp_\alpha\cdot\wp_\alpha^+=\sum_{k=1}^r v_k v_k^T.$$

Correspondingly:

$$Z(\wp_\alpha)\equiv E_K-\wp_\alpha^+\cdot\wp_\alpha,\quad Z(\wp_\alpha^*)\equiv E_{m\times n}-\wp_\alpha\cdot\wp_\alpha^+=E_K-\sum_{k=1}^r v_k v_k^T.$$

10. **Basic operators PdI theory for a cortege operators: grouping operators.**

Grouping operators (G- operators) [Donchenko, Zinko, Skotarenko, 2012], designated below by $R(\wp_\alpha),\ R(\wp_\alpha^*)$, are also "paired" operators, and are determined by the relations:

$$R(\wp_\alpha)=\wp_\alpha^+\left(\wp_\alpha^+\right)^*=\left(\wp_\alpha^*\wp_\alpha\right)^+,\ R(\wp_\alpha^*)=\left(\wp_\alpha^*\right)^+\left(\left(\wp_\alpha^*\right)^+\right)^*=\left(\wp_\alpha^+\right)^*\wp_\alpha^+=\left(\wp_\alpha\wp_\alpha^*\right)^+.$$

11. **Theorem 8.** G-operators for the cortege operators $\wp_\alpha,\ \wp_\alpha^*$ can be represented by the next expression:

$$R(\wp_\alpha^*)X=\sum_{i=1}^r\lambda_i^{-2}U_i(U_i,X)_{tr}=\sum_{i=1}^r\lambda_i^{-2}U_i trU_i^TX=\sum_{i=1}^r\lambda_i^{-2}U_i trX^TU_i,i=\overline{1,r}$$

with

$$U_i=\frac{1}{\lambda_i}\sum_{k=1}^K v_{ik}A_k,i=\overline{1,r};$$

besides, the quadratic form $(X,R(\wp_\alpha^*)X)_{tr}$ is determined by the relation:

$$(X,R(\wp_\alpha^*)X)_{tr}=\sum_{k=1}^r\lambda_k^{-2}(U_k,X)_{tr}^2,$$

12. **Theorem 9**. Quadratic form $(X,R(\wp_\alpha^*)X)_{tr}$ may be written as:

$$(X,R(\wp_\alpha^*)X)_{tr}=\sum_{i=1}^{r}\lambda_i^{-4}v_i^T\begin{pmatrix} trA_1^TXtrA_1^TX & trA_2^TXtrA_2^TX & \cdots & trA_1^TXtrA_K^TX \\ trA_2^TXtrA_1^TX & trA_2^TXtrA_2^TX & \cdots & trA_2^TXtrA_K^TX \\ \cdots & \cdots & \cdots & \cdots \\ trA_K^TXtrA_1^TX & trA_K^TXtrA_1^TX & \cdots & trA_K^TXtrA_1^TX \end{pmatrix}v_i=$$

$$=\sum_{i=1}^{r}\lambda_i^{-4}\left\{v_i^T\begin{pmatrix} trA_1^TX \\ \cdots \\ trA_K^TX \end{pmatrix}\right\}^2=\sum_{i=1}^{r}\lambda_i^{-4}\left\{v_i^T\wp_\alpha^*X\right\}^2$$

Importance of G-operators is determined by their properties, represented by the next two theorems.

13. **Theorem 10.** For any $A_i, i=\overline{1,K}$ of $\alpha=(A_1\vdots\ldots\vdots A_K)\in R^{(m\times n),K}$ the next inequalities are fulfilled:

$$(A_i,R(\wp_\alpha^*)A_i)_{tr}\le r, i=\overline{1,K}, r=rank\wp_\alpha .$$

14. **Theorem 11**. For any $A_i, i=\overline{1,K}$ of $\alpha=(A_1\vdots\ldots\vdots A_K)\in R^{(m\times n),K}$ the next inequalities are fulfilled:

$$(A_i,R(\wp_\alpha^*)A_i)_{tr}\le r_{\min}\le r, i=\overline{1,K}, r=rank\wp_\alpha, r_{\min}=\max_{i=\overline{1,n}}(A_i,R(\wp_\alpha^*)A_i)_{tr}\le r_{\min}\le r, i=\overline{1,K}, r=rank\wp_\alpha .$$

Comment to the theorems 10, 11. These theorems give the minimal grouping ellipsoids for the matrixes $A_i, i=\overline{1,K}$. In order to build it one only has to construct cortege operator $\wp_\alpha$ by the cortege $\alpha=(A_1\vdots\ldots\vdots A_K)\in R^{(m\times n),K}$.

Linear discrimination problem in matrix Euclidean space $R^{m\times n}$

Linear discrimination problem (LDP) is the problem of separating of two classes, represented by correspond learning sample by appropriate hyper plane, For Euclidean spaces R^m this problem was formulated and successfully solved on the base of PdI technique in [Kirichenko, Lepeha, 2002]. This problem is formulated and solved below on base of PdI technique developed for matrices on the base of cortege operators represented early in this article.

The wording of the problem.

Let $X_1,\ldots,X_K\in R^{m\times n}$ united collection of matrixes from learning sample, represented two classes:

$$X_j\in KI_1, j\in J_1, X_j\in KI_2, j\in J_2:\ J_1\cap J_2=\varnothing, J_1\cup J_2=\{1,2,\ldots,K\} .$$

It is necessary to find $\Delta>0$ and design linear functional $A: R^{m\times n}\to R^1$ in such a way that

$$(A,X_j)_{tr}>\Delta, j\in J_1, (A,X_j)_{tr}<-\Delta, j\in J_2$$

We will designate by $\Omega(\Delta)$ the domain of real-valued vector $y^T=(y_1,\ldots,y_K)$ from R^K with the components which satisfy to the next constraints: $y_j>\Delta, j\in J_1, y_j<-\Delta, j\in J_2$

LDP solution.

LGP solution for matrices spaces is because the vector of "discriminating" values $\left((A,X_1)_{tr},...,(A,X_K)_{tr}\right)^T$ of discriminating linear forms $(A,X)_{tr}$ determines the value of conjugate operator to cortege operator $\wp_\chi : \chi = (\mathrm{X}_1\vdots,...,\vdots\mathrm{X}_K)$ on argument A.

The next theorem then is valid.

Theorem 12. LDP is equivalent of linear equation problem $\wp_\chi^* X = y$ for cortege operator $\wp_\chi, \chi = (\mathrm{X}_1\vdots,...,\vdots\mathrm{X}_K)$ and $(A,X_j) > \Delta, j \in J_1, (A,X_j) < -\Delta, j \in J_2$.

Proof. Indeed, fulfilling of (1) means that vector

$$y : \left((A,X_1),...,(A,X_K)\right) \equiv y^T \tag{5}$$

belongs to $\Omega(\Delta)$.

Theorem 13. Allows to conclude, that (5) is equivalent to solvability the equation $\wp_\chi^* \mathrm{A} = \mathrm{y}, \chi = (\mathrm{X}_1\vdots,...,\vdots\mathrm{X}_K), \mathrm{y} \in \Omega(\Delta)$. And, thus, the proof of the theorem is finished.

Theorem 14. LDP is solvable if there exists $y_* \in \Omega(\Delta) \subseteq R^K$ and correspond solution is determined by the equality

$$A = \wp_\chi^{*+} y_* \tag{6}$$

Corollary 1. LDP is solvable if there exists $y_* \in \Omega(\Delta) \subseteq R^K$ for which the next condition is fulfilled

$$y_*^T Z(\wp_\alpha) y_* = y_*^T (E_K - \sum_{k=1}^{r} v_k v_k^T) y_* = 0$$

and correspond solution is still determined by (6).

Theorem 15. LDP is equivalent to quadratic optimization problem for quadratic form

$$y^T Z(\wp_\alpha) y = y^T (E_K - \sum_{k=1}^{r} v_k v_k^T) y$$

In domain $\Omega(\Delta) \subseteq R^K$. If the solution y_* of the optimization problem in the domain gives minimum, that equal zero, then matrix A - LDP solution, is determined by equality

$$A = \wp_\chi^{*+} y_*.$$

Linear Discrimination Problem: algorithm

When saying about the algorithm of matrix LDP problem with united for two classes collection of matrices $X_1,...,\mathrm{X}_K \in R^{\mathrm{m}\times n}$ then it starts with the first step".

1-st step: calculation of Gram' matrix for collection of matrices $X_1,...,\mathrm{X}_K \in R^{\mathrm{m}\times n}$:

$$F = \begin{pmatrix} trA_1^T A_1,...,trA_1^T A_K \\ \cdots \\ trA_K^T A_1,...,trA_K^T A_K \end{pmatrix} = \begin{pmatrix} (A_1,A_1)_{tr},...,(A_1,A_K)_{tr} \\ \cdots \\ (A_K,A_1)_{tr},...,(A_K,A_K)_{tr} \end{pmatrix}.$$

And then:

2-nd step: computing the singularities $(v_k, \lambda_k^2), \lambda_k^2 > 0, \ k = \overline{1,r}, \quad r = rankF$;

3-d step: calculating the matrixes $E_K - \sum_{k=1}^{r} v_k v_k^T$ of quadratic form;

4-th step: calculating the minimum of quadratic form $y^T \left(E_K - \sum_{k=1}^{r} v_k v_k^T \right) y$ in domain $\Omega(\Delta) \subseteq R^K$ (numerical methods) and correspondent argument y_*;

4-th step: verification the conditions of zero value of minimum: if $y_*^T \left(E_K - \sum_{k=1}^{r} v_k v_k^T \right) y_* = 0$:

5-th step:

- If condition is fulfilled: $y_*^T \left(E_K - \sum_{k=1}^{r} v_k v_k^T \right) y_* = 0$, then computing linear form A of LDP-solution accordingly to relation $A = \wp_{\chi}^{*+} y_*$;
- If condition is not fulfilled, then LDP is unsolvable.

Matrix Linear Discrimination Problem in clasterization

Theorem 11 can be applied to problems of matrix clustering just in the same way as in [Donchenko, Krak, Krivonos, 2007] it has been done for R^m.

Conclusion

Conception of enriching the standard considering the ”representatives” in Applied Math to be the feature vectors: elements from Euclidean space, - has been further developed in the paper (see, also, [Donchenko, Zinko, Skotarenko, 2012]). Using matrixes as the “representatives” of the real objects is main idea of the conception. This mean, that matrix instead vector represents all principal features of the objects in applied fields. Support of this concept requires the development of technologies handling with matrixes similar techniques operating with vectors from Euclidean spaces. SVD-technique as well as PIMP - technique are the priority among them. The results of such type are represented in the paper. These results demanded a generalization of matrix algebra and transforming it in algebra of matrix and vector cortege as well as definition and using the linear cortege operator. Correspond results are represented in the paper of the authors [Donchenko, Zinko, Skotarenko, 2012]. Using that handling technique for matrix features (“matrix feature vectors”) make it possible to put and fully solute the Linear Discrimination problem for two collection of matrixes. Corresponded solution uses standard SVD and PIMP for Gramian matrix of united collections and solution of quadratic optimization in a domain of appropriate. Thus, the development of matrix technique manages to reduce to existing technique for real valued vectors. Solution of Linear Discrimination Problem for matrixes is similar to corresponded result for real-valued vectors in [Kirichenko, Krivonos, Lepeha, 2007] or [Donchenko, Krak, Krivonos, 2012]. The two obvious application areas are worth mentioning within the context of the application of these results. These are: speech recognition and image processing. Matrixes naturally represent the objects under consideration, namely, spectrograms and digital images.

Bibliography

[Albert, 1972] Albert A. E. Regression and the Moore-Penrose pseudoinverse.-Academic Press.- 1972.- 180 стор.

[Ben-Israel, Greville,2002] Adi Ben-Israel, Thomas N. E. Greville Generalized Inverses. Theory and Applications.- Second Edition.- Springer-Verlag New York, Inc. -2003.-420 p

[Berry, 2004] Michael W. Berry Editor Survey of Text Mining: Clustering, Classification, and Retrieval Springer-Verlag New York, Inc. 2004. 244 p.

[Donchenko, Kirichenko, Krivonos, 2007] Donchenko V., Kirichenko M., Krivonos Yu. Generalizing of neural nets: Functional nets of special type// International Journal Information Theories and Applications. – N 3. –V.14. – 2007. – P. 259-266.

[Donchenko, Kirichenko, Serbaev, 2004] Donchenko V.S. Kirichenko M.F., Serbaev D.P. Recursive regression transformation and dynamical systems// Proceedings: of the Seventh International Conference "Computer Data analysis and Modeling: robustness and computer intensive methods". – V.1. – September 6-10, 2004. – Minsk. – P. 147-151

[Donchenko, Zinko, Skotarenko, 2012] Donchenko V, Zinko T, Skotarenko F. "Feature Vectors" in Grouping Information Problem in Applied Mathematics: Vectors and Matrixes. - Vitalii Velichko, Alexey Voloshin, Krassimir Markov(ed.)- Problems of Computer Intellectualization.- Institute of Cybernetics NASU, ITHEA.-Kyiv, Ukraine -Sofia, Bulgaria.- 2012.- P.111-124.

[Forster, Hölzl, 2004] John Forster and Verner Hölzl Applied Evolutionary economics and complex systems.-2004-- 293 p.

[Friedman, Kandel, 2000] Menahem Friedman Abraham Kandel Introduction to pattern recognition statistical, structural, neural and Fuzzy logic approaches world scientific publishing co. Pte. Ltd.-reprinted 2000.-329 p.

[Haykin, 2001] Haykin S. Neural Networks, A Comprehensive Foundation. Prentice Hall, Upper Saddle River, New Jersey 07458. – 1999.– 842 p

[Ivachnenko, 1995] Ivachnenko A.G and Muller J. A. Present state and new problems of further GMDH development.-SAMS, 1995.- pp.3-16.

[Kirichenko, 1997] N. F. Kirichenko. Analytical Representation of Perturbation of Pseudoinverse Matrices. Cybernetics and Systems Analysis. Vol.33, Number 2, P. 230-239, March-April 1997.

[Kirichenko, Donchenko, Serbaev, 2005] Kirichenko N. F., Donchenko V.S., Serbaev D.P. Nonlinear recursive nonlinear Transfomations: Dynamic sestems and Optimizations.//Cybernetics and Systen Ansalysis.– V.41, №3.– May 2005.– p.364-373.

[Kohonen, 2001] Kohonen T. Self-Organizing Maps -3-d ed. - Tokyo: Springer, 2001, 501 p.

[Nashed, 1976] Nashed M. Zuhair, Votruba G.F., editors. A Unified Operator Theory of Generalized Inverse.//Proceedings of an Advanced Seminar Sponsored by the Mathematical Research Center, The University of Wisconsin, Madison, October 8-10, 1973. – New York, Academic Press, 1976.–1200 p.

[Vapnik, 1998] Vapnik, V. N. Statistical Learning Theory. New York: Wiley, 1998

[Yeates, Wakefield, 2004] Donald Yeates, Tony Wakefield Systems Analysis And Design. Pearson Education. 2004. - 499 p.

Authors' Information

Volodymyr Donchenko *– Professor, Taras Shevchenko National University of Kyiv, Volodymyrs'ka street, Kyiv, 03680, Ukraine; e-mail: voldon@bigmir.net.*

Fedir Skotarenko *- Post - graduate student, Taras Shevchenko National University of Kyiv.*

TOWARDS A SEMANTIC CATALOG OF SIMILARITY MEASURES

Alfredo Sánchez-Alberca, Rafael Lahoz-Beltra and Juan Castellanos-Peñuela

Abstract: *The tasks that encloses the Semantic Web often requires semantic comparisons between classes, properties, relations and instances, that mainly consist in semantic similarity. As a consequence, the number of similarity functions used in this field has greatly increased in the last years. However, the selection of the best similarity functions to use in each case is a difficult task usually guided by heuristics and intuition. To overcome this problem, in this paper we propose a taxonomy of similarity functions as a first step to a semantic catalog of similarity functions. This taxonomy is induced by another taxonomy of data types. Our taxonomy allows selecting the similarity functions applicable to a pair of entities based on the data type used to represent them. It also eases the composition of functions according the composition of data types, and supports the inheritance of applicable functions from the more abstract to the more specific data types. As a consequence, it facilitates the application of similarity functions in an automated way. Finally, we also populate this taxonomy with some of the most popular similarity functions used in Semantic Web.*

Keywords: *Semantic Web, semantic comparison, similarity measures*

ACM Classification Keywords*: 1.5.4 Pattern recognition applications. I.2.4 Knowledge Representation Formalisms and Methods.*

Introduction

Similarities play a key role in the process of semantic comparison of entities, i.e. classes, properties, relations and instances, in the Semantic Web. Each entity is described by a set of attributes of different types. In the case of ontologies any concept or class has an identifier (IRI), a name (string), a description in natural language (string) and, most likely, attributes whose range is a basic data type e.g. Boolean, integer, float, date or string, a composed data type e.g. set, list, etc., or other classes. However, entities are also described by their relationships with other entities. For example, a concept can be part of other concept, i.e. meronymy/holonymy relationship , or a specialization, i.e. hypernymy/hyponymy relationship, or other “ad hoc” relation. In addition, in populated ontologies or databases, concepts have also extensional definitions through their instances. So, to compare two entities we need to decide which properties should be taken into account as well as choose the best similarity functions to use for comparing them.

To illustrate the problem with an example, we are developping an ontology of spanish wines, as Spain is one of the majors producers of wine. Our intend is to compare wines, but if we want to compare two wines, first we need to establish which are the main wine characteristics to compare (grape variety, alcohol content, ph, colour, bouquet, flavor, price, etc.), and then how to compare them. For instance, if we are interested in tasting what similarity function should we use? Which functions could we really use? Furthermore, once we have measured the similarity between these characteristics, how will this result be used to calculate the similarity between the wines?

Reviewing the literature on similarity functions used for comparing entities in ontologies [Euzenat and Shvaiko, 2007; Cohen et al., 2003; Koudas et al., 2006; Cross and Sudkamp, 2002], we will note that there are a lot of similarity functions and their formalization sometimes is unclear for non-mathematicians. By and large, there are no rules that point out the similarity functions to use in each case and consequently the choice of an appropriate similarity function is not an easy task, being guided many times by heuristics or intuition. Even, in many occasions the choice is strongly conditioned by the understanding of the similarity function more than by its adequacy.

In order to overcome these problems, in this work we propose a new taxonomy of similarity functions that could help to guide the semantic comparison of entities. This classification is built according to the data type of the ontology or schema entities compared. The taxonomy defines a hierarchy of similarity functions based on the `is_a` relation for data types. In this way, similarity functions for a general data type could be applied to all its more specific descendant data types. It also eases the composition of functions according the composition of data types. This could be very helpful in order to automate the calculation and aggregation of multiple similarity measures from different similarity functions, especially when it is not clear which similarity function performs better.

At the same time, we have populated the taxonomy with the most used similarity functions in the Semantic Web. Based on this catalog, we also have developed a Java library of similarity functions called SIMEON (Similarity Measures for Ontologies).

The structure of the paper is as follows: First we review the notion of similarity and distance; next, we introduce the structure of the taxonomy of similarity functions; then we present some examples of similarity functions for simple and compound data types respectively, showing their classification in the taxonomy and illustrating their use; and, finally we draw the conclusions and point out future work.

Similarity functions

The similarity notion has been studied deeply in psychology. In this field, several theories has been proposed for measuring similarities, for example, see [Tversky, 1977] and [Goldstone, 1999]. This section deals with the definitions of similarity and distance functions proposed in the abovementioned theories. These definitions have been widely adopted by the artificial intelligence community.

Definition 1 (Similarity function)**.** Given a set of objects or entities E, a *similarity function* is a function $\sigma : E \times E \longrightarrow \mathbb{R}^+$ that maps every pair of entities of E with a real number that expresses the grade of likeness between the entities and that satisfies the following properties:

$$\text{Non negativity: } \forall x, y \in E, \sigma(x, y) \geq 0, \tag{1}$$

$$\text{Maximality: } \forall x, y \in E, \sigma(x, x) \geq \sigma(x, y). \tag{2}$$

Some authors add to these properties *symmetry* but, as showed in [Tversky, 1977], it does not hold in some contexts.

Definition 2 (Symmetric similarity function)**.** A *similarity function* $\sigma : E \times E \longrightarrow \mathbb{R}^+$ is *symmetric* if satisfies

$$\text{Symmetry: } \forall x, y \in E, \sigma(x, y) = \sigma(y, x). \tag{3}$$

In the same way, it is possible to define a function to measure dissimilarity.

Definition 3 (Dissimilarity function)**.** Given a set of objects or entities E, a *dissimilarity function* is a function $\delta : E \times E \longrightarrow \mathbb{R}^+$ that maps every pair of entities of E with a real number that expresses the grade of unlikeness between the entities and that satisfies the following properties:

$$\text{Non negativity: } \forall x, y \in E, \delta(x, y) \geq 0, \tag{4}$$

$$\text{Minimality: } \forall x \in E, \delta(x, x) = 0. \tag{5}$$

There are more constraining notions of dissimilarity, such as distances or metrics.

Definition 4 (Distance)**.** Given a set of entities E, a *distance* is a dissimilarity function that satisfies

$$\text{Definiteness: } \forall x, y \in E, \delta(x, y) = 0 \text{ iff } x = y, \tag{6}$$

$$\text{Symmetry: } \forall x, y \in E, \delta(x, y) = \delta(y, x), \tag{7}$$

$$\text{Triangular inequality: } \forall x, y, z \in E, \delta(x, y) + \delta(y, z) \geq \delta(x, z). \tag{8}$$

In order to facilitate comparison, the functions of similarity and dissimilarity should be normalised.

Definition 5 (Normalised (dis)similarity function)**.** A (dis)similarity function is *normalised* if their range is the real unit interval $[0, 1]$.

Very often, similarity functions are built from dissimilarity functions or distances.

Definition 6 (Correspondence between similarity and dissimilarity functions)**.** A similarity function σ and a dissimilarity function δ, are correspondent if

$$\sigma(x, y) = \pi(\delta(x, y)),$$

for some isomorphism $\pi : R \longrightarrow [0, 1]$ that inverts the order (decreasing monotony) and such that $\pi(0) = 1$.

Some popular isomorphisms π are

a) $\pi(x) = 1 - x$, when δ is normalised.

b) $\pi(x) = 1 - \dfrac{x}{\max \delta}$, when δ has a maximal value.

c) $\pi(x) = 1 - \dfrac{x}{1 + x} = \dfrac{1}{1 + x}$, when δ is not bounded.

Taxonomy of similarity functions

We have found in literature many similarity functions that have been used in the Semantic Web, particularly for ontology or schema matching. In [Ehrig et al., 2005] there is a classification of similarity functions for ontology matching, based on the source of information taken for assessing the similarity measure, either from the ontologies being matched (with their attributes, relations and instances), or from other related ontologies (through previous mappings). But we have no found any classification based on data types of the entities to compare with the similarity functions. Hence, we are interested in building a taxonomy of similarity functions according to this classification.

Data types define the semantic of data. According to ISO/IEC 11404 for general purpose data types [iso, 2007] a data type is, from the conceptual point of view, a set of different values characterized by a set of properties and operations. This definition includes both the intensional part of the type, which provides the properties that their values satisfy, and the extensional one, which gives the explicit set of values that conform the data type. The taxonomy established in this work is based in the intensional part of the specification, that is, in the properties of values of data types that allow or disallow the use of different similarity functions.

According to the data type complexity of their domain, we distinguish the following similarity functions:

1. **Similarity functions for simple data types**. These are similarity functions whose domain is a pair of basic data types, such as numeric types (e.g. integers, real numbers, etc.), characters, Booleans, etc. Basic similarity functions well known fall into this category, which is dealt with in the next section.

2. **Similarity functions for compound data types**. These are similarity functions whose domain is a pair of data types composed of elements of other data types (usually a collection of elements). The functions for these data types are the most interesting part of the taxonomy because most similarity functions used in ontology or schema matching fall in this category. The classification under this category is explained after the section corresponding to simple data types.

Figure 1 shows the taxonomy built according to this criterion. This taxonomy defines an `is_a` hierarchy of similarity functions based on the data types of their domain[1]. Hence, any similarity function defined for a data type could be calculated for any of its descendant data types in the hierarchy.

[1] We refer here to the mathematical sense of the word domain, that is, the set of elements where a function is defined.

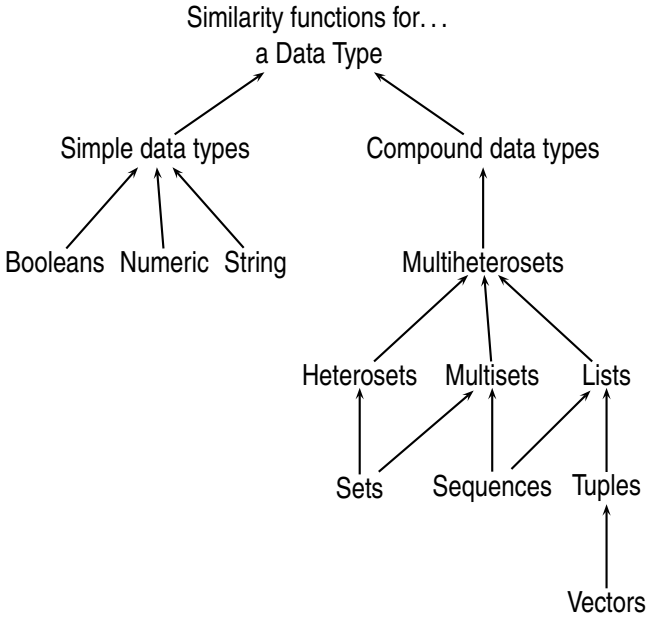


Figure 1: Similarity functions taxonomy based on domain data types. The hierarchy is organized according to the `is a` relation. We have omitted the "Similarity functions for" prefix in each node to narrow the graph.

At the root of the hierarchy we define the most trivial similarity function, the identity similarity, that is valid for all data types.

Definition 7 (Identity similarity)**.** Given a universe E of elements of any data type, the *identity similarity function* is a similarity function $\sigma_{id} : E \times E \longrightarrow [0, 1]$ defined as

$$\sigma_{id}(s,t) = \begin{cases} 1, & \text{iff } s = t; \\ 0, & \text{iff } s \neq t. \end{cases}$$

The next sections present the main categories of this taxonomy and guive some examples of common similarity functions that populate them.

Similarity functions for simple data types

Simple data types are data types whose elements cannot be decomposed in simpler ones. To identify similarity functions for this part of the taxonomy, we have to take into account that the comparison between elements of a data type depends on its semantics and its properties. For example, a number is represented by a symbol, but its associated semantics is a magnitude expressed by a quantity, hence it is possible to use similarity functions that compare quantities. The words of a language do not represent quantities, but they have a meaning in the language that permits their comparison by means of synonymy, hyponymy or meronymy relations.

In this section we review the most important similarity functions for Boolean, numeric and string data types.

Similarity functions for Booleans

A *boolean* is a data type that only has two possible values, usually represented as 0 and 1 (or `False` and `True` respectively).

This is the most simple data type, and the only similarity function for them is the *Boolean identity function*, which coincides with the XNOR operation for bits.

XNOR similarity is the basis to compare bits sequences in computer communication, for example. Moreover, this similarity could be used for comparing boolean attributes of ontology or schema entities.

Similarity functions for numeric data types

A *number* is a data type that represents a quantity. There are many numeric data types, but the most common are Natural $\mathbb{N}$, Integer $\mathbb{Z}$, Rational $\mathbb{Q}$ and Real $\mathbb{R}$. The fact that a number has associated a quantity permits defining an order relation between the numbers that could be used to compare them by means of their difference.

The most important distance for two numbers x and y is the *difference* that is defined as $\delta(x, y) = |x - y|$. This distance (and the corresponding similarity function) is not normalized, but usually it is normalized dividing it by the maximal distance in the set of numbers to compare.

Similarity functions for strings, words or terms

A *string* is a sequence of symbols or characters in an alphabet. In natural languages, strings forms words, terms, and texts. *Words* are the units of a language [Wilson, 2008], and a *term* is a sequence of few words in a language, which represents a concept. Words and terms could be seen as a sequence of elements of type character, and from that point of view, they could be managed as a compound data type, but here, words are considered indivisible in elements belonging to simpler data types.

Natural languages have the capacity of expressing the same or similar concepts with different words or terminological variations. According to [Maynard and Ananiadou, 1999] the main kinds of word variations are

Morphological: It is a variation in the form of a word based on the same linguistic root. There are three types: inflection, derivation and a combination of both.

Syntactic: It is a variation in the grammatical structure of a word. There are three types: insertion, permutation and coordination.

Morphosyntactic: It is a combination of both variations, morphological and syntactic.

Semantic: It is a variation in the whole form of a word, providing a synonym, hypernym or hyponym.

Techniques for detecting morphological variations are based on stemming algorithms that transform a word in its linguistic root, eliminating prefixes or suffixes of number, gender, and tense. One of the algorithms most used for the English language is Porter algorithm [Porter, 1980].

Techniques for detecting syntactic variations are based on rewriting rules, such as the rule that inverts the order of two consecutive words in a term. These rules are detected by analysing the repetition of patterns in a huge linguistic corpus.

On the other hand, semantic terminological variations could be assessed with the help of dictionaries or thesauri, e.g. WordNet [Miller, 1995]. Some similarities functions use the synsets (sets of synonyms) of thesauri and others use their taxonomies. The taxonomy of a thesaurus defines a conceptual space by means of hyponymy or meronymy relations ($\sqsubseteq$) and it is possible to exploit the semantic association between concepts through these relations in order to establish their similarity. The most famous similarity functions based on thesauri are the *information content similarity function* and the *Resnik similarity function* [Resnik, 1995].

The other option to measure the semantic similarity between concepts in a taxonomy is by means of the distance between the corresponding nodes. The logic behind is that two concepts are more similar as shorter is the path between them. The easiest way to calculate this distance is to count the number of arcs, that is, the length of the shortest path between nodes. This idea was proposed in [Rada et al., 1989] and later applied to WordNet in

[Lee et al., 1993] and [Leacock and Chodorow, 1998]. The most important similarity functions that apply this are the shortest path similarity and the ancestor similarity [Maedche and Zacharias, 2002].

Finally, there is also the possibility of combining both alternatives, information content based and path distance based, as shown in [Hahn and Chater, 1997].

Similarity functions for compound data types

Compound data types are constructed from elements of other data types following some structure. For comparing compound data types it is necessary to consider not only the similarity between the components data types, but the information included in the structure itself. It is important to emphasize the properties of every type of structure in order to define similarity functions.

Considering the main structural properties, we present a novel classification of aggregated data types according to four fundamental properties: homogeneity, size, order, and element uniqueness.

Homogeneity refers to the condition that all elements are, or are not, of the same base data type.

Size refers to the fact that the number of elements that conforms the collection is fixed or variable. If it is fixed, the comparison is always carried out between entities with the same size. If it is variable, a comparison between entities with different sizes is possible.

Order refers to fact that elements in the collection must follow an established order. This order has nothing to do with the underlying or natural order of data type elements, but with the structure of the collection itself.

Uniqueness refers to the condition that element repetitions are, or are not, allowed.

If we make a grid over the above properties we find the data types shown in Table 1. In that table some cells of the grid are empty, but this fact does not mean that a data type of such characteristics does not exist. For example, the upper left corner cell corresponds to what in combinatorial are *variations*, which are ordered collections of data with fixed size of the same type that do not repeat, or if we relax the order restriction we have *combinations*. However, these data types are rarely used in schemas or ontologies and will not be dealt with in this article.

Table 1: Classification of the collection data types according to properties of homogeneity, size, order and element uniqueness.

		Fixed size		Variable size	
		With uniqueness	Without uniqueness	With uniqueness	Without uniqueness
Homogeneous	With order		*Vector*		*Sequence*
	Without order			*Set*	*Multiset*
Heterogeneous	With order		*Tuple*		*List*
	Without order			*Heteroset*	*Multiheteroset*

The above four properties establish, in fact, constrains that give structure to data types. For example, a data type with order is more structured than a data type without order, and a data type with fixed size is more structured than a data type with no restriction about size. Thereby, the more structured a data type is, the more information we have to calculate semantic similarities.

In addition, these properties induce an `is a` hierarchy over the data types of table 1 and over the similarity functions applied to the entities of these data types, as shown in Figure 1. In this hierarchy the similarity functions corresponding to the most structured data types are in the lowest levels. This hierarchy facilitates the categorization and the application of similarity functions for these data types because similarity functions for a data type could

be applied to all its descendant data types. For example, similarity functions for a data type without uniqueness restriction as multisets could also be applied to data types with uniqueness restriction as sets, because a set `is a` particular case of multiset.

Hence, in the rest of this section, we proceed to present the categories of compound similarity functions, giving some examples common similarity functions that populate them.

Similarity functions for multiheterosets

A *multiheteroset* is a variable-sized unordered collection of elements of different types where repetitions of elements are allowed. Multiheterosets are the less structured data types because they have no restriction. As a consequence, there are few ways of comparing them, especially when data types of multiheteroset elements are incomparable. The most intuitive function results from the comparison between the intersection and the union of the multiheterosets, but taking into account element repetitions, and it is known as *Jaccard similarity function* [Jaccard, 1901].

It is possible to assign different weights to commonalities and discrepancies, as for example does *Dice's similarity function*, that assigns double weight to coincidences [Dice, 1945].

Similarity functions for heterosets

A *heteroset* is a variable-sized unordered collection of elements of different types where repetitions of elements are not allowed.

Heterosets are a particular case of multiheterosets; therefore, every similarity function of the previous section could be used for heterosets.

In addition to the previous similarity functions, here is possible to follow a more general approach that comes from Tversky's contrast model [Tversky, 1977], where similarities between two objects A and B depends on the features common to A and B, features in A but not in B and features in B but not in A. Some well-known similarity functions derived from this model are the *Manhattan*, *Simpson* and *Euclidean similarity functions* [Holliday et al., 2002].

Similarity functions for multisets

A *multiset* is a variable-sized unordered collection of elements of the same type where repetitions of elements are allowed.

As multisets are a particular case of multiheterosets, all similarity functions for multiheterosets could be used with multisets.

From a statistical point of view, multisets are samples with an associated frequency distribution. Hence, it is possible to compare distributions to asses a similarity measure. A well-known test for comparing distributions is *Chi-square test* [Manning and Schütze, 1999].

Another different approach is to assign a weight to each element, which depends not only on its frequency in the multiset, but also on its frequency in other multisets in the same application domain. A well-known function that uses this idea is TF$\times$IDF (*Term Frequency* $\times$ *Inverse Document Frequency*); this function is usually applied in the vector space model for comparing documents (viewed as multisets of words) in information retrieval [Salton et al., 1975].

Similarity functions for sets

A *set* is a variable-sized unordered collection of different elements of the same type where repetitions of elements are not allowed.

Sets are particular cases of heterosets and multisets, therefore all similarity functions for these data types could be used for sets. These similarity functions only take into account whether sets have or not common elements; however, when elements are different they do not take into account their differences, because many times are incomparables. However, as elements of sets are of the same type, they can be easily compared. Therefore, if there exists a similarity function or distance for the elements of the set, it is possible to calculate the similarity between two sets as a function of the similarity between their elements. There are several alternatives: the most optimistic is to take the two elements more similar, and the most pessimistic is to consider the two elements less similar. The problem of these measures is that only one pair of elements is considered. Most of the time it is much more reasonable to consider similarities between all possible pairs (average similarity) However, this last function is not really a similarity function because it does not satisfy the maximality condition (2). According to [Valtchev and Euzenat, 1997], to satisfy maximality condition it is necessary to match elements and to measure the similarities between the elements matched. This optimal matching must satisfy the following:

a) Every element of A must match at most one element of B and vice versa.
b) The number of elements matched must be maximal.
c) The matching must make the similarity maximum.

The search of such optimal matching is an optimization problem that has been addressed in [Ahuja et al., 1993].

Similarity functions for lists

A *list* is a variable-sized ordered collection of elements of different types where repetitions of elements are allowed.

The only constrain for the lists is order. If we relax this constrain, the lists could be seen as multiheterosets; then we can use similarity functions for multiheterosets to compare lists.

Exploiting the order restriction provides different strategies for comparing lists. The easiest strategy is to count matching elements at the same list position as does the *Hamming's similarity function*.

It is also possible to consider matching elements in a fixed-size window, as does the *Jaro's similarity function*, that also takes into account transposition of matching elements in the window [Jaro, 1989]. Another variation of this similarity is *Jaro-Winkler's similarity function*, that give higher similarity values to sequences with longer common prefixes [Winkler, 1999].

Although Jaro's and Jaro-Winkler's similarity functions are both valid for lists, they came up in the field of record linkage and are mainly used with strings, which are a particular case of sequences.

Another natural way of comparing lists is through their sub-lists, especially when order is so important as to not permit transpositions. The comparison is done by measuring the length of the biggest common sub-lists. This similarity function is useful to identify identical whole parts in both lists, something very important when these parts define some properties of the lists. Variations of this similarity only take into account prefix or suffix sub-lists.

Similarity functions for sequences

A *sequence* is a variable-sized ordered collection of elements of the same type where repetitions are allowed.

As we have seen, words or terms are a special class of symbol sequences in an alphabet. Therefore, within the context of schema and ontology comparison, the measures presented in this section are used often with labels of entities.

Sequences are a particular case of lists where elements belong to a unique domain; hence, every similarity function for lists could be used for sequences.

On the other hand, it is possible to consider sequences as multisets that relax the order condition and thus to apply the similarity functions for multisets to compare sequences.

To compare sequences, like in the case of lists, the most trivial strategy is to compare elements in the same position. If we only count matches we have Hamming's similarity function; however, this function only takes into account coincidences between elements in the same positions. Therefore, when elements do not match, the function does not measure discrepancies between elements. Now that elements are comparable because they are of the same type, it is possible to extend Hamming's similarity when dealing with similarities between elements in the same position.

A different strategy is to compare all the proper sub-sequences of a fixed length of a sequence. These subsequences are known as n-*grams* [Kondrak, 2005]. It is possible to compare the multisets of n-grams of two sequences using any of the similarity functions for multisets as, for example, Jaccard's similarity function. When we compare terms, this function gives good results for abbreviations.

Another typical strategy is to measure the number of elemental operations that transforms a sequence into another [Hahn and Chater, 1997]. A well-known distance that makes use of this strategy is the edit distance of Levenshtein [Levenshtein, 1966], which only considers three possible operations: insertion, deletion and substitution of an element.

Similarity functions for tuples

A *tuple* is a fixed-size ordered collection of elements of different types where repetitions of elements are allowed.

Tuples are often used in schemas and ontologies to record objects with attributes of different types.

Tuples could be seen as lists that relax the fixed-size restriction; in this case it is possible to use the similarity functions for lists to compare tuples.

Given that tuples have fixed size and are ordered, elements in the same position are in correspondence; hence, it is possible to compare elements in the same position and to combine the similarities obtained with an aggregation function. The most well-known family of similarity functions for tuples is based on averages. For example, if we consider the identity similarity function to compare elements that are in the same position in the tuples, the resulting average similarity function is Hamming's similarity function. When elements of tuples have different importance it is better to use a *weighted average*.

Similarity functions for vectors

A *vector* is a fixed-size ordered collection of elements of the same type where repetitions of elements are allowed.

Vectors are a particular case of tuples and, consequently, it is possible to apply the similarity functions for tuples to compare vectors.

In vectors, as in tuples, there is a correspondence between elements in the same position; besides similarity functions for vectors often combine similarities between elements though using the same similarity function for each position in the vector. There are different ways of combining elements similarities, and one of the most well-known is the family of *Minkowski's distances*. This family offers many well-known distances as, for instance, the *Manhattan distance* or the *Euclidean distance*.

Conclusions and future work

Up to date many similarity functions have been published, but they are not presented and organized in a way that the selection of one or several of them to do semantic comparisons is an easy task. To solve this problem, we have presented a taxonomy of similarity functions. The structure of the taxonomy is induced by a taxonomy of data types, in a way that functions applicable to compare entities of a particular data type are applicable to every data type that is a subclass of this. The compound data types are classified according to their homogeneity, size variation, order and uniqueness. It should be added that the composition of data types allows the composition of

similarity functions. That is, if two entities e_1 and e_2 are, both of them, represented by a data type t composed by the data types $t_1, t2, \ldots, t_n$, then the function used to measure the similarity between e_1 and e_2 can be obtained by composing similarity functions that correspond to $t_1, t_2, \ldots, t_n$. For example, if the input can be represented by a vector of numbers, the similarities between numbers can be combined through the similarity functions for vectors. The function that can be used in the composition depends on where t is inside the taxonomy.

With this taxonomy, the steps to select a similarity function are as follows:

1. To identify the data type that can be used to represent the entities to be compared.
2. To obtain the applicable functions according to the taxonomy.
3. To select the functions to be used according to the criteria of the experts in the issue, the previous use of the functions, etc. At this step, if it is not clear which similarity function to use, it is possible to apply all the similarity functions that populate that category and to decide after analysing the results.

Inside every category of the taxonomy, we have shown some of the most-used similarity functions though it must be said that many others have not been included here for size restrictions.

On the basis of this taxonomy we have built a Java library called SIMEON (Similarity Measures for Ontologies)[2] where the most popular similarity functions has been implemented.

In the near future we plan to populate the taxonomy with new similarity functions and to experiment applying them to an wine ontology, which we have recently build in other project, in order to discover which combinations of similarity functions perform better to compare wines.

Acknowledgements

This paper is published with partial support by the project ITHEA XXI of the ITHEA ISS (www.ithea.org) and the ADUIS (www.aduis.com.ua).

Bibliography

[iso, 2007] Iso/iec 11404:2007. information technology – general-purpose datatypes (gpd). http://www.iso.org/iso/catalogue_detail.htm?csnumber=39479.2007.

[Ahuja et al., 1993] Ahuja, R., Magnanti, T., and Orlin, J. *Network Flows: Theory, Algorithms and Applications*. Prentice Hall, New Jersey, 1993.

[Cohen et al., 2003] Cohen, W. W., Ravikumar, P., and Fienberg, S. E. A comparison of string distance metrics for name-matching tasks. In Knoblock, C. and Kambhampati, S., editors, *Proceedings of IJCAI-03 Workshop on Information Integration*, pages 73–78, Acapulco, Mexico, 2003.

[Cross and Sudkamp, 2002] Cross, V. and Sudkamp, T. A. *Similarity and compatibility in fuzzy set theory : assessment and applications*, volume 93 of *Studies in Fuzziness and Soft Computing*. Physica-Verlag, Heidelberg; New York, 2002.

[Dice, 1945] Dice, L. R. Measures of the amount of ecologic association between species. *Ecology*, 26(3):297–302, 1945.

[Ehrig et al., 2005] Ehrig, M., Haase, P., Stojanovic, N., and Hefke, M. Similarity for ontologies — a comprehensive framework. In Bartman, D., Rajola, F., Kallinikos, J., Avison, D., Winter, R., Ein-Dor, P., Becker, J., Bodendorf, F., and Weinhardt, C., editors, *Proceedings of the 13th European Conference on Information Systems (ECIS 2005)*, Regensburg, Germany, 2005.

[2] https://github.com/asalber/simeon

[Euzenat and Shvaiko, 2007] Euzenat, J. and Shvaiko, P. *Ontology Matching*. Springer, Berlin, Heidelberg, 2007.

[Goldstone, 1999] Goldstone, R. Similarity. In Wilson, R. and Keil, F., editors, *MIT encyclopedia of the cognitive sciences*, pages 763–765, Cambridge, Massachusetts. MIT Press, 1999.

[Hahn and Chater, 1997] Hahn, U. and Chater, N. Concepts and similarity. In Lamberts, L. and Shanks, D., editors, *Knowledge, Concepts and Categories*, Cambridge, Massachusetts. Physhology Press/MIT Press, 1997.

[Holliday et al., 2002] Holliday, J. D., Hu, C. Y., and Willett, P. Grouping of coefficients for the calculation of inter-molecular similarity and dissimilarity using 2d fragment bit-strings. *Combinatorial Chemistry and High-Throughput Screening*, 5(2):155–166, 2002.

[Jaccard, 1901] Jaccard, P. Étude comparative de la distribution florale dans une portion des alpes et des jura. *Bulletin del la Société Vaudoise des Sciences Naturelles*, 37:547–579, 1901.

[Jaro, 1989] Jaro, M. A. Advances in record-linkage methodology as applied to matching the 1985 census of Tampa, Florida. *Journal of the American Statistical Association*, 84(406):414–420, 1989.

[Kondrak, 2005] Kondrak, G. N-gram similarity and distance. In Consens, M. P. and Navarro, G., editors, *12th International Conference String Processing and Information Retrieval (SPIRE)*, volume 3772 of *Lecture Notes in Computer Science, Berlin, Germany*, pages 115–126, Buenos Aires, Argentina. Springer, 2005.

[Koudas et al., 2006] Koudas, N., Sarawagi, S., and Srivastava, D. Record linkage: similarity measures and algorithms. In Chaudhuri, S., Hristidis, V., and Polyzotis, N., editors, *ACM SIGMOD International Conference on Management of Data*, pages 802–803, Chicago, Illinois. ACM, 2006.

[Leacock and Chodorow, 1998] Leacock, C. and Chodorow, M. Combining local context and wordnet similarity for word sense identification. In Fellfaum, C., editor, *MIT Press*, pages 265–283, Cambridge, Massachusetts, 1998.

[Lee et al., 1993] Lee, J. H., Kim, M. H., and Lee, Y. J. Information retrieval based on conceptual distance in is-a hierarchies. *Journal of Documentation*, 49(2):188–207, 1993.

[Levenshtein, 1966] Levenshtein, V. Binary codes capable of correcting deletions, insertions, and reversals. *Cybernetics and Control Theory*, 10(8):707–710. Original in *Doklady Akademii Nauk SSSR* 163(4): 845–848, 1965.

[Maedche and Zacharias, 2002] Maedche, A. and Zacharias, V. Clustering ontology-based metadata in the semantic web. In Elomaa, T., Mannila, H., and Toivonen, H. T. T., editors, *13th European Conference on Machine Learning (ECML'02) 6th European Conference on Principles and Practice of Knowledge Discovery in Databases (PKDD'02)*, Helsinki, Finland, 2002.

[Manning and Schütze, 1999] Manning, C. D. and Schütze, H. *Foundations of Statistical Natural Language Processing*. MIT Press, Cambridge, Massachusetts, 1999.

[Maynard and Ananiadou, 1999] Maynard, D. and Ananiadou, S. Term extraction using a similarity-based approach. In Bourigault, D., Jacquemin, C., and Lhomme, M.-C., editors, *Recent Advances in Computational Terminology*, pages 261–278. John Benjamins, Amsterdam (NL), 1999.

[Miller, 1995] Miller, A. Wordnet: A lexical database for english. *Communications of the ACM*, 38(11):39–41, 1995.

[Porter, 1980] Porter, M. An algorithm for suffix stripping. *Program*, 14(3):130–137, 1980.

[Rada et al., 1989] Rada, R., Mili, H., Bicknell, E., and Blettner, M. Development and application of a metric on semantic nets. *IEEE Transactions on Systems, Man and Cybernetics*, 19(1):17–30, 1989.

[Resnik, 1995] Resnik, P. Using information content to evaluate semantic similarity in a taxonomy. In Perrault, C. R., editor, *Proceedings of the 14th IJCAI*, pages 448–453, Montréal (Canada), 1995.

[Salton et al., 1975] Salton, G., Wong, A., and Yang, C.-S. A vector space model for automatic indexing. *Communications of the ACM*, 18(11):613–620, 1975.

[Tversky, 1977] Tversky, A. Features of similarity. *Psychological Review*, 84(4):327–352, 1977.

[Valtchev and Euzenat, 1997] Valtchev, P. and Euzenat, J. Dissimilarity measure for collections of objects and values. In Liu, X., Cohen, P. R., and Berthold, M. R., editors, *Advances in Intelligent Data Analysis, Reasoning about Data, Second International Symposium, IDA-97*, volume 1280 of *Lecture Notes in Computer Science, Berlin, Germany*, pages 259–272, London, United Kingdom. Springer, 1997.

[Wilson, 2008] Wilson, B. The natural language processing dictionary. http://www.cse.unsw.edu.au/~billw/nlpdict.html, 2008.

[Winkler, 1999] Winkler, W. E. The state of record linkage and current research problems. Technical Report Statistical Research Report Series RR99/04, U.S. Bureau of the Census, Washington, D.C., 1999.

Authors' Information

Alfredo Sánchez-Alberca *- San Pablo CEU University of Madrid, Urbanización Montepríncipe, s.n. 28668 Boadilla del Monte, Spain; e-mail: asalber@ceu.es*
Major Fields of Scientific Research: Artificial intelligente, Machine learing, Knowledge modelling and representation, Ontologies and Semantic Web

Juan Castellanos-Peñuela *- Head of Natural Computing Group, Faculty of Computer Sciences, Polytechnic University of Madrid, Campus de Montegancedo, s.n., 28668 Boadilla del Monte, Spain; email: jcastellanos@fi.upm.es*
Major Fields of Scientific Research: Natural computing, Formal language and Automata theory.

Rafael Lahoz-Beltra *- Faculty of Biological Sciences, Complutense University of Madrid, 28040 Madrid, Spain; e-mail: lahozraf@ucm.es*
Major Fields of Scientific Research: Evolutionary computation, Embryo development modelling and Design of bioinspired algorithms.

CONSTRAINED OBJECT-CHARACTERIZATION TABLES AND ALGORITHMS[1]

Hasmik Sahakyan

Abstract: *Let C be a collection of objects, characterized by the set $A = \{a_1, \cdots, a_n\}$ of binary attributes. We consider problems of the following type: given an object-characterization table, it is to check if there exists a subset M in C of a given size, such that each attribute of A is satisfied by a given number of objects in M. Additional restriction may be applied such as - the number of matches of each object in M is limited. In this paper we investigate particular cases of the general problem, and consider approximation solutions by means of binary classification trees.*

Keywords: *Classification tree, covering, greedy algorithm.*

ACM Classification Keywords: *F.2.2 Nonnumerical Algorithms and Problems*

1. Introduction

Let C be a collection of objects (repetitions is allowed), where every object is given by the same set $A = \{a_1, \cdots, a_n\}$ of binary attributes. An object may satisfy the attribute a_i or not. Consider the following problem: is there an m-subset $M \subseteq C$, such that each attribute a_i is satisfied at least by one object of M for $i = 1, \cdots, n$? In practical level our interest is in representing the diversity of attribute values by the narrow subsets of objects. Machine learning techniques provide algorithmic means of the problem. Mathematically, the problem is related to transversals and sets of different representatives. As formal description related to our problem we will consider the MINIMUM COVER combinatorial problem which is one of the well known NP-complete algorithmic problems [Garey, Johnson, 1979].

MINIMUM COVER (MC)

> Given a finite set S, a collection C of subsets of S, and a positive integer $K \leq |C|$. Does there exist a cover $C' \subseteq C$ of S such that $|C'| \leq K$, i.e. does there exist a subset $C' \subseteq C$ such that $|C'| \leq K$ and every element of S is in at least one subset of C'.

Additional requirements that are common for application problems restrict the domain of possible solutions and create sub-problems, which can remain NP-complete or can have polynomial solutions. Restrictions can be applied on the number of occurrences of elements of S in subsets of C. For example, the number of occurrences of elements of S in subsets of C can be bounded from above by a constant t. However this particular sub-problem of MC is also NP-complete for $t > 1$ [Garey,Johnson, 1979]. The problem is not easier when the number of occurrence of i-th element of S is bounded by t_i, $i = 1, \cdots, |S|$.

We consider different sub-problems when restrictions are applied on the number of occurrence of the elements of S in C'. Assume that the number of occurrences of elements of S in subsets of C' is bounded from below: the i-th

[1] Partially supported by grants № SCS 13RF-088, and № 3-1B340 of State Committee of Science of Ministry of education and science of Republic of Armenia

element of S have to occur in at least s_i elements of C'. In this way the i-th element of S is covered at least s_i times.

MINIMUM $(s_1, \cdots, s_n)$-COVER

> Given a finite set S, a collection C of subsets of S, and a positive integer $K \leq |C|$. Does there exist an $(s_1, \cdots, s_n)$- cover $C' \subseteq C$ of S such that $|C'| \leq K$, i.e. does there exist a subset $C' \subseteq C$ such that $|C'| \leq K$ and the i-th element of S belongs to at least s_i subsets of collection C'?

MINIMUM $(s_1, \cdots, s_n)$-COVER is NP-complete, because of MINIMUM COVER with $s_i = 1$ is its sub-problem.

In this paper together with the decision versions of formulated problems we investigate also their *existence versions*.

$(s_1, \cdots, s_n)$-EXISTENCE

> Given a finite set S and a positive integer m. Does there exist a collection C of subsets of S such that $|C| \leq m$ and the i-th element of S belongs to at least s_i elements/subsets of C?

Further we apply more additional restrictions:

- On the number of repetitions in C;
- On the collection C of subsets of S.

The paper is organized as follows: in Section 2 below we present the problems in terms of binary matrices, and consider complexity issues. Section 3 describes hierarchical classification approach with constraints. We investigate counterparts of $(s_1, \cdots, s_n)$-existence in hierarchical classification area and describe approximation algorithms with help of binary classification trees.

2. Problems given in terms of binary matrices

Decision Problems

Let C be a collection of objects given by the value vectors of a set of binary attributes, $\{a_1, \cdots, a_n\}$. We identify each object of C with a binary sequence of size n in the following way: i-th component of the sequence is 1 if and only if the object satisfies the i-th attribute. In this manner we receive a binary matrix of size $|C| \times n$, where rows correspond to the objects of C, columns correspond to the attributes. If t_i is the number of objects satisfying attribute $a_i, i = 1, \cdots, n$, then the i-th column of matrix contains t_i ones. Now we present covering problems in terms of binary matrices and investigate complexity issues of these problems.

MINIMUM $(s_1, \cdots, s_n)$-COVER ($(s_1, \cdots, s_n)$-MC)

> Given a binary matrix M of size $m \times n$, a positive integer $m' \leq m$, and an integer sequence $(s_1, \cdots, s_n)$, where $0 \leq s_i \leq m'$ for $i = 1, \cdots, n$. Does M contain a submatrix M' of size $m' \times n$ such that the i-th column of M' contains at least s_i ones.

Whit an additional constraint applied on the number of repetitions of rows, the problem is formulated as follows:

MINIMUM $(s_1, \cdots, s_n)$-COVER WITH LIMITED REPETITIONS ($(s_1, \cdots, s_n)$-MC-LR)

> Given a binary matrix M of size $m \times n$, positive integers $m' \leq m$ and $r \geq 1$, and an integer sequence $(s_1, \cdots, s_n)$, where $0 \leq s_i \leq m'$ for $i = 1, \cdots, n$. Does M contain a submatrix M' of size $m' \times n$ with at most r repetitions of each row such that the i-th column of M' contains at least s_i ones.

MINIMUM $(s_1, \cdots, s_n)$-COVER WITH NO REPETITIONS ($(s_1, \cdots, s_n)$-MC-NR)

Given a binary matrix M of size $m \times n$, a positive integer $m' \leq m$, and an integer sequence $(s_1, \cdots, s_n)$, where $0 \leq s_i \leq m'$ for $i = 1, \cdots, n$. Does M contain a submatrix M' of size $m' \times n$ with different rows such that the i-th column of M' contains at least s_i ones.

In $(s_1, \cdots, s_n)$-MC-NR we can assume that M consists of different rows, and thus $m \leq 2^n$, and every column of M contains at most 2^{n-1} ones.

Theorem 1. $(s_1, \cdots, s_n)$-MC-NR is NP-complete.

Proof. We prove that $(s_1, \cdots, s_n)$-MC $\propto$ $(s_1, \cdots, s_n)$-MC-NR.

Consider an instance I_1 of $(s_1, \cdots, s_n)$-MC.

Instance I_1: matrix M_1 of size $m \times n$, a positive integer $m' \leq m$, and an integer sequence $(s_1, \cdots, s_n)$.

Now we transform I_1 into the instance I_2 of $(s_1, \cdots, s_n)$-MC-NR:

Instance I_2: matrix M_2 of size $m \times (n + m)$, where the first n columns of M_2 coincide with M_1, and the last m columns compose the unit $m \times m$ matrix; a positive integer $m' \leq m$ and an integer sequence $(s_1, \cdots, s_n, 1, \cdots, 1)$ of size $n + m$.

The Figure 1 below demonstrates the construction of M_2.

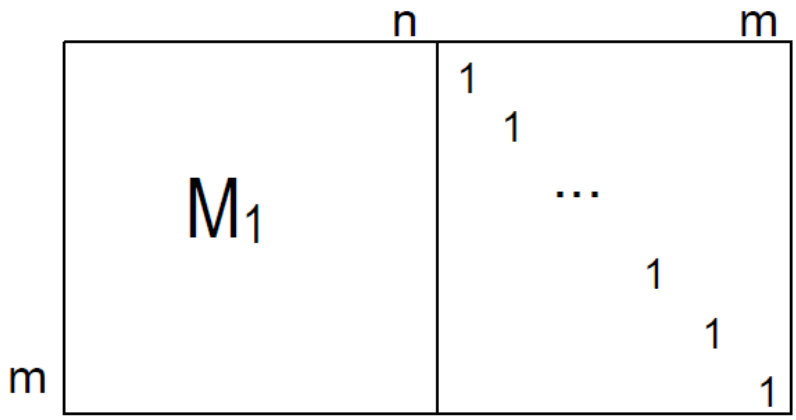


Figure1

It is easy to check that I_1 is a positive instance of $(s_1, \cdots, s_n)$-MC if and only if I_2 is a positive instance of $(s_1, \cdots, s_n)$-MC-NR. The transformations from one to the other instance are in polynomial time complexity.

Corollary. $(s_1, \cdots, s_n)$-MC-LR is NP-complete.

Existence versions

Below we consider the existence versions of the decision problems under consideration. While in covering problems above we require that the matrix contains at most given number of rows and each column of the matrix contains at least the given number of ones, - in existence versions we will require that that the matrix contains exactly m rows and the i-th column has exactly s_i ones. Theorem 2 and Theorem 3 below imply that in this way we do not make the problems easier.

$(s_1, \cdots, s_n)$- EXISTENCE

Given positive integers m, n, and an integer sequence $(s_1, \cdots, s_n)$, where $0 \leq s_i \leq m$ for $i = 1, \cdots, n$. Does there exist a binary matrix M of size $m \times n$ with the column sum vector $(s_1, \cdots, s_n)$.[1]

$(s_1, \cdots, s_n)$- EXISTENCE WITH LIMITED REPETITIONS ($(s_1, \cdots, s_n)$- EXISTENCE-LR).

Given positive integers m, n, r and an integer sequence $(s_1, \cdots, s_n)$, where $0 \leq s_i \leq m$ for $i = 1, \cdots, n$. Does there exist a binary matrix M of size $m \times n$ with the column sum vector $(s_1, \cdots, s_n)$, such that the number of repetitions of each row is limited by r.

$(s_1, \cdots, s_n)$- EXISTENCE WITH NO REPETITIONS ($(s_1, \cdots, s_n)$- EXISTENCE-NR)

Given positive integers m, n, and an integer sequence $(s_1, \cdots, s_n)$, where $0 \leq s_i \leq m$ for $i = 1, \cdots, n$. Does there exist a binary matrix M of size $m \times n$ with the column sum vector $(s_1, \cdots, s_n)$ and with different rows.

Notice that $(s_1, \cdots, s_n)$ - EXISTENCE-NR is a sub-problem of $((s_1, \cdots, s_n)$-MC-NR) with M being the $2^n \times n$ binary matrix whose rows are all binary sequences of size n. In this case, $m' \leq 2^n$, $0 \leq s_i \leq \min\{m', 2^{n-1}\}$.

Complexity

$(s_1, \cdots, s_n)$- EXISTENCE has obvious solution - the matrix exists if and only if $0 \leq s_i \leq m$.

$(s_1, \cdots, s_n)$- EXISTENCE-NR is equivalent to the known hypergraph degree sequence problem.

HYPERGRAPH DEGREE SEQUENCE PROBLEM

Given positive integers m, n, and an integer sequence $(s_1, \cdots, s_n)$. Does there exist a simple hypergraph with n vertices and m hyperedges, such that s_i is the degree of the i-th vertex (number of hyperedges containing the i-th vertex).

Hypergraph degree sequence problem is one of the known open problems in the hypergraph theory [Berge, 89], for which nor a polynomial algorithm is found neither the NP-completeness is proved.

On the other hand $(s_1, \cdots, s_n)$- EXISTENCE-NR is a sub-problem of $(s_1, \cdots, s_n)$- EXISTENCE-LR with $r = 1$, and thus:

$(s_1, \cdots, s_n)$- EXISTENCE-NR and $(s_1, \cdots, s_n)$- EXISTENCE-LR are open problems.

Theorem 2. [Sah, 2009]

Let M be a binary matrix of size $m \times n$ with different rows and with column sum vector $(s_1, \cdots, s_n)$, where $s_i > m/2$ for some index i. Then there exist M', a binary matrix of size $m \times n$ with different rows, with column sum vector $(s_1, \cdots, s_i - 1, \cdots, s_n)$.

Theorem 3. Let M be a binary matrix of size $m \times n$ with different rows and with column sum vector $(s_1, \cdots, s_n)$, where $s_i \geq m/2$ for $i = 1, \cdots, n$ and $s_j > m/2$ for some index j. Then there exists a binary matrix of size $(m + 1) \times n$ with different rows and with the column sum vector $(s_1, \cdots, s_n)$.

Proof. $s_j > m/2$ for some index j implies that in the j-th column of M the number of ones is greater than the number of zeros. Hence there is a binary sequence of size n with 0 in j-th position, which is not contained in M. Compose M' a binary matrix of size $(m + 1) \times n$ by appending this sequences into

[1] $(s_1, \cdots, s_n)$, is the column sum vector of M if the i-th column of M contains exactly s_i ones.

M. Then M' has column sum vector $(s'_1, \cdots, s'_n)$ greater than $(s_1, \cdots, s_n)$ in several components. According to Theorem 2 there exists a matrix with the column sum vector $(s_1, \cdots, s_n)$.

Further we will restrict attention with $(s_1, \cdots, s_n)$- EXISTENCE-NR and $(s_1, \cdots, s_n)$- EXISTENCE-LR.

3. Hierarchical classification with constraints

Hierarchical classification

In general, hierarchical classification addresses the problem of mapping the sets of classified objects into hierarchies of classes. Many important real-world classification problems are hierarchical in their nature, and it is useful when such classes are organized into a class hierarchy - typically in a tree diagram.

Top-down method in hierarchical classification starts from the root of the tree. The set of all objects is initially considered as a single class assigned to the root. Then a classifier/test applied to this set divides the set into smaller subsets/classes, each assigned to a child node of the root in the tree structure. This process is continued until each subset/class contains single object, assigned to nodes – and these are leaf nodes of the tree.

The resulting tree structure can be reduced at any level between the root and the leaf nodes – depending on how deep the classification in the hierarchy is performed. Depending on application problem, hierarchical classification algorithm can always reach a leaf node, or can stop at any level of the hierarchy, - in case if there is a constraint - limitation on the number of classes, sizes of classes, etc.

Constraints

Construction of classification trees, i.e. partitioning each set of objects assigned to a not leaf node, is performed by applying classifiers. The goal in classification problems is to create classifiers in such a way, that they divide the set of objects into classes such that objects in the same class are similar to one another and dissimilar to objects in other classes. For example the CART [Brei, 84] tree divides the sets by the impurity target. Coming from application problems, an additional functional for optimization can be given, and the problem is in constructing a classification tree with optimal value of the functional. In particular, optimality might be required for the height of the tree, the sub-tree weights on layers, the number of sub-trees, and others.

Solving $(s_1, \cdots, s_n)$ - EXISTENCE-NR by classification trees

C is a collection of objects given by the value vectors of a set $\{a_1, \cdots, a_n\}$ of binary attributes. Two objects of C are different if there is an attribute, which is satisfied for one object, and is not satisfied for the other. Otherwise two objects coincide. For each attribute a_i consider s_i - the number of objects satisfying a_i. In these conditions, the problem is in existence/construction of a set of m different vectors/objects.

We will seek approximate solutions of the problem with help of hierarchical classification trees and top down method.

Construction of the tree

The whole set of m objects, which is to be constructed is virtually assigned to the root vertex of the tree. Thus at the root vertex we have one class consisting of m objects. Then we apply a classifier to split the set of objects into two parts depending on whether or not objects satisfy the first attribute. The classifier C_0 on the root is obvious - it divides the whole set into an s_1-subset and an $(m - s_1)$-subset. On the first level of the tree we create two nodes (roots of the left and right subtrees), and assign them s_1 and $(m - s_1)$ objects. Further we will enumerate subtrees on each level and denote by $d_{i,j}$ the size of subset assigned to the j-th subtree on the i-th level. $d_{1,1} = s_1$ and $d_{1,2} = m - s_1$ on the first level. Then we build classifiers $C_{1,1}$, $C_{1,2}$ for partitioning subsets on the first level taking into account that s_2 objects must satisfy the second attribute. $C_{1,1}$ can generate any integer partition of $d_{1,1}$: $d_{1,1} = d_{2,1} + d_{2,2}$, and $C_{1,2}$ can generate any integer partition of $d_{1,2}$: $d_{1,2} = d_{2,3} +$

$d_{2,4}$, provided that $d_{2,1} + d_{2,3} = s_2$, and $d_{2,2} + d_{2,4} = m - s_2$. Let on the k-th level we have constructed p subtrees associated with $d_{k,1}, \cdots, d_{k,p}$-sets respectively. We apply classifiers $C_{k,1}, \cdots, C_{k,p}$ to divide these sets, given that the summary number of all objects associated with the left subtrees equals s_{k+1}. As far as all nodes on the n-th level of the tree must be assigned single objects, and on the other hand, the tree height is known beforehand, - then the classifiers must follow particular goals.

The global/per tree goal can be for example, to achieve required sizes of classes on closer to the root levels.

Local/per level/ goals /or optimization criteria/ must be chosen in such a way that they achieve the optimal value on the solution tree. For example, the criterion can be:

1. Maximization of the number of pairs of different objects;
2. Maximization of the number of different objects;
3. Maximization of the number of odd size-subsets.

Maximization of the number of pairs of different objects

This criterion was investigated in [Sah, 2010; Sah, Asl, 2011] and a greedy algorithm G was proposed, which on each level provides the optimal value of the number of pairs of different objects. According to G each subtree on the same level k, must be partitioned in such a manner that $s_k - (m - s_k)$ or $(m - s_k) - s_k$ difference is distributed equally.

Algorithm G

Without loss of generality we can assume that $s_k \geq (m - s_k)$ for $k = 1, \cdots, n$.

Step1. Construction of the first level of the tree

The first level contains the roots of two subtrees: the right subtree associated with the s_1-set; and the left subtree associated with the $(m - s_1)$-set. We denote the sizes of these sets by $d_{1,1}$ and $d_{1,2}$. Hereafter the first sub-index will indicate the number of level and the second – the number of subtrees at the level. Odd numbers correspond to right subtrees, and even numbers correspond to left subtrees. Thus,

$d_{1,1} + d_{1,2} = m$, $d_{1,1} = s_1$, and the number of pairs of different objects is equal to: $d_{1,1} \cdot d_{1,2}$.

Let we have constructed the first $k-1$ levels of the tree. In general, $(k-1)$-th level contains 2^{k-1} subtrees. Empty subtrees are possible, and let assume that $(k-1)$-th level contains p non-empty subtrees associated with $d_{k-1,1}, d_{k-1,2}, \cdots, d_{k-1,p}$-sets, respectively. Objects coincide within the same set and differ otherwise. When all subtrees on some level are associated with single objects, then at this level all objects are differentiated and the maximum number of pairs of different objects is already achieved. Further constructions are arbitrary.

Step k. Each subtree associated with $d_{k-1,i}$ -set is partitioned into left and right subtrees: $d_{k-1,i} = d_{k-1,i,l} + d_{k-1,i,r}$, taking into account that $\sum_{i=1}^{p} d_{k-1,i,l} = m - s_k$ and $\sum_{i=1}^{p} d_{k-1,i,r} = s_k$. The number of pairs of different objects will increase by $\sum_{i=1}^{p} d_{k-1,i,l} \cdot d_{k-1,i,r}$.

We will realize partitions of sets having a goal to minimize size differences.

The idea is in the following: if $s_k = m - s_k$ for $i = 1, \cdots, n$, then we would divide every set into 2 equal (± 1) parts and assign them to the left and right subtrees respectively. This will lead to 1-size sets in logarithmic number ([Knuth, 1973]) steps. Furthermore, among all integer partitions of $d_{k-1,i} = d_{k-1,i,l} + d_{k-1,i,r}$ the largest value of $d_{k-1,i,l} \cdot d_{k-1,i,r}$ is achieved when $d_{k-1,i,l} = d_{k-1,i,r}$. Thus following this strategy would lead to the goal, but in general at each step k we have $s_k - (m - s_k)$ difference. Trying to be closer to equal sizes of sets us:

1) Distribute $s_k - (m - s_k)$ difference among sets/subtrees keeping a "homogeneous" distribution;

2) Divide the remaining sets/subtrees into 2 equal parts.

Example

$m = 14$, $n = 6$,$(s_1, s_2, s_3, s_4, s_5, s_6) = (7,10,10,12,12,12)$. Figure 2 demonstrates the construction of a tree by the algorithm G. Circles correspond to the nodes, and the numbers in circles are sizes of associated subsets. On the last level of the tree there are 2 nodes associated with 2-sets (filled circles). Thus in this example two pairs of objects are not differentiated.

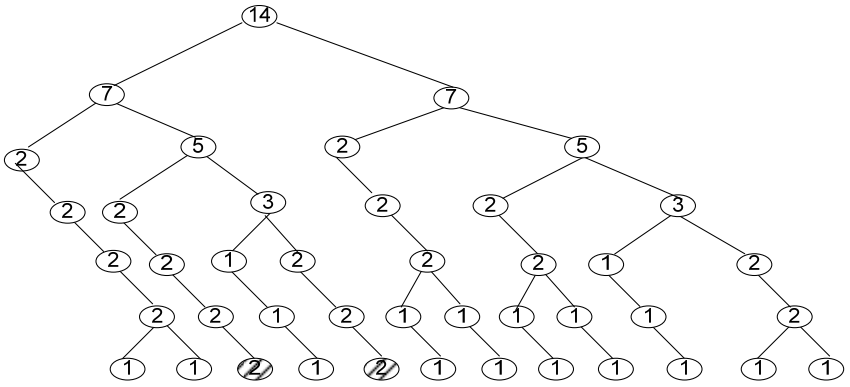


Figure 2

The second criterion, maximization of the number of different objects is not reasonable as creating 1-size classes at closer to the root levels will cause large sets afterwards, which may not be differentiated till the n-th level of the tree.

Maximization of the number of odd size sets

The main reasoning for using this criterion is as follows:

For each attribute a_k the number of objects is given satisfying to it. If $s_k = m - s_k$, then all classifiers on the k-th level would divide every set into 2 equal (± 1) parts. As much s_k is greater than $m - s_k$, so the partitions of the sets are unequally. The algorithm G distributes firstly the $s_k - (m - s_k)$ difference, and then divides the remaining sets into 2 equal parts. So, odd-size sets will reduce the difference $s_k - (m - s_k)$.

Below we construct algorithm G', which maximizes the number of odd-size sets in local step. G' can be considered as a modification of the algorithm G.

Algorithm G'

Assume that $s_1, \cdots, s_n$ are arranged in the increasing order. In each step we:

- Apply the algorithm G;
- Modify the partitioning.

Suppose that $d_{k-1,j}$ is the size of some set on the $(k-1)$-th level. Consider the partitioning of $d_{k-1,j}$ on the k-th level. We distinguish the following cases:

a) $d_{k-1,j}$ is odd.

Any partitioning of $d_{k-1,j}$ will produce one odd and one even size. Algorithm G' does not change the partitioning, except the cases mentioned in b) 2) below.

b) $d_{k-1,j}$ is even.

Suppose that: $d_{k-1,j} = d_{k-1,j,l} + d_{k-1,j,r}$ by G.

Distinguish the following cases:

1) $d_{k-1,j,l}$ and $d_{k-1,j,r}$ are both odd.
   Algorithm G' leaves this partitioning unchanged, except the cases mentioned in 2) below.
2) $d_{k-1,j,l}$ and $d_{k-1,j,r}$ are both even, or one of them is equal to 0.
   Algorithm G' partitions as:
   $d_{k-1,j} = (d_{k-1,j,l} + 1) + (d_{k-1,j,r} - 1)$ or $d_{k-1,j} = (d_{k-1,j,l} - 1) + (d_{k-1,j,r} + 1)$,
   if it is possible to shift the partitioning of some odd-size set; and s_k is enough to allow this modification.

It is clear that the algorithm G' is optimal locally, as any other partitioning cannot increase the number of odd sizes.

Two additional notes:

i. In case of the same number of odd sizes, it is preferable to have maximal number of non-empty sets.
ii. If $\log_2(d_{k-1,j}) > n - k$, then modify the partitioning to avoid this size.

Consider the example above and apply the algorithm G'. Figure 3 demonstrates the construction of the tree by G'.

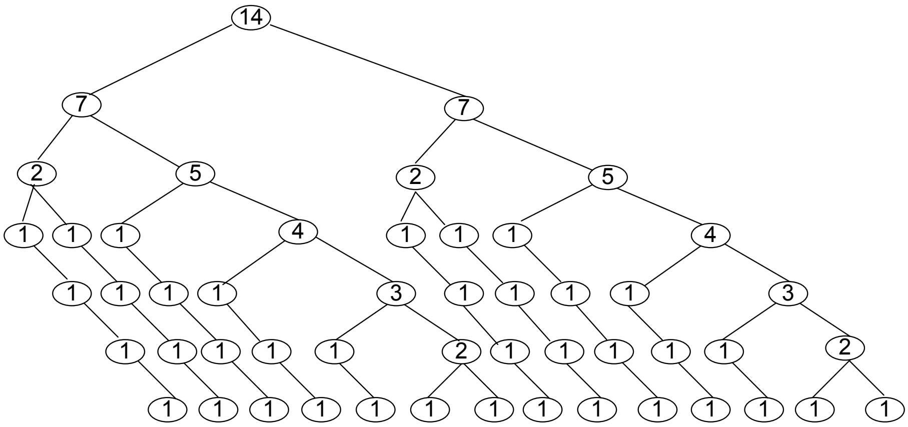


Figure 3

Now all pairs of objects are differentiated.

Solving $(s_1, \cdots, s_n)$- EXISTENCE-LR by classification trees

Construction of the tree is similar to the case of $(s_1, \cdots, s_n)$- EXISTENCE-NR. The only difference is that all nodes on the n-th level can be assigned at most r objects.

Local /per level/ goals /or optimization criteria/

1. Maximization of the number of pairs of different $\leq r$ size subsets;
2. Maximization of the number of $\leq r$ size-subsets;
3. Minimization of the largest size subset.

Maximization of the number of pairs of different $\leq r$ size subsets

It is obvious that the maximization of pairs of different objects provides the maximization of pairs of different $\leq r$ size subsets.Thus the algorithm G provides local optimal solution for $(s_1, \cdots, s_n)$ - EXISTENCE-LR.

By maximizing $\leq r$ -size classes or minimizing the largest size at closer to the root levels possibly will cause large sets of objects on low levels. Thus the two other criteria are seemed not reasonable.

Conclusion

Constrained object-characterization matrices appear in many areas raising questions about their existence or requiring algorithms that can construct them. Set theoretical and combinatorial relation is able to analyze the global picture but as we see here several hard fundamental problems appear that resist to be solved for many decades. The finite, combinatorial nature of the problem is complementarily related to step by step constructing technique such as greedy algorithms and hierarchical classifications. Combining the two parts – combinatorial and heuristic, gives a general picture of the current state in this area of subsets and set systems, element weights, classifications, hierarchies and algorithms of approximation.

Bibliography

[Asl, Kh, 2011] Aslanyan L., Khachatryan R., Association rule mining with n-dimensional unit cube chain split technique, Information Theories and Applications, Vol. 17, Number 2, pp. 108-123, 2010-11.

[Asl, Sah, 2009] L. Aslanyan H. Sahakyan, Chain Split and Computations in Practical Rule Mining, Book 08 Classification, Forecasting, Data mining, Institute of Information Theories and Applications FOI ITHEA 2009, pp.132-135.

[Berge, 89] C. Berge, Hypergraphs, North Holland, Amsterdam, 1989.

[Brei, 84] L. Breiman, J.H. Friedman, R.A. Olshen and C.J. Stone, Classification and Regression Trees, Belmont, CA: Wadsworh, Inc., 1984.

[Garey, Johnson, 1979] M.R. Garey, D.S.Johnson, Computers and Intractability: A Guide to the Theory of NP-Completeness. W. H. Freeman & Co. New York, NY, USA ©1979, ISBN:0716710447.

[Knuth, 1973] D. Knuth, The Art of Computer Programming, vol.3. Sorting and Searching, Addison-Wesley Publishing Company, 1973.

[Kuhn, 05] F. Kuhn, P. von Rickenbach, R. Wattenhofer, E. Welzl, and A. Zollinger. Interference in cellular networks: The minimum membership set cover problem. In Proc. 11th COCOON, volume 3595 of LNCS, pages 188–198. Springer, 2005.

[Sah, 2009] H. Sahakyan, Numerical characterization of n-cube subset partitioning, Discrete Applied Mathematics 157 (2009) 2191-2197.

[Sah, 2010] H.Sahakyan, Approximation greedy algorithm for reconstructing of (0.1)-matrices with different rows, Information Theories and Applications, Vol. 17, Number 2, pp. 124-137, 2010-11.

[Sah, 2013] H. Sahakyan, (0,1)-Matrices with different rows, IEEE Explore, 7p. DOI:10.1109/CSITechnol.2013.6710342

[Sah, Asl, 2008], H. Sahakyan, L. Aslanyan, Lagrangean approximation for combinatorial inverse problems, Book 1 Algorithmic and Mathematical Foundations of the Artificial Intelligence, Institute of Information Theories and Applications FOI ITHEA, 2008, pp. 14-20.

[Sah, Asl, 2011] H. Sahakyan, L. Aslanyan, Evaluation of Greedy algorithm of constructing (0,1)-matrices with different rows, Information Technologies & Knowledge Vol.5, Number 1, pp. 55-66, 2011.

Authors' Information

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

AN APPROACH TO THE MODELING OF THE COGNITIVE ABILITY MANAGING THE FOCUS OF ATTENTION ON THE BASIS OF THE REFLEXIVE MECHANISM OF THE SYSTEM

Olga Nevzorova, Vladimir Nevzorov

Abstract*: The article reveals an approach to modelling of the cognitive ability managing the focus of attention (an ability to switch one's attention) which is based on the reflexive mechanisms developed in "OntoIntegrator" system. There were considered basic experiments to classify the mathematical texts, and demonstrated the reflection mechanisms for the solution of a set task.*

Keywords: *ontology, reflection, attention switch, ontological mark-up of the text*

ACM Classification Keywords: *H.3 INFORMATION STORAGE AND RETRIEVAL H.3.1 Content Analysis and Indexing - Linguistic processing*

Introduction

In the modern researches the problem of a reflection is considered, at least, in three directions: when studying thought, self-awareness of the personality, and also communication and cooperation processes. The study of reflection when solving different problems of the cogitative activity is aimed to detect the structure of system of knowledge and the organization of the thinking process. The article presents the reflection in terms of thinking processes, self-awareness by the system of structures of the knowledge representation and thinking mechanisms. In order to develop the ontological reflection it is necessary to work out the mechanism allowing tracing internal logic of the content of knowledge, including models of knowledge representation and knowledge processing.

The article suggests a way to model the cognitive ability to control the focus of the attention, considered in a context of the problems of the psychology of thinking. By [Gippenreyter, 2001] the attention is described as the most important component of the different psychical processes, and basic properties of attention, such as concentration, range, distraction, short span of attention, intensity, stability and refocusing (Figure 1) are distinguished.

This article examines the mechanisms of the attention switch, executed on the basis the reflection mechanisms of the "OntoIntegrator" system [Nevzorova et. al., 2013]. In the system the ontology-linguistic approach [Nevzorova, 2007] integrating conceptual and technological decisions, allowing designing the solutions of complex tasks of the texts processing in the semantic space of knowledge, represented as a system of ontological models, is consequently executed. On one hand, the system of ontological models structures the semantic space on the other hand it controls the solution of applied linguistic tasks.

Developed tools in the "OntoIntegrator" system are intended for processing of texts in Russian language and contain a number of specialized databases built on the basis of processing of the corpuses in Russian language. On one hand, the system of ontological models structures the semantic space on the other hand it controls the solution of applied linguistic tasks.

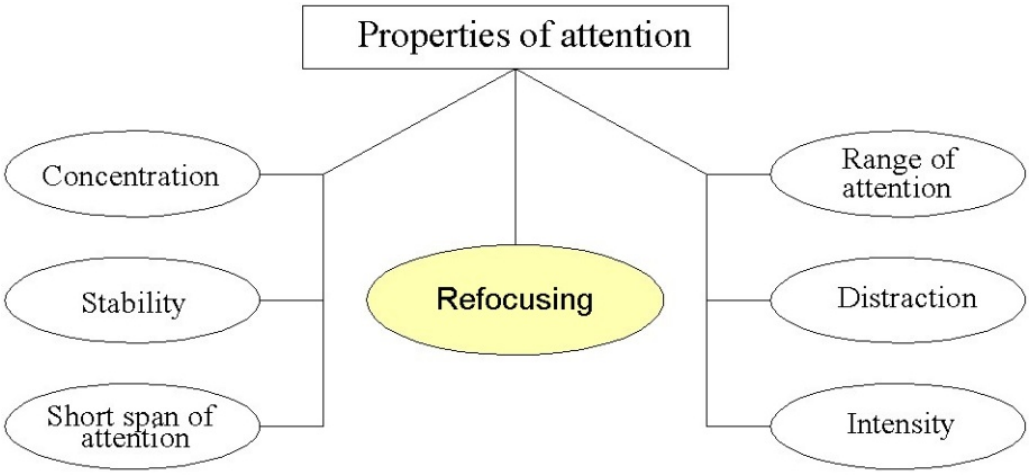


Figure 1. Basic properties of attention

In psychology under the attention switch it is meant the conscious and knowledgeable, willful and purposeful change of the direction of thinking from one object to another caused by the setting of a new goal. In our experiments, there will be considered the task of automatic switch of the focus of attention in order to classify mathematical texts according to their the areas of knowledge. «The attention switch» of the system will be performed on the basis of specified criteria, and by the focus change during the process of the mathematical document analysis various resources of the system, providing relevant text processing, will automatically connect.

The reflection mechanisms of the “OntoIntegrator” system

The process of solving of an applied linguistic problem in the "OntoIntegrator" system executed under the management of the ontological models system [Nevzorova et. al., 2013]. The system of ontological models includes different types of ontologies: the applied ontology, the ontology of models and the ontology of the task planning.

From the point of view of the structural organization the system of ontological model is a three-component associative system. The components of the system are the ontology of the task planning, the ontology of the models and the applied ontology. In the system the interpretation of the applied ontology as a set of ontologies of different problem domains (and, accordingly, of different semantic interpretations) is allowed: external, connected by a user, and internal, integrated into the “OntoIntegrator” system (with the possibility of the replenishment, editing and calculating support) to address a wide range of applied tasks. The examples of the built-in ontologies are the ontology of the generally accepted abbreviations, the ontology of the markers to annotate the output text according to the results of applied tasks solution, etc.

An important component of the ontological system is the “reflexive core” component of the ontology of models, which allows to model in a system the capacity to reflect. The reflexive core is the system-forming part in building of the ontological system, it contains the links to all types of the concepts and relations in the ontological system, as well as the set of admissible functions of the interpretation, defining the choice of the types of the concepts and relations used, and this information is presented as an internal model of the ontology of models. In other words, in the ontological system there is the knowledge about the structure of the system, organized as the domestic knowledge of the system. The reflexive core can be configured (redefined) by an "OntoIntegrator" system user based on their own interpretation of the concepts of the ontology of the models.

Let's consider one of the typical models of the reflection usage in the ontological system. In Figure 2 is shown an example of reflection, related to a change (redefinition) of the links in the structure of a concept-model. New model preserves the structural properties of the original model, but it changes the semantics of its links. All the transformations are performed on the level of the reflective core.

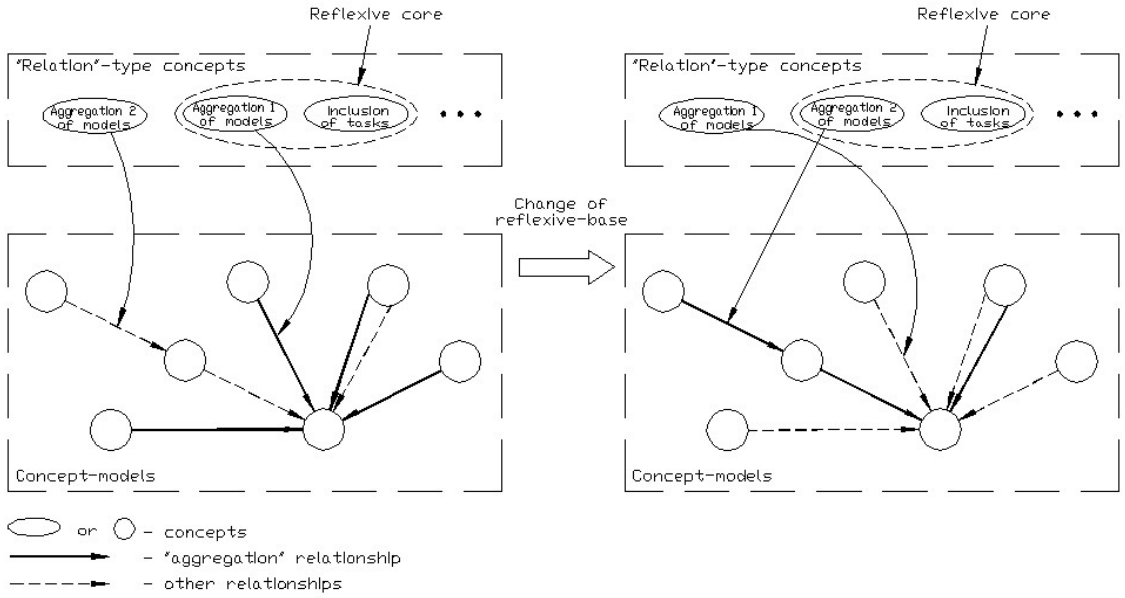


Figure 2. Reflection as a management tool of the structure of the model (tasks)

The OntoMath^{pro} ontology of the professional mathematics

The system of ontological models of the "OntoIntegrator" system contains different ontologies, including applied ontology relating to different subject domains. Replenishment of the applied ontology can be done through the development of the new ontologies by the means of the system and by the conversion of the public ontologies in OWL, XML/RDF formats. In recent years, the authors of the article together with the fellow mathematicians from Kazan Federal University participated in the development of the ontology of professional mathematics OntoMath^{pro}, used in experiments for the classification of mathematical texts, described in section 4.

The main objective of the OntoMath^{pro} ontology is to provide an informative terminological resource for the automatized processing of the electronic professional mathematical publications in Russian language. The OntoMath^{pro} ontology contains definitions of generally accepted mathematical concepts as well as evolving terminology, mainly in the following areas of mathematics: the theory of numbers, the theory of sets, algebra, geometry, mathematical logic, discrete mathematics, the theory of algorithms, mathematical analysis, differential equations, numerical techniques, the theory of probability and mathematical statistics.

The above-mentioned classic and applied mathematical sections are selected as basic from the positions of the traditional mathematics teaching, and on the basis of requirements of word-processing tasks from the collection of mathematical scientific publications of the journal "The Proceedings of higher education institutions Mathematics".

The sources for determination of the concepts of the semantics for the OntoMath^{pro} were: classical textbooks of the relevant sections of mathematics, electronic resources such as Wikipedia and Cambridge Mathematical Thesaurus, scientific articles from the journal "The Proceedings of higher education institutions Mathematics", and

also the professional knowledge of the ontology developers. As OntoMath^{pro} representation languages were chosen OWL-DL/RDFS languages, which are high logical expressive means and algorithmically solvable not only theoretically but practically with the help of such tools of the logic output, as Pellet and Fact++. Let us note the principles of modeling, adopted in OntoMath^{pro} ontology. Regarding "subclass – class" they distinguish two hierarchies in OntoMath^{pro}: hierarchy of the sections of mathematics and the hierarchy of the elements of the mathematical knowledge. The first hierarchy presents taxonomy of the main sections of mathematics. The most fundamental sections are geometry and analysis are developed in more detail, for example, they highlight such sections of geometry as analytical geometry, differential geometry, fractal geometry and others, as well as sections of the analysis that is functional analysis and complex analysis.

The top level of the hierarchy of the elements of the mathematical knowledge is represented by three types of classes: i) basic mathematical concepts (for example, set, operator, function, tensor); ii) root elements of the relevant sections of mathematics (for example, an element of the mathematical analysis theory, the element of the theory of numbers); iii) general scientific notions (for example, theorem, task, method, formula, statement). It is acceptable for a class to be in different hierarchies (the mechanism of a multiple inheritance is supported). For example, the class of "Chernov's Theorem" is a subclass of the "Theorem" class and is a subclass of "An element of the theory of differential equations" class.

Fragment of the hierarchy of the OntoMath^{pro} ontology is in Figure 3.

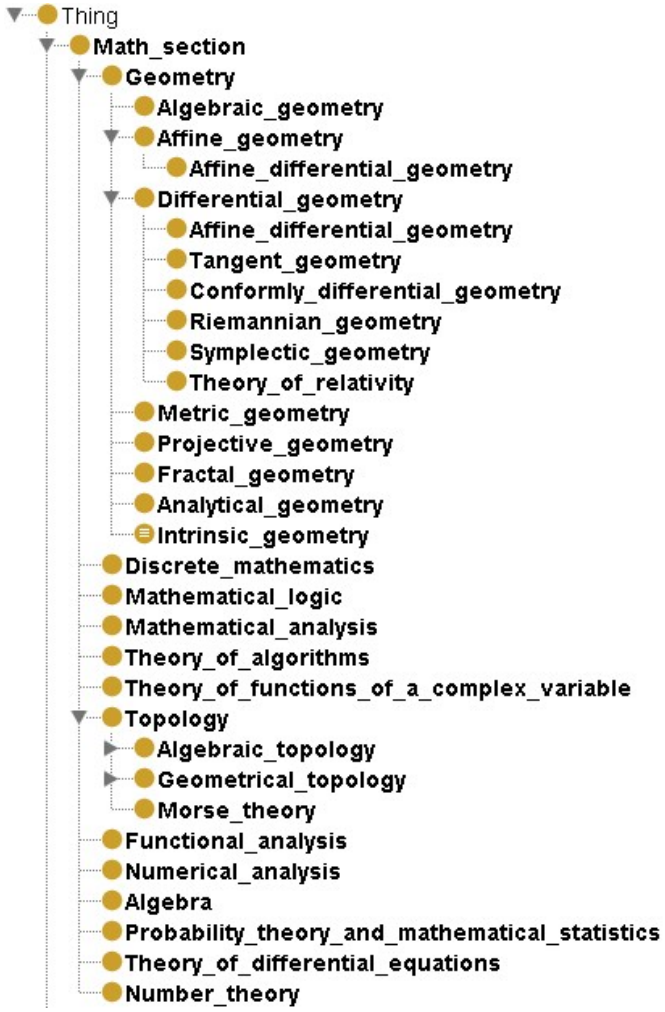


Figure 3. The hierarchy of the sections of mathematics in OntoMath^{pro} ontology

In the OntoMath^{pro} ontology there are defined four types of relations between classes:

- "The subclass - the class" relations (or 'ISA relations");
- Directed object relation of ownership between an element of the mathematical knowledge and the mathematic section;
- Directed object relation of the logical dependency between the elements of the mathematical knowledge, i.e. when one term participates in at least one of the options of another term definition;
- Symmetric object relation of associativity, i.e. it is used when the terms are logically related to each other, but one cannot clearly establish the direction of logical dependency.

The ontology is executed in the popular language OWL-DL of ontology and is available in the file form in a standard RDF/XML format, so it can be used by various modern means of work with ontologies, such as the Protégé editor and Jena software library.

Experiments with the mathematical texts

For experiments on simulation of the mechanism of the attention switch on the basis of the reflexive mechanisms of the "OntoIntegrator" system there was selected a task of the mathematical texts classification based on the mathematical ontology system related to various sections of mathematics. The system of mathematical ontologies was built on the basis of the OntoMath^{pro} ontology of professional mathematics and it included ontologies related to sections of the differential equations, numerical analysis, algebra, the probability theory and mathematical statistics. In the experiments following below, we treat (conditional) mathematical terms as multivalued that relate simultaneously to different mathematical ontologies.

This method of reasoning may be transferred in the same manner to the situation of a real multivaluedness with the same formulation of the classification task.

The article offers a spectral-ontological method of analysis of texts, containing multiple terms, based on the technology of the ontological marking in order to switch automatically the focus of the "Ontointegrator" system on the basis of the reflexive mechanism. Under the ontological range of the text it is understood the distribution of the number of ontological inputs into the text relating to different ontologies, according to its sentences (and/or paragraphs, sections). An example of ontological text spectrum of the lecture courses on differential equations of the 1st order is shown in Figure 4 [Russian University of Chemical Technology (RCTU), 2014]. Our attention is drawn to more dense distribution of the conceptual terms, relating to the problematic area of the lecture, in the beginning of the text for the courses.

In the "Ontointegrator" system there are two possibilities of ontological text processing with the multivalued terms usage:

1) "Simultaneous" ontological marking of the text with the help of several applied ontologies (external from the user and internal in-built), with multiple concepts in unambiguous interpretation;
2) Switch of the aggregation relation underlying the structuring of the applied ontology with the help of reflexive mechanism that allows one to choose specific interpretation of the multivalued concept in the ontology containing its multivalued interpretation.

In order to execute the first way, the system provides a mechanism to include the external applicable ontologies into the process of ontological marking by adding them into the special list (see Figure 5).

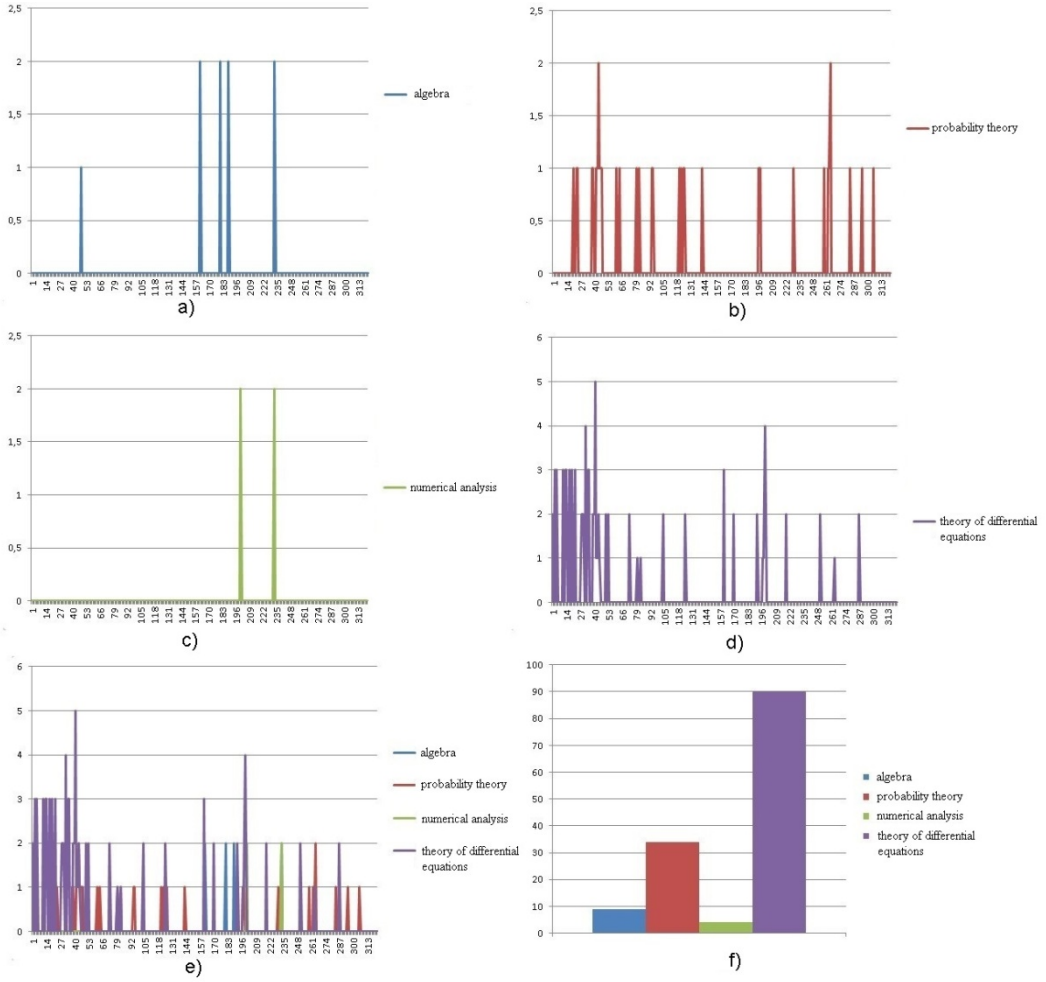


Figure 4. Example of the ontological text spectrum - On the x-axis there is a number of a sentence and on the y-axis is the number of objects of text that are marked by the ontological inputs: a) for ontology "Elements of the theory of algebras"; b) for the ontology “Elements of the probability theory and mathematical statistics"; c) for the ontology "Elements of the numerical analysis theory"; d) for the ontology "The differential equations theory"; e) the combined ontological spectrum for 4 ontologies; f) a histogram of the total number of the marked text objects for each ontology.

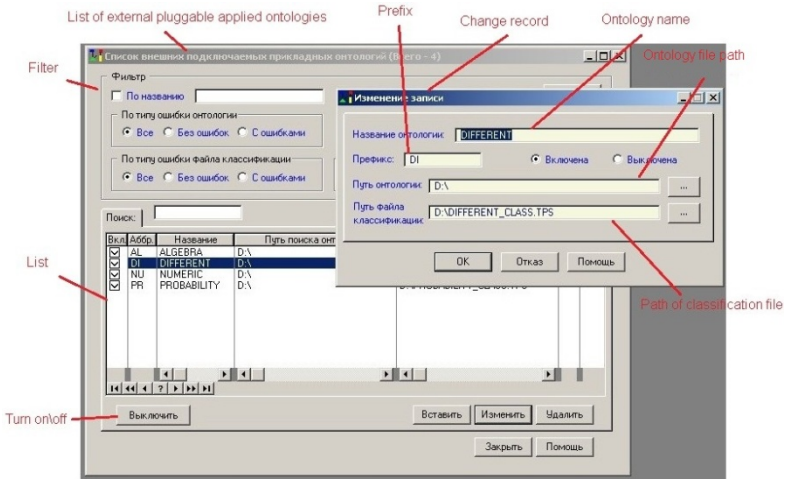


Figure 5. Connection of the external applied ontologies to the process of ontological marking

In the list there are: the place of localization of the connected ontology as well as its classification file corresponding to the current ontology of the models; the mode of ontology usage in the ontological marking: enabled or disabled. Besides, there is an option in the analyzer settings of text to change the global configuration of the ontological marking mode (see Figure 6). These settings determine which ontologies will be used in the process of the ontological text marking.

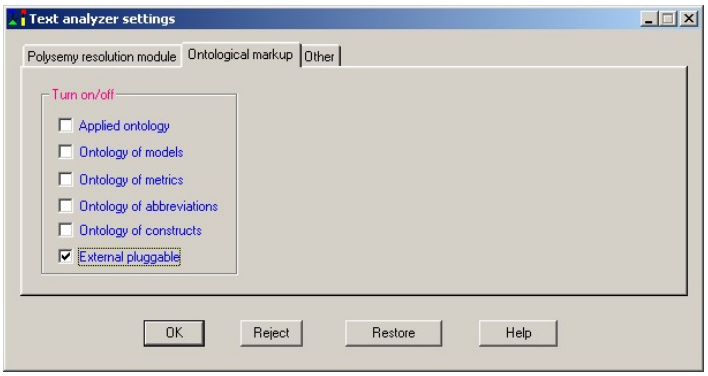


Figure 6. Setting of the ontological marking mode in the text analyzer

The second method is based on reflexive mechanism described in [Nevzorova et. al., 2013].

The ontology of the models of the system of ontologies is constructed as a three-level system of concepts, in which the concepts of the upper level determine the type of the concepts of the lower level (see Figure 7). The type of concept is connected with the model of interpretation and software support for processing of the text. Executive core of the "Ontointegrator" system is the “Processor” module that allows to choose any four concepts of the 2nd level, which are interpreted as a "property", "relation", "link" and "execution", and two concept of the third level out of the group of the concepts of the "relation" type, which are interpreted as the relationship of "aggregation of the models” in the ontology of models and "task integration" in the ontology of tasks planning.

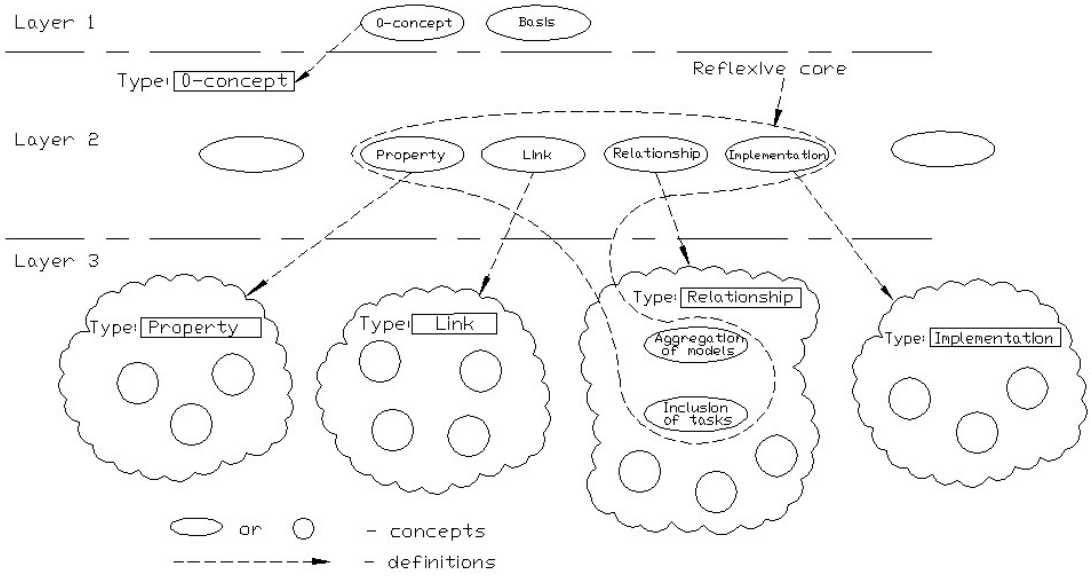


Figure 7. The three-level structure of the ontology of the models

Such a structure of the ontology of the models allows executing the mechanism of reflection by the dynamic substitution of some concepts of the reflexive core by the other during the text processing. In particular, the usage of some relations of the "aggregation of the models" (which can be given a typical for the problem domain names, for example, "algebraic interpretation") allows one to build an ontology, containing multivalued concepts with multivalued interpretation (see Figure 8).

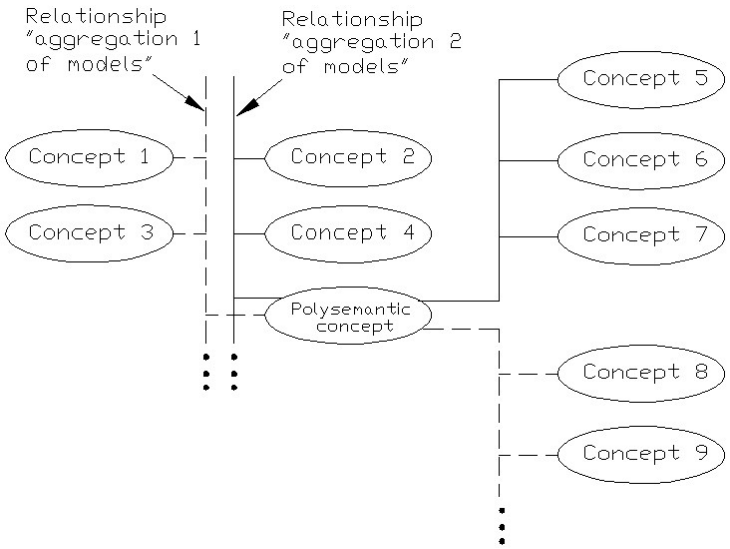


Figure 8. The structure of the ontology, containing multivalued concepts with a multivalued interpretation

Thus, the "Ontointegrator" system, equipped with the mechanisms of analysis and selection of the current text interpretation during its processing can automatically switch the "focus of attention" by a simple replacement in the dynamic mode, in the reflexive core one concept “an aggregation relation of a model N" with another concept of "an aggregation relation of a model M".

In this work there was made an attempts to use for these purposes the ontological range of the text received and analyzed during the process of its processing. For these experiments in the developed OntoMath^{pro} mathematical ontology, containing currently 3178 concepts, 4121 text equivalents (inputs) in the linguistic shell as well as 4321 links on the properties and relations, there were set aside into the independent ontologies four sub-ontologies related to various sections of the mathematical knowledge: "elements of the theory of algebras", "elements of the probability theory and mathematical statistics", "elements of numerical analysis theory" and "elements of the theory of differential equations" (the parameters of the specified ontologies are shown in Table 1).

Table 1

Ontology name	Number of concepts	
	Total	Unambiguous
Elements of the theory of algebras	222	221
Elements of the probability theory and of mathematical statistics	557	548
Elements of numerical analysis theory	469	461
Elements of the theory of differential equations	369	357

The texts for the experiments were taken from the collection of the mathematical articles in XML format [Nevzorova et al., 2013] and were downloaded from the Internet sites of the higher educational institutions. The text processing procedure included ontological marking and building of the ontological spectrum of text similar to figure 4.

The carried out experiments demonstrated stable operation of the reflection mechanism designed in the “Ontointegrator” system as a way of managing of the model structure. Primary analysis of the experimental results showed that the ontological spectra of the analyzed texts even on the basis of four used ontologies are individual (i.e. unique) in the framework of the considered problem area. Various typical sections of articles and lecture courses have a specific structure of the ontological spectra.

It can be assumed that the switching mechanism of the "focus of attention" can be based on the parametric analysis of the ontological spectrum of the current paragraph, and the spectral-ontological method itself can be used in various tasks of the analysis and the processed text classification. It should be noted that the spectral-ontological method of text analysis requires a more formal study and further development.

The usage of the reflexive mechanism of the "Ontointegrator" system allows carrying out the analysis of the text by the adequate to its content interpretations of multivalued concepts by simple replacement in the dynamic mode relations of the "aggregation models” that underlie the building of the ontologies with different interpretation of the multivalued concepts.

Conclusion

In the article was considered the approach to modeling of the cognitive ability to manage the focus of attention in text processing, which is based on reflexive mechanisms developed in the “OntoIntegrator” system. The process is carried out by a multi-level system of ontologies, including different types of ontologies: the applied ontologies, the ontology of models and ontology of tasks planning. An important component of the ontological system is a component of the ontology of models that is «the reflexive core”, which allows simulating in the system the capacity to reflect.

In order to switch automatically the focus of the "Ontointegrator" system on the basis of reflexive mechanism there was suggested a spectral-ontological method of texts analysis, containing multiple terms, based on the technology of ontological marking. The carried out experiments showed the stable work of the reflexive mechanism. The conclusion is drawn that the parametric analysis of the ontological spectrum of text can be used for switching of the focus of attention, and for solving of the problems of analysis and classification of the processed texts.

Acknowledgements

This work is executed with financial support from RFBR (Russian Foundation for Basic Research), project # 12-07-00550.

Bibliography

[Gippenreyter, 2001] The psychology of attention / Eds. Yu.B. Gippenreyter, V.Ya. Romanov, M., 2001. 858 c. (Series: Chrestomathy of psychology) In Russian.

[Nevzorova et al., 2013] Olga Nevzorova, Nikita Zhiltsov, Danila Zaikin, Olga Zhibrik, Alexander Kirillovich, Vladimir Nevzorov, Evgeniy Birialtsev, Bringing Math to LOD: A Semantic Publishing Platform Prototype for Scientific Collections in Mathematics // The Semantic Web – ISWC 2013. Lecture Notes in Computer Science. Volume 8218, 2013, pp. 379-394.

[Nevzorova et. al., 2013] Nevzorova O.A., Nevzorov V.N. Model of ontological system with reflexive core // Proc. of the 15th International Workshop on Computer Science and Information Technologies (CSIT' 2013), Vienna-Budapest-Bratislava, Sept. 15-21, 2013, Vol. 1, P. 7-12.

[Nevzorova, 2007] Nevzorova O.A. Ontolinguistic systems: technologies of interaction with applied ontology, Memoirs of Kazan State University, Physics and mathematics, 2007, Vol. 149, book 2. P. 105-115.

[Russian University of Chemical Technology (RCTU), 2014] Lectures on theory of differential equations of 1st order. Department of Higher Mathematics of Russian University of Chemical Technology named after D.I. Mendeleev [E-resource] http:\\kvm.muctr.ru.

Authors' Information

Olga Nevzorova *– Vice-director of Research Institute of Applied Semiotics of Tatarstan Academy of Sciences, Kazan Federal University. P.O. Box: 420111, Bauman str., Kazan, Russia; e-mail: onevzoro@gmail.com*

Major Fields of Scientific Research: Natural language processing, Artificial intelligence

Vladimir Nevzorov *– Associated Professor the Department of Information Technologies of Computer Design, Kazan National Research Technical University named after A.N. Tupolev. P.O. Box: 420111, K. Marks str., 10, Kazan, Russia, e-mail: nevzorovvn@gmail.com*

Major Fields of Scientific Research: Natural language processing, Artificial intelligence

DYNAMIC INTEGRATED EXPERT SYSTEMS: THE EVOLUTION OF AT-TECHNOLOGY WORKBENCH

Galina Rybina, Victor Rybin

Abstract*: The current state and trends in the area of dynamic intelligent systems are analyzed. The first experimental results obtained in the development of certain provisions and components of the task-oriented methodology for the integrated expert systems construction during the transition to unformalized problems in dynamic areas of concern solving are considered. State and tendencies of evolution of modern development tools for static and dynamic expert systems development are being analyzed. New facilities of unique Russian instruments for support of integrated expert systems development – the AT-TECHNOLOGY workbench – connected with dynamic integrated expert systems development are in question.*

Keywords*: dynamic intelligent system, dynamic integrated expert systems, task-oriented methodology, AT-TECHNOLOGY workbench, development tools real time, simulation modeling, temporal reasoning, temporal reasoner.*

ACM Classification Keywords*: I.2.1 Applications and Expert Systems; I.2.5 Programming Languages and Software*

Introduction

In the context of the analysis of trends in modern tools for intelligent systems development in general, it should be noted that at present the highest growth rates are observed in the area of development tools (DT) for support the development of static and dynamic expert systems.

As it was shown in [1], this is due, primarily, that the scientific and commercial progress in the development of static expert systems (ESs) in the mid-80's and integrated expert systems (IESs) in the late 90's, led to increasing demand and expansion of industrial use and development of applied ESs and IESs for a wide class of problems of real practical importance and complexity, including the dynamic areas of concern (AoCs).

To these dynamic IESs, which function, as a rule, in real time (IESs RT), there are imposed some new requirements such as [1]: performance, storage and analysis of time-varying data from external sources, the implementation of concurrent temporal reasoning about multiple asynchronous processes (tasks), including those with limited resources (time, memory), the possibility of modeling the external world and its various states, etc. That's why in the architecture of any IES RT (compared with static ES) there are included some special components that simulate the external environment and allow to interact with hardware in real time. It significantly complicates the process of developing such systems and requires highly skilled developers and experts [1, 2]. So a robust DT, providing the development of the dynamic ESs and IESs, capable to operate in dynamic AoCs with possible adjustments in strategies for reasoning and updating the knowledge base directly in a process of reasoning are needed.

In [1, 2, 3] there is presented the most complete overview of domestic developments related to the creation of dynamic IESs for electro-physical complex management, diagnostics of complex technical systems, prelaunch launch vehicles control, radio-ecological monitoring of areas adjacent to the nuclear power plants, atmosphere air monitoring, forecasting of on-board micro-computers radiation damage, etc. Description of some foreign dynamic ESs to control a continuous production process in chemistry, pharmacy, production of cement, food products,

aerospace research, transportation and processing of oil and gas, management of nuclear and thermal power plants, finance operations, etc. is given in [2, 4].

Analyzing the success of Russian scientific schools in the area of dynamic intelligent systems, it should be noted diverse range of theoretical and methodological research related to the creation of the theory of constructing a specific class of dynamic intelligent systems based on the rules [5], the results obtained in simulation of time (temporal) reasoning for intelligent decision support systems for real-time [6], the cycle of research and development in the field of dynamic systems under the guidance [7], as well as a number of other interesting theoretical works [8, 9]. However, in general, the problem of creating powerful software tools to support the development of dynamic intelligent systems, in particular the dynamic ESs and IESs, practically, hasn't been intended.

A new research project related to the development of the instrumental base of the AT-TECHNOLOGY workbench to the level of the modern DT for support the development of Dynamic IESs is in question of this paper.

Evolution features of instrumental base for dynamic expert systems at the turn of XX-XXI centuries

One of the first foreign tools for dynamic ESs development, which was intended for character computer Symbolics and was named Picon, was created by Lisp Machine Inc (USA) in 1985. The success of Picon led to that group of leading developers had founded a company Gensym and released DT G2 based on evolution of ideas used in Picon in 1988. Today the world is widely using version 8.3 of G2, and a number of problem-oriented DT, based at the G2 core for developing applications for special AoCs and special classes of problems [10, 11].

2-3 years after Gensym Corp. some other companies begun to create their own DT, the most famous of which are: RT Works (Talarian, USA), COMDALE/C (Comdale Techn., Canada), COGSYS (SC, USA), ILOG Rules (ILOG, France). According to the data given in [2,4], the objective comparison of the two most advanced systems - G2 and RT Works, which took place through the development of the same application by two organizations - NASA (USA) and Storm Integration (USA), showed a significant superiority of G2 (Gensym Corp.).

In general we can say that today in the field of instrumental base for support the development of dynamic ESs and IESs there is not such a great diversity, such as for static ESs, where there are hundreds of DT for different applications, with different power and cost (from several hundred dollars to several tens of thousands of dollars). Comparative analysis of foreign commercial DT for dynamic ESs [2], as well as experience of prototyping Dynamic IESs [2, 12], accumulated in the laboratory of Intelligent Systems and Technologies of Department of Cybernetics of National Research Nuclear University MEPhI (Moscow Engineering Physics Institute), has shown significant advantages of G2 over other commercial DT of this class (RTWorks, R * Time, L * Star, TCD Expert, etc.), features of which do not exceed 50% of G2 features.

The high complexity and the difficulty of DT for the ESs and IESs development, the lack of funding and legal basis for protecting intellectual property rights are serious impediments to the creation of Russian DT appropriating to the level of G2 (Gensym Corp.). Significantly reducing or even exceeding the far gap between world and Russian level of solving tasks of such complexity and importance can be overcame by using the experience of developing and using powerful tools to build static IESs - a third generation of the AT-TECHNOLOGY workbench [1].

Some information about the AT-TECHNOLOGY workbench

Since the mid 90's in the laboratory of Intelligent Systems and Technologies of Department of Cybernetics of NRNU MEPhI research and development has been conducting, within which was created a new generation of DT

– complex AT-TECHNOLOGY for the automated development of software of IESs - one of the most complex class of intelligent systems, theory and technology of construction of which are described in detail in [1]. The instrumental software the AT-TECHNOLOGY workbench, transformed during this period to several technology generations of WorkBench type (i.e., integrated environment that includes tools of various levels of complexity, the selection administration of which is implemented by system as well as by knowledge engineers), supports the original task-oriented methodology proposed by G. Rybina [1].

This toolkit is a coherent set of tools for computer aided Design of IESs at all stages of the life cycle, also providing elements of "intellectualization" of the process of developing applied IESs based on knowledge about typical design procedures for the development of systems and their components for specific classes of problems and AoCs. Today the AT-TECHNOLOGY workbench is a powerful workstations for knowledge engineers, as well as undergraduate and postgraduate students studying the theory and technology of IESs construction. Since 1995 the complex has been used in the educational process of NRNU MEPhI and other universities.

More information about basic features and architecture of the AT-TECHNOLOGY workbench (including base and web-oriented version) can be found in [1, 13 - 17].

The evolution of instrumental base of the AT-TECHNOLOGY workbench for dynamic IES development

Long-term experience of using of the AT-TECHNOLOGY workbench for support of development of IESs on the basis of the task-oriented methodology [1] has shown that the class of problems where AoC is dynamic meets often enough. Therefore a cycle of researches directed on development of functionality of a the AT-TECHNOLOGY workbench (base and web versions) for supporting of development of dynamic ESs with scaled architecture, i.e. IESs, functioning in real time, began some years ago [1, 13].

In developing of IES RT as well as any other dynamic system, it is necessary to consider behavior of system in time i.e. to establish interrelation between parameters during the different moments of time, therefore as shown in [1, 13, 17], the integral stage of IES RT development is the test of components of created system with simulation model (SM) that provides possibility of an estimation of working capacity of dynamic IES, without expensive introduction of system. Methods and the means of simulation modeling focused on the description of a concrete class of SM are most often applied to modeling of conditions of complex discrete systems.

From a position of the task-oriented methodology and the toolkit supporting it − a the AT-TECHNOLOGY workbench, the greatest interest is represented by problems of integration traditional ESs with SM in architectures of the applied Dynamic IESs constructed on principles of deep integration of components [1], because in this case conceptual unity of all used approaches, models and methods as during development of AoC model and environment model, and in the course of searching of the decisions received in the conditions of temporary restrictions and other is provided.

As simulation modeling is the widest applied approach of modeling behavior of various systems in time now there is variety domestic and foreign DT, allowing to develop and make experiments on SMs, however the greatest attention, proceeding from the purposes of the given work, the concept of the environment of the imitating modeling RAO [18] intended for development and debugging of SMs in RAO language [18], having unlike other similar systems the most developed theoretical base. Certainly, it is not provided the full requirements of deep integration of components ES and SM in any system, nevertheless, using and development of the RAO-method, realizing the task-oriented approach to construction of SMs, is represented justified and expedient regarding development of architecture of a the AT-TECHNOLOGY workbench by the subsystem of simulation modeling of an external environment. Therefore on the basis of the analysis of base principles of construction of

corresponding components of systems G2, RAO, etc. in the case of development of functionality of a the AT-TECHNOLOGY workbench for support of construction IESs RT, the tools providing construction of SMs of complex discrete systems and making experiments on these models have been realized first of all.

In the architecture of a prototype of a subsystem of simulation modeling [13] functionality of the developed tools is divided between two global modules - the module of development of SM, which provides development support out and debugging of SM and other functions demanding the visual interface, and the module of calculation of SM, which provides methods for calculation of conditions SM in each step of discrete time during imitational experiment. The module of development of SM comprises a number of the components providing various aspects of construction and using of model, namely:

- The models manager, which provides version control of developed SM and allowing to create new model on the basis of existing, to store various sets of models in the form of treelike structure;
- The models editor, which intended for construction of SM by the visual editor tools, allowing to create objects, setting their properties and attributes, and also to establish the connection between objects of model in a graphic mode;
- The component of visualization allows to draw animation shots and moving between them, and each shot of animation corresponds to a condition of SM on a certain step of discrete time;
- The analysis component, which supports the analysis of results of experiments on SM (reviewing of detailed reports of experiments, and also the various reports constructed by results of experiment that allows not only to draw a conclusion on success of experiment but also to reveal lacks of modeled system);
- The module of calculation of SM contains some components which provides different aspects of calculation of conditions of model, namely;
- Component of calculation of conditions of SM, counting a new condition of model in a certain step of discrete time on the basis of a model condition on the previous step of time taking into account the carried out operations initiated by IES or other operations of model;
- Component of generation of the probabilistic events, which realizes mechanism simulating random indignations in modeled system (the given mechanism initiates performance of certain operations of SM during the random moments of discrete time according to the law of distribution of probabilistic variables);
- The component of trace, which provides gathering statisticians and other necessary information on a condition of SM on each step of discrete time for the purpose of subsequent using by a component of the analysis of results for constructing various reports and schedules by results of experiment.

In the context of integration of tools for development of SM with corresponding components of base version of the AT-TECHNOLOGY workbench two main problems. Their decisions will allow integrating tools for development of SM into structure of the AT-TECHNOLOGY workbench.

The first problem is reduced to interaction maintenance between universal AT-REASONER of the complex [1, 13] and a subsystem of simulation modeling so that in the course of making reasoning, AT-REASONER could initiate actions (operation) over SM, which changes a condition of model for a certain interval of discrete time. Current version of the AT-REASONER provides a wide spectrum of functionality, which, in particular, can make calls of functions of special components, using COM-technology. It allows developing the interaction mechanism between AT-REASONER and a subsystem of simulation modeling, having realized the necessary interface.

The second problem is a sharing of working memory by a subsystem of imitational modeling and other components of the AT-TECHNOLOGY workbench that is possible by development of a special component - a gateway which synchronizes conditions of working memory of IES RT and conditions of SM into each step of discrete time during carrying out of experiment with SM. The basic functions of the given component are:

- Maintenance of the unified program interface, allowing to change a condition of working memory of IES RT by the subsystem of simulation modeling;
- Scanning of working memory in each step of discrete time and transferring information about the changes brought in working memory by components of the AT-TECHNOLOGY workbench to a subsystem of simulation modeling.

Integration of the developed tools for development of the SMs with base subsystems of a the AT-TECHNOLOGY workbench is rather not trivial problem, therefore for these purposes the special program stand has been created. It provides carrying out of debugging of integration mechanisms of various components of the complex [8].

Researches, that has been made at the program stand, has shown possibility in a mode of real time to transfer data about changes of the conditions of SM in working memory of the AT-REASONER, and also to receive and execute influences on the model, initiated by the AT-REASONER during the reasoning.

Therefore a following important stage of development of the AT-TECHNOLOGY workbench for support of the Dynamic IESs development is an expansion of possibilities of the AT-REASONER to make temporal reasoning which is connected with creation of special temporal reasoner. In the results of comparison of various models of time representation by several criteria the model of Allen's interval time logic [19] has been chosen (due to its possessing balanced enough characteristics by all criteria and having the developed mathematical methods). The preliminary experimental researches connected with modeling of architecture of a prototype of the temporal reasoner using of G2 system, have shown efficiency of application of Allen's interval logic proceeding from the purposes of the given work.

More information about temporal reasoning implementation in AT-TECHNOLOGY workbench can be found in [17, 19 - 28].

Conclusion

Further evolution of the AT-TECHNOLOGY workbench tools to support development of Dynamic IESs (so-called "dynamic version") involves the empowerment of the AT-REASONER for the implementation of temporal reasoning [17, 28], the development of tools to get data from external sources in real time, and implementation of other mechanisms for constructing Dynamic IESs within the confines of the task-oriented methodology.

Acknowledgements

This work was supported in part by the Russian Foundation for Basic Research under Grant No. 12-01-00467.

Bibliography

1. Рыбина Г.В. Теория и технология построения интегрированных экспертных систем. М.: Научтехлитиздат, 2008. − 482с.
2. Рыбина Г.В., Паронджанов С.С. Технология построения динамических интеллектуальных систем: Учебное пособие. М.: НИЯУ МИФИ, 2011. − 240с.

3. Рыбина Г.В. Практическое применение задачно-ориентированной методологии построения интегрированных экспертных систем (обзор приложений в статических и динамических проблемных областях)//Приборы и системы. Управление, контроль, диагностика. 2011. № 12. С. 10-28.

4. Попов Э.В., Фоминых И.Б., Кисель Е.Б., Шапот М.Д. Статические и динамические экспертные системы. Москва: Финансы и статистика, 1996. − 320с.

5. Осипов Г.С. Методы искусственного интеллекта. – М.: ФИЗМАТЛИБ, 2011. – 296с.

6. Еремеев А.П., Троицкий В.В. Концепции и модели представления времени и их применение в интеллектуальных системах//Новости искусственного интеллекта. 2004. №1. С. 6–29.

7. Стефанюк В.Л. “Динамическая экспертная система и логическая проблема фрейма,” Международная конференция по искусственному интеллекту(AIS’07/CAD - 2007). Москва: Физматлит, 2007. Т.2. С. 107-113.

8. Плесневич Г.С. “Метод аналитических таблиц для логики событий” Труды Международной конференции "Интеллектуальное управление: новые интеллектуальные технологии в задачах управления (ICIT'99)", Переславль-Залесский: ИПС РАН, 1999.

9. Виньков М.М, Фоминых И.Б. “Темпоральные немонотонные логические системы: взаимосвязи и вычислительная сложность,” Искусственный интеллект и принятие решений. 2008. № 4. С. 19-25.

10. G2 Platform: http://www.gensym.com/?p=what_it_is_g2

11. Инструментальные средства G2 создания экспертных систем реального времени: http://infogoz.vimi.ru/otct/Infogoz/KN6/138.htm

12. Рыбина Г.В., Мозгачев А.В., Со Ти Ха Аунг Построение динамических интегрированных экспертных систем на основе задачно-ориентированной методологии//Информационно-измерительные и управляющие системы. 2012. №8. Т.10. С. 4-12.

13. Рыбина Г.В. Инструментальные средства для построения динамических интегрированных экспертных систем: развитие комплекса АТ-ТЕХНОЛОГИЯ // Искусственный интеллект и принятие решений. 2010. №1. С. 41-48.

14. Рыбина Г.В. Теоретические основы задачно-ориентированной методологии построения интегрированных экспертных систем//Приборы и системы. Управление, контроль, диагностика. 2011. №7. с. 27-42.

15. Рыбина Г.В., Демидов Д.В., Шанцер Д.И., Мозгачев А.В. Динамические интегрированные экспертные системы: новые возможности инструментального комплекса АТ-ТЕХНОЛОГИЯ//Информационно-измерительные и управляющие системы. 2011. №6. С. 7-15.

16. Рыбина Г.В., Шанцер Д.И., Мозгачев А.В., Блохин Ю.М. Динамические интеллектуальные системы на основе интегрированных экспертных систем//Приборы и системы. Управление, контроль, диагностика, 2012 №5 С. 13-19.

17. Рыбина Г.В., Мозгачев А.В., Паронджанов С.С., Со Ти Ха Аунг Динамические интегрированные экспертные системы: методы представления и обработки темпоральных знаний//Приборы и системы. Управление, контроль, диагностика, 2013. №6. С.63-73

18. Емельянов В.В., Ясиновский С.И. Введение в интеллектуальное имитационное моделирование сложных дискретных систем и процессов. Язык РДО. − М.: АНВИК, 1998. – 533с.

19. Allen J. F. Maintaining knowledge about temporal intervals//Communications of the Association for Computing Machinery. 1983. Vol. 22. P. 832-843.

20. Hayes P., Allen J. A common-sense theory of time//Proceedings of the 9th International Joint Conference on Artificial Intelligence. - Los Angeles, CA. 1985. – P.528-531.

21. Ladkin P. Time representation: A taxonomy of interval relations//Proceedings AAAI-86. Philadelphia, PA. 1986, 360-366.

22. Hayes P., Allen J. Short time periods//Proceedings of IJCAI-87. - Los Angeles, CA. 1987. - P. 981-983.

23. Ladkin P. Constraint reasoning with intervals: a tutorial, survey and bibliography//ICSI TR-90-059. November 1990.

24. Davis W., Carnes J. Clustering temporal intervals to generate reference hierarchies//Proceedings of the 2nd International Conference on Principles of Knowledge Representation and reasoning. - Morgan Kaufman. 1991. – P. 111-117.

25. Morris R., Khatib L. An interval-based temporal relational calculus for events with gaps//Journal of Experimental and Theoretical Artificial Intelligence. 1991. N. 3. P.87-107.

26. Nebel B., Bürckert H.-J.: Reasoning about Temporal Relations: A Maximal Tractable Subclass of Allen's Interval Algebra//Journal of the ACM. 1995. N. 42. P. 43–66.

27. Van Beek P., Manchak D.W. The design and experimental analysis of algorithms for temporal reasoning//Journal of Artificial Intelligence Research. 1996. N. 4. P. 1–18.

28. Рыбина Г.В., Рыбин В.М., Со Ти Ха Аунг Некоторые аспекты применения имитационного моделирования в динамических интегрированных экспертных системах//Приборы и системы. Управление, контроль, диагностика. 2014. №3. С.35-45.

Authors' Information

Galina Rybina *– Doctor of Technical Science, Professor cybernetics department of* NRNU *MEPhI. RF President education award winner. Accents on intelligent systems and technologies, static, dynamic and integrated expert systems, intelligent dialogue systems, multi-agents systems, workbenches.*

E-mail: galina@ailab.mephi.ru

Victor Rybin *- Doctor of Technical Science, Professor department of Automation of NRNU MEPhI. Accents on automation and electronics, electro physical complex, automatic control system, intelligent control systems, dynamic intelligent systems.*

E-mail: vmrybin@yandex.ru

IMPACT OF SUBPIXEL PARADIGM ON DETERMINATION OF 3D POSITION FROM 2D IMAGE PAIR

Lukas Sroba, Rudolf Ravas

Abstract*: The idea of subpixel feature detection is widely used in image processing area in these days and it has a high significance in many practical applications. This paper deals with impact of subpixel paradigm on accuracy of 3D coordinates determination using information related to corresponding 2D image pairs. In the other worlds, there is a study how the subpixel corner points detection could influence the accuracy of 3D reconstruction. For that reason our work contains this comparison for image pairs having various mutual position and resolution using data from given datasets. This contribution could answer the question if the paradigm of subpixel detection could be useful in VSLAM methods and other practical tasks where the precision is a key.*

Keywords: *subpixel corner detection, 3D scene reconstruction, accuracy comparison*

ACM Classification Keywords: *I.4.8 Scene Analysis - stereo*

Introduction

The area of corner point's detection is well known and very often used in various areas of image processing and computer vision. The tasks such like motion tracking, object detection and recognition, robot navigation, stereo matching or 3D modeling are the illustrative examples.

There is no exact definition what the corner point is, but except many other statements it could be: point where at least two edges are intersected, point having the smallest radius of curvature or point around which is high change of brightness intensity in all directions.

As it is known, the smallest part of an image is a pixel. We usually cannot access information "between" pixels. But there is possibility to use some mathematical techniques to interpolate or approximate the brightness intensity and find the chosen features in subpixel accuracy.

One of the applications where the localization of corner points is crucial is 3D scene reconstruction. For that there is possible to use the theory of epipolar geometry and fundamental matrix [Hartley, 2003].

The accuracy improving character of subpixel detection in case of homography determination is slightly described in [Sroba, 2013].

In the next sections there will be the basic principles of pixel and subpixel corner detection described very briefly. Also the theory of fundamental matrix and its relation to obtaining the 3D coordinates will be slightly mentioned.

Pixel and subpixel corner detection

Many corner detectors were invented over the years and the Harris corner detector is one of the most famous. This detector was first time mentioned in [Harris, 1988]. The main idea is to find the minimum of brightness intensity difference between chosen parts of an image (marked as W) and shifted parts of image in all directions. There is first-order Taylor series approximation for that purpose used. First step is determination of matrix M as it is shown in (1).

$$M(x,y) = \sum_{W} \left(\begin{bmatrix} I_x \\ I_y \end{bmatrix} \cdot \begin{bmatrix} I_x & I_y \end{bmatrix} \right) \quad (1)$$

The variables I_x and I_y represent the approximations of derivation (also known as differences) in horizontal and vertical direction. As soon as this matrix M is calculated for every pixel in image, the next step is the definition of values contained in matrix C. This matrix has the same size as tested image and is also known as cornerness map. There is a lot of ways how to calculate this matrix, for instance [Shi, 1994]. Last step is looking for the elements in matrix C having the highest values. These points are after global and local thresholding marked as found corner points.

As it is obvious, this algorithm can be used to find corner points in pixel accuracy. Here it will be shortly mentioned two ways how to obtain the subpixel coordinates of corner points. These two approaches were also implemented in our comparison. For some other algorithms of subpixel corner detection see [Sroba, 2012].

Both methods using the previously found corner point in pixel accuracy as initial step. Once this point was detected, its position according of first approach [Brown, 2005] is refined to subpixel accuracy by fitting the 2D quadratic surface to the corner strength function in the local neighbourhood and its maximum is found. The equation of surface is following:

$$h(x,y) = ax^2 + bxy + cy^2 + dx + ey + f \quad (2)$$

When the coefficients are calculated, the assumption that the maximum of corner map corresponds to the first derivation of this function equals to zero could lead us to the final corner point subpixel coordinates very easily.

The second approach [Rea, 2009] is basically very similar to previous one. The only difference is that the subpixel shifts are determined for x and y direction separately using quadratic curve equation:

$$h(x,y) = ax^2 + bx + c \quad (3)$$

The final subpixel corner point position is combination of both shifts and is calculated by using the same assumptions as before.

3D scene reconstruction

One of the ways how to obtain 3D world coordinates from 2D stereo images is to use the principles of epipolar geometry. The first step is to determine the fundamental matrix. This matrix has 3x3 size and describes a line (*epilolar line*) on which the corresponding point on the other image must lie. The 8-point algorithm [Sur, 2008] could be used for calculation. Based on this algorithm, there is necessary to find at least 8 pairs of corresponding points in both images. It there is more corresponding points detected and system is overdetermined, the solution can be obtained using the method of least squared for example.

Next step is the decomposition of fundamental matrix into rotation R matrix and translational t vector [Wang, 2000]. It gives us the information about relation between 3D world coordinates and 3D camera coordinates and it is also known as extrinsic camera parameters. If the 3D position could be localized, the intrinsic camera parameters also have to be stated. These parameters can be found by camera calibration [Zhang, 2004]. The relation between image and 3D coordinates is shown in (4).

$$\begin{bmatrix} u/w \\ v/w \\ w \end{bmatrix} = [K] \cdot [R \mid t] \cdot \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \quad (4)$$

The symbols *u/w* and *v/w* represents 2D image pixel coordinates, *K* is 3x3 intrinsic camera matrix, *R*|*t* is 3x4 extrinsic camera matrix and *X*, *Y*, *Z* are 3D world coordinates of chosen point.

Because the 3D world coordinates are the ones we need to determine, for this purpose are triangulation methods used, the direct linear transformation (*DLT*) algorithm [Hartley, 2003] for example.

Experimental tests

As it was already said, this paper deals with comparison of pixel and subpixel corner detection in case the determination of 3D position from image pair was taken into account. For that reason were the images from two datasets tested [Strecha, 2006; Strecha, 2004]. The direct web links to these datasets and more information are stated in acknowledgement section.

There is a lot of ways how to compare and describe the accuracy of 3D reconstruction and camera calibration. We decided to use reprojection error criterion (dataset 1) and also direct comparison of found and ground truth translation vectors (dataset 2). Both comparisons are described in this section.

Let's start with first dataset and first comparison. One of the reasons why this dataset was chosen is because it contains chessboard segments where the *X*-corners are easily detected.

The whole process of testing was following: Firstly the pair of images containing the same scene was chosen from dataset. The example of tested image is shown in Figure 1. These two images have different angle of view and scene distance. Then the corner detection was applied and corresponding points in both images were found. We used not only traditional pixel but also both mentioned subpixel approaches. A next step was stating the fundamental matrix and extrinsic camera matrix. The intrinsic camera matrices were included to tested dataset. Last step was to determine the 3D world coordinates using triangulation as it was described in previous section.

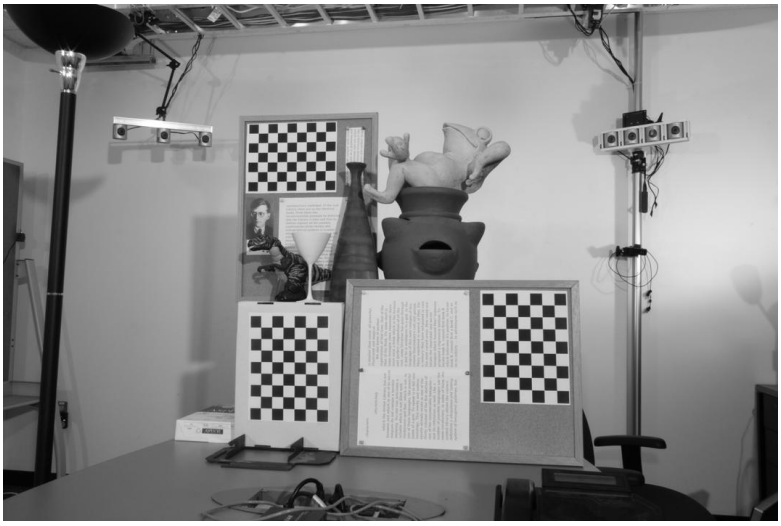


Figure 1. The example of tested image from dataset 1

As it was said, in this case, we decided to use reprojection error criterion. By definition the reprojection error is geometric error corresponding to the image distance between a projected point and a measured one. It is used to quantify how closely an estimate of 3D point recreates the point's true projection. In case of absenting ground truth information such as translation vector or rotation matrix, it is reasonable criterion for solution optimality evaluation.

In our case it means we have determined the 3D position of detected corner points (also extrinsic camera parameters were stated) and then we have reprojected these found 3D points back into 2D camera pixel coordinates. The example of reprojection the 3D points into 2D camera planes is shown in Figure 2.

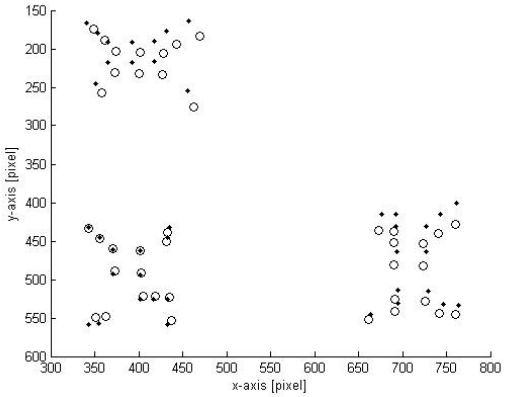


Figure 2. The example of reprojection the 3D points into 2D camera plane

Let's say a little bit more about specifics of our experiment. We have chosen 24 tested images from dataset, where the both halves of this set showing the same scenes only the resolution is different. First half (12 images) had resolution 1068x712 pixels and other half (12 images) had resolution 4272x2848 pixels. These two subsets were divided into the 6 pairs, where two of them had relatively large mutual longitude angle (roughly 90°), another two had relatively medium longitude angle (roughly 30°) and last two pairs had relatively small longitude angle (roughly 10°).For every from given pairs were 10 corresponding points detected and the reprojection error tests was performed as it was mentioned before. The results were statistically analysed and listed in the table.

In case of comparison based on dataset 2 the principle of the test is basically the same. There were again the image pairs for testing chosen, 10 corresponding points between images found and pixel and both subpixel algorithms tested. In the contrary of first comparison, this dataset contains ground truth information (translation vector and rotation matrix for every image) and the images are not consisting from chessboard segment. The examples of tested images from this dataset are shown in Figure 3.

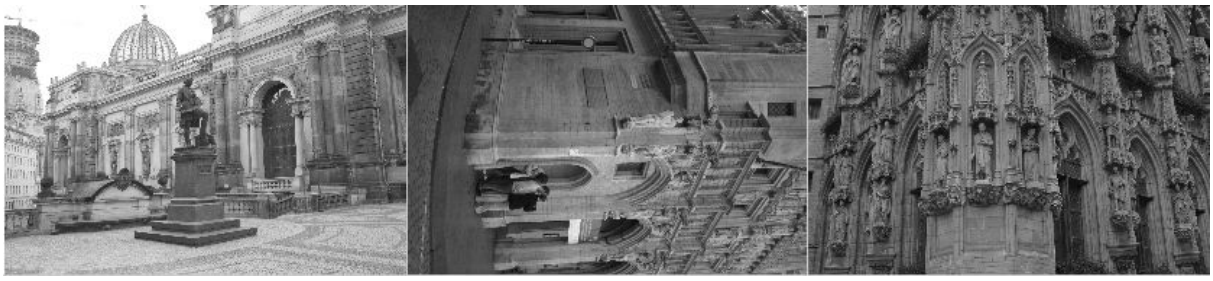


Figure 3. The example of tested images from dataset 2

Because of possibility to directly compare ground truth information from this dataset, we decided to evaluate our obtained and original ground truth translation vectors.

For this comparison we have chosen 9 tested image pairs (3 pairs for every from 3 different scene). These images have resolution 3072x2048 pixels (6 pairs), respectively 2048x1360 pixels (3 pairs) and different angle of

view and scene distance. For every from these pairs were relative translation vectors computed and compared with dataset data. The results were statistically analysed and listed in the table.

Experimental results

There are two tables presented in this section. In the first one is corresponding with first comparison and another one with second comparison.

Let's describe the values in first table a little bit more in detail. As you can see all results related to traditional pixel detection method have values equal to one. Subpixel method using quadratic curve is marked as *method A* and method using quadratic surface as *method B* respectively. For every single tested image the arithmetical mean (*AM*) and standard deviation (*SD*) of reprojection error (2D Euclidean distance between detected and reprojected points) were calculated. Then the ratio of subpixel and pixel results (arithmetic mean and standard deviation) for particular image was stated. Because there were multiple images in the same category (number of detected points, method, resolution, angle), these values were averaged. So there are averaged ratios of reprojection error arithmetic means and reprojection error standard deviations for particular parameters listed in Table 1.

Table 1. Arithmetical mean and standard deviations ratios of reprojection errors

resulted ratios	method	small resolution			high resolution		
		large angle	medium angle	small angle	large angle	medium angle	small angle
AM ratio	pixel	1	1	1	1	1	1
	subpixel A	0.74	0.17	1.64	1.19	0.79	0.79
	subpixel B	0.74	0.51	1.50	1.29	1.77	0.57
SD ratio	pixel	1	1	1	1	1	1
	subpixel A	0.75	0.16	1.78	1.21	0.90	0.81
	subpixel B	0.74	0.65	1.27	1.31	3.32	0.60

As it is possible to see, subpixel corner detection approaches could not improve the results in every testing case we consider. But it is obvious, that the subpixel detection in case of small resolution and large or medium angle improved the reprojection error for every tested number of detected corners. On the other side, if we consider the small mutual longitude angle between the images, the results are worse if subpixel detection is involved. Using images with higher resolution, the results vary a little bit more so it is not possible to unambiguously see the influence of subpixel detection.

The notation in case of second comparison and Table 2 is the same as before. As it was already mentioned, this test consisted of obtained and ground truth translation vectors comparison. Because the resulted translation vectors from our reconstruction are normalized (by the nature of used algorithm) and we usually don't know the exact units of ground truth datasets (plus the translation vectors are mostly calculated to be relative to chosen reference point in this kind of ground truth datasets), the relative translation vectors (between two corresponding images) coming from dataset were normalized (set the norm of vector to be equal to one) for proper comparison. It means that every vector elements (*x*, *y* and *z*) in every comparison case were divided by Euclidean norm of this same vector.

For every tested pair there was 3D Euclidean distance (*ED*) between found and original translation vector calculated and these values were listed in Table 2. The arithmetical means *AM* (what we can consider as systematic part of error) and standard deviations *SD* (random part of error) from these values were also stated. As it is obvious from these results, the subpixel detection doesn't have any impact on accuracy of relative translation between two images determination. The values of both arithmetical means and standard deviations are almost identical and they are changing very slightly.

So the final conclusion based on these test is that subpixel corner detection doesn't have significant impact on accuracy of 3D position from image pair determination. The improvement of accuracy was more or less significant only in case of relatively small resolution images and the not relatively small displacement between them. This could be very often also the case of VSLAM methods. The reason why there were no improvements in comparison 2 might be the fact that in regular images (without chessboard segments) having highly different mutual position could be the right stating of corresponding points a little bit problematic or uncertain.

Table 2. Comparison of obtained and normalized ground truth translation vectors

tested pair	ED		
	pixel	subpixel A	subpixel B
1	0.1163	0.1266	0.1389
2	0.0792	0.0911	0.0853
3	0.1550	0.1339	0.1317
4	0.0582	0.0749	0.0741
5	0.0217	0.0315	0.0261
6	0.0786	0.0758	0.0736
7	0.0818	0.0769	0.0790
8	0.0647	0.0432	0.0453
9	0.0124	0.0146	0.0141
AM	0.0742	0.0743	0.0742
SD	0.0438	0.0402	0.0424

To finally prove or disprove the suitability of subpixel paradigm on 3D position determination or VSLAM methods is of course necessary to test this theory in more robust and thorough way.

Conclusion

This paper has dealt with impact of subpixel paradigm on determination of 3D position from 2D image pair. We have tried to answer the question, if the subpixel detection can increase the accuracy of 3D reconstruction. We have decided to use the Harris detector as traditional approach to detect the corner points and two subpixel corner detection methods, which were described in theoretical part of this paper.

We implemented the experiment, where the 3D positions of detected corner points using fundamental matrix and epipolar geometry theory were reconstructed. The details about this experiment are mentioned in experimental part of this contribution.

As the accuracy criterion were chosen the reprojection error and translation vectors comparison.

The arithmetical mean and standard deviation of this reprojected error based on data from all involved corners were stated for every tested image. Because the tested images were divided into specific groups (resolution, angle, method), the obtained results were averaged and listed in the table 1.

The table 2 contains the 3D Euclidean distances between obtained translation vectors and ground truth translation vectors coming from dataset. The arithmetic mean and standard deviation from these values for particular method were also calculated and stated in the table.

As far as it was found in this paper, the subpixel corner detection doesn't have significant impact on accuracy of 3D position determination under the circumstances we were discussed. The improvement of accuracy was more or less significant only in case of relatively small resolution images and the not relatively small displacement between them. This is very often exactly the case of VSLAM or visual servoing methods. The reason why there were no improvements in comparison 2 might be the problematic determination of corresponding corner points in case of chosen dataset images (no easy detected chessboard segments, highly different mutual position, different light conditions and so on).

The final statement subpixel detection effect on accuracy of 3D scene reconstruction requires the more robust and deep testing of course. It is one of our goals to implement the subpixel idea to one of VSLAM methods and evaluate the results in future.

Acknowledgement

The work presented in this paper was supported in part by Ministry of Education of the Slovak Republic under grant No. 1320 within the Framework of the program to Support Young Researchers: Teoreticka a prakticka studia vyuzitia SLAM metod.

This work was also supported by the Slovak Research and Development Agency under grant No. APVV-0469-12 and by the Slovak Grant Agency VEGA under grant No. 1/0963/12.

Special thanks to The Laboratory for Engineering Man/Machine Systems (LEMS) from Electrical Sciences faculty of the School of Engineering at Brown University, from where the tested dataset were obtained - http://vision.lems.brown.edu/ and also to Computer Vision Laboratory (CVLAB) from Ecole Polytechnique Federale de Lausanne (EPFL) for dataset http://cvlab.epfl.ch/data/strechamvs/ providing.

Bibliography

[Brown, 2005] M. Brown, R. Szeliski, and S. Winder, Multi-image matching using multi-scale oriented patches, International conference on Computer Vision and Pattern Recognition CVPR2005, 2005, 510-517.

[Harris, 1988] Ch. Harris, M. Stephens, A combined corner and edge detectors, In Alvey Vision Conference, 1988, 147-152.

[Hartley, 2003] R. Hartley, A. Zisserman, Multiple view geometry in computer vision, 2nd edition, Cambridge University Press, 2003, 239-259.

[Hartley, 2003] R. Hartley, A. Zisserman, Multiple view geometry in computer vision, 2ndedition, Cambridge University Press, 2003, 89-90.

[Rea, 2009] M. Rea, D. Mc. Robbea, D. Elhawary, Z. Tse, M. Lamperth, and I. Young, Sub-pixel localization of passive micro-coil fiducial markers in interventional MRI, MAGMA, 2009.

[Shi, 1994] J. Shi, C. Tomasi, Good features to track, IEEE Conference on Computer Vision and Pattern Recognition, 1994.

[Sroba, 2012] L. Sroba, R. Ravas, The subpixel X-corners detectors and their window shift robustness comparison, ELITECH '12, 14th Conference of Doctoral Students, ISBN 978-80-227-3705-0, 2012.

[Sroba, 2013] L. Sroba, R. Ravas, Impact of subpixel paradigm on homography determination, ELITECH '13, 15th Conference of Doctoral Students, ISBN 978-80-227-39470-4, 2013.

[Strecha, 2004] C. Strecha, R. Fransens, L. Van Gool, Wide-baseline stereo from multiple views: a probabilistic account, CVPR, 2004, 552-559..

[Strecha, 2006] C. Strecha, R. Fransens, L. Van Gool, Combined depth and outlier estimation in multi-view stereo, CVPR, 2006, 2394-2401.

[Sur, 2008] F. Sur, N. Noury, M. Berger, and I. Grandest, Computing the uncertainty 0f the 8 point algorithm for fundamental matrix estimation, In Proc. BMVC, 2008.

[Wang, 2000] W. Wang, H. Tsui, A SVD decomposition of essential matrix with eight solutions for the relative positions of two perspectives cameras, 15th International Conference on Pattern Recognition, 2000.

[Zhang, 2004] Z. Zhang, Camera calibration, Emerging topics in computer vision, Prentice Hall Professional Technical Reference, 2004, 4-43.

Authors' Information

Lukas Sroba *– PhD. student, Faculty of Electrical Engineering and Information Technology, Slovak University of Technology in Bratislava, Slovakia; e-mail: lukas.sroba@stuba.sk*

Major Fields of Scientific Research: Image processing, Measurement theory

Rudolf Ravas *– Associate Professor, Faculty of Electrical Engineering and Information Technology, Slovak University of Technology in Bratislava, Slovakia; e-mail: rudolf.ravas@stuba.sk*

Major Fields of Scientific Research: Image processing, Measurement theory

ANALYSIS OF THE PROPERTIES OF ORDINARY LEVY MOTION BASED ON THE ESTIMATION OF STABILITY INDEX

Lyudmyla Kirichenko, Vadim Shergin

***Abstract**: The work proposes a method for estimating the stability index of alpha-stable distributions by using moments of fractional order. Provided numerical modeling has fully justified all of the results. Comparative analysis of the efficiency among the proposed method of estimating the stability index and widely used methods was performed. Proposal method is much simpler, far faster and substantially less memory required.*

Estimation of generalized Hurst exponent from time series of the ordinary Lévy process was performed. Multifractal fluctuation analysis method and evaluation based on stability index estimation were compared. The results of numerical modeling showed that proposed method for estimating the fractal properties of the ordinary Lévy process, based on stability index estimation via fractional order moments is a much more accurate.

***Keywords**: alpha-stable variables, stability index estimation, fractional order moments, multifractal stochastic processes, Hurst exponent, generalized Hurst exponent, ordinary Levy motion.*

***ACM Classification Keywords**: G.3 Probability and statistics - Time series analysis, Stochastic processes, G.1 Numerical analysis, G.1.2 Approximation - Wavelets and fractals*

Introduction

Parameters estimation of the random variables is one of the major problems of mathematical statistics. Among a set of different distribution laws the special place is taken by alpha-stable distributions because just and only these laws may limit the distribution of sums of independent identically distributed random variables [Gnedenko, 1954]. Such distributions are widely used in models of stochastic processes describing a wide range of processes and phenomena (e.g., financial and stock market indices, river sinks, medical applications). High peaks, heavy tails and self-similarity are characteristics of such time series [Gnedenko, 1954, Zolotarev, 1986].

In general, case alpha-stable random variable is characterized by four parameters [Zolotarev, 1986], specifies the index of stability $0 < \alpha \le 2$, offset, scale and symmetry measures. Estimation of these parameters is a difficult task. This is partly caused by those facts that with few exceptions pdf's and cdf's of stable distributions are not expressed in terms of elementary functions.

Despite the variety of methods and algorithms have been developed for solving this problem, none of them provides a statistical efficiency of the resulting estimates (in the sense of reaching Cramer-Rao bound). Furthermore, many of the techniques have a high computational complexity or other drawbacks. Thus, developing new methods for estimating the parameters of alpha-stable distributions remains an actual problem.

Historically, the first groups of methods for estimating the parameters of stable distributions are ones based on order statistics, i.e. quantiles [Fama, 1971; McCulloch, 1986; Garcia, 2011]. These methods are characterized by low computational complexity however their performance (estimation accuracy) is also low, especially applying to indices of stability and symmetry measure estimation. Furthermore, such methods are very sensitive to sample truncation. Nevertheless, due to its simplicity, these methods are widely used both independently and as a parts of other, more complex methods for obtaining an initial approximations of estimates [Borak, 2010].

Another common class of methods for stability index estimating is based on the tails behavior studying [Hill, 1975; Dufour, 2010]. One of the fundamental property of stable distributions is asymptotically power law of the cdf: $P(X > x) \sim x^{-\alpha}$ as $X \to \infty$, $\alpha \neq 2$. The main disadvantage of this method is the bias of the resulting estimates. Furthermore, the effectiveness of such techniques depends essentially on the volume of the sample.

The maximum likelihood method just gives the most accurate estimates of the parameters of stable distributions [Nolan, 2001]. However, its computational complexity is very high, that is caused by both the properties of the method and the computational complexity of calculating the pdf's of stable distributions. That is why this class of methods is not common.

Just methods for estimating the parameters of stable distributions, based on the transition to the frequency domain, are the most common now [Koutrouvelis, 1980; Chenyao, 1999]. Not parameters p of pdf's $f(x;p)$ themselves are estimated by this methods but parameters of their characteristic functions $\varphi(t;p) = M(e^{itx}) = \int_{-\infty}^{\infty} e^{itx} f(x;p)dx$ are. This is because the stable distribution characteristic functions, in contrast to their pdf's, have a relatively simple mathematical form. These methods provide a sufficiently high accuracy of estimation, but are also quite complex computationally.

Stochastic process $X(t)$ is called self-similar, if the process $a^{-H}X(at)$ has the same finite-dimensional distributions, as the original process $X(t)$ has. Parameter H , called Hurst exponent (or Hurst index), is a measure of self-similarity of a stochastic process. There exist a lot of methods for Hurst index estimation based on a time series data, but most of them are designed for processes with finite second-order moments only, so they have substantial errors in estimation the Hurst index [Kirichenko, 2011]. Processes with independent increments which are identically distributed alpha-stable variables have self-similar properties which are completely determined by the stability index α [Cont, 2004]. Therefore, in this case, one should focus on the correct evaluation of stability index using the time series data.

Problem domain

Moment's method is classical method of point estimation. It is characterized by low computational complexity. However, its application is actually limited by class of distributions subordinated to normal, i.e. having stability index $\alpha = 2$. It's caused by the fact that random variable has no moments of order equal to or higher than α when $\alpha < 2$. In [Zolotarev, 1986] logarithmic moment's method for stable distributions parameters estimation was proposed. This method is simple to implement, but efficiency of estimates obtained by is lower than one obtained by methods based on using the transition to frequency domain.

However, the concept of the moment of a random variable can be generalized to noninteger $s \in \mathbb{R}$. It is known [Uchaikin, 2008] that for any α there exists an infinite set of s ensuring an existence of s-order moment. Thus, stable distributions parameters estimation via fractional moments is a new approach for solving the problem.

This paper is aimed to developing a method for estimating the stability index of α-stable distributions via fractional order moments and applying this method for studying the fractal properties of stochastic processes.

Estimation the stability index of alpha-stable laws via fractional moments method

The absolute moment order s of a random variable with the pdf $f(x)$, considered as function on s , is called as bilateral Mellin transform [Uchaikin, 2008]:

$$\left(\mathcal{M}f(x)\right)(s)=\int_{-\infty}^{\infty}|x|^{s}\,f(x)dx\,. \tag{1}$$

It's widely known [Zolotarev, 1986], that α-stable variable has a moments of order $-1<s<\alpha$. For strictly alpha-stable random variables Mellin transform has a closed form:

$$(\mathcal{M}g(x;\alpha,\rho,\lambda))(s)=\lambda^{s}\,\frac{\cos(\frac{\pi s}{2}(2\rho-1))\cdot\Gamma(1-s/\alpha)}{\cos(\frac{\pi s}{2})\cdot\Gamma(1-s)}, \tag{2}$$

where $\Gamma(x)$ is the gamma function.

In the case when random variable is symmetric ($\rho=1/2$) and has a unity scale factor ($\lambda=1$) expression (2) takes the form

$$(\mathcal{M}g(x;\alpha))(s)=\frac{\Gamma(1-s/\alpha)}{\cos(\frac{\pi s}{2})\cdot\Gamma(1-s)}=\frac{\Gamma(1-s/\alpha)}{\chi(s)}, \tag{3}$$

where $\chi(s)=\cos(\frac{\pi s}{2})\cdot\Gamma(1-s)\geq 1$.

Replacing in this relation the theoretical value of the moment $(\mathcal{M}g(x;\alpha))(s)$ with its sample value $Z_n(s)$, we can get an estimate of the stability index α:

$$\hat{\alpha}(n,s)=\frac{s}{1-\Gamma^{-1}(\chi(s)\cdot Z_n(s))}=\frac{s}{1-\Gamma^{-1}(1+Y_n(s))}, \tag{4}$$

where

$$Z_n(s)=\frac{1}{n}\sum_{k=1}^{n}|X_k|^{s}\,,\quad Y_n(s)=\chi(s)\cdot Z_n(s)-1\,. \tag{5}$$

Estimator (4), despite the simplicity of mathematical notation, has the obvious disadvantage of using function $\Gamma^{-1}(u)$, which is the inverse of the gamma function $u=\Gamma(x)$. This function not only applies to elementary, but not implemented in any of the known engineering and mathematical packages. Thus, one should numerically solve the nonlinear equation $\Gamma^{-1}(u)=\underset{x\in(0;1)}{sol}\left\{\Gamma(x)=u\right\}$, or (which is slightly simpler) optimization problem $\Gamma^{-1}(u)=\underset{x\in(0;1)}{\arg\min}\left\{\left(\Gamma(x)-u\right)^{2}\right\}$ for computing estimate (4) directly. This fact substantially reduces the usefulness of the proposed method and, on the other hand, makes it impossible to analyze the properties of esimate (4) by analytical methods.

As known from mathematical analysis, the function $\Gamma^{-1}(u)$, hence the function

$$x=f(y)=\frac{1}{1-\Gamma^{-1}(1+y)}, \tag{6}$$

are continuous and monotonically decreasing on the range $y\in(0;\infty)$, $x\in(1;\infty)$. Apparently, by approximating (6) with convergent series (on $1/y$) we can get the desired estimate of stability index in a much simpler form than (4), while ensuring any preassigned accuracy. In [Shergin, 2014] it was shown that the linear approximation

$$x^{(1)} \approx a+b/y\ ,\ (a = 1.19236\ ,\ b = 0.64072) \tag{7}$$

provides a relative error not exceeding 3.5%, which is sufficient for practical use. The second order approximation has the form

$$x^{(2)} \approx a+b/y+c/y^2\ ,\ (a = 1.11877,\ b=0.70107,\ c= -0.012374) \tag{8}$$

Given the expressions (7)-(8) we obtain the approximate estimates of stability index:

$$\hat{\alpha}(n,s) \approx s\left(a+\frac{b}{Y_n(s)}\right) = s\left(a+\frac{b}{\chi(s)\cdot Z_n(s)-1}\right) \tag{9}$$

$$\hat{\alpha}(n,s) \approx s\left(a+\frac{b}{Y_n(s)}+\frac{c}{Y_n^{\,2}(s)}\right) \tag{10}$$

In [Shergin, 2013] it was shown that on the range $s \in (-1;0)\cup(0;\alpha)$ estimator (4) is consistent and asymptotically unbiased (case $s = 0$ should be excluded as degenerate) and bias value of estimators (9)-(10) caused by an error of series expansion of function (6).

To obtain an asymptotic expression for variance of estimates $D\left[\hat{\alpha}(n,s)\right]$ it was used the fact that $Y_n(s)$ (5) is a cumulative average of independent identically distributed random variables with support $\mathrm{supp}(Y) = [0,\infty)$, hence asymptotic distribution of (5) (as $n \to \infty$) can be described by some infinitely divisible law. As such law the gamma distribution has been chosen. The resulting expression for variance of estimates has the form

$$D[\hat{\alpha}(n,s)] \approx \frac{b^2 D_0(\alpha,s)}{n}, \tag{11}$$

where

$$D_0(\alpha,s) = \frac{s^2\left(\dfrac{\chi^2(s)}{\chi(2s)}\Gamma(1-2s/\alpha)-\Gamma^2(1-s/\alpha)\right)}{\left(\Gamma(1-s/\alpha)-1\right)^4}. \tag{12}$$

From the expression (12) it follows that the variance of the estimates (4), (9)-(10) is finite on the range $s \in (-\frac{1}{2};\frac{\alpha}{2})$. Plot of the function (12) (from [Shergin, 2013]) is shown on Figure 1.

As one can see on Figure 1, for each value of stability index α there exist such values of fractional order s, equal to

$$s_{\min}(\alpha) = \underset{-1/2<s<\alpha/2}{\arg\min}\ \left(D_0(\alpha,s)\right), \tag{13}$$

which provide a local minimum value of the asymptotic variance of estimates. Thus, despite the fact that the estimates (4) are consistent and asymptotically unbiased for any values of s of range $s \in (-1;0)\cup(0;\alpha)$, estimation accuracy will be the higher as s the closer to $s_{\min}(\alpha)$.

Graphical representation of $s_{\min}(\alpha)$ is shown on Figure 2a, graphs of function (12) $D_{0,\min}(\alpha) = D_0(\alpha, s_{\min}(\alpha))$, corresponding to $s = s_{\min}(\alpha)$ are shown on Figure 2b.

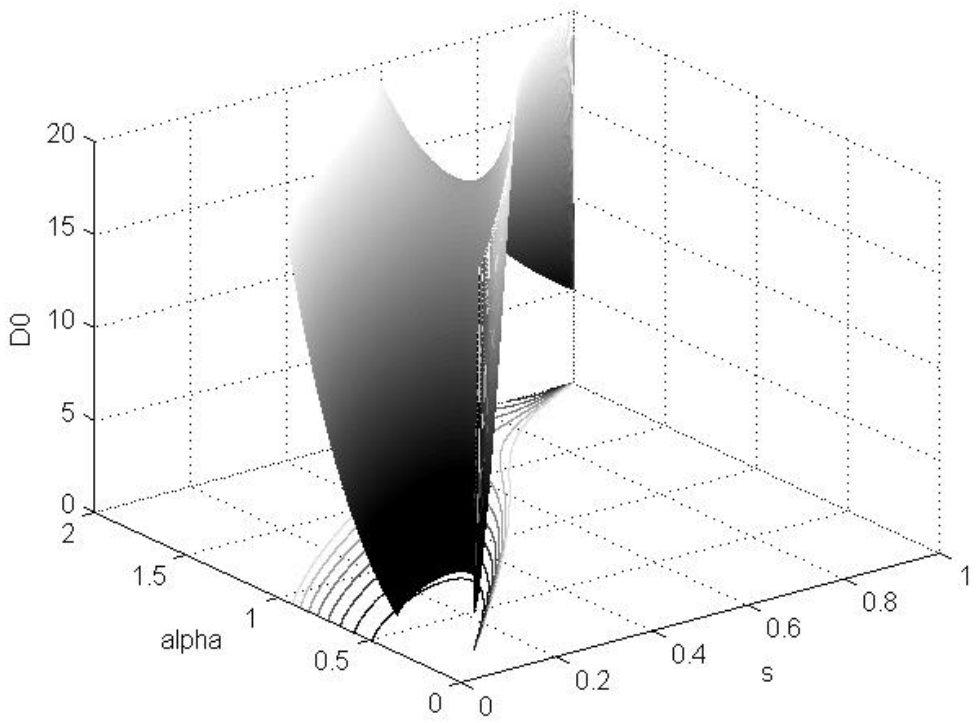
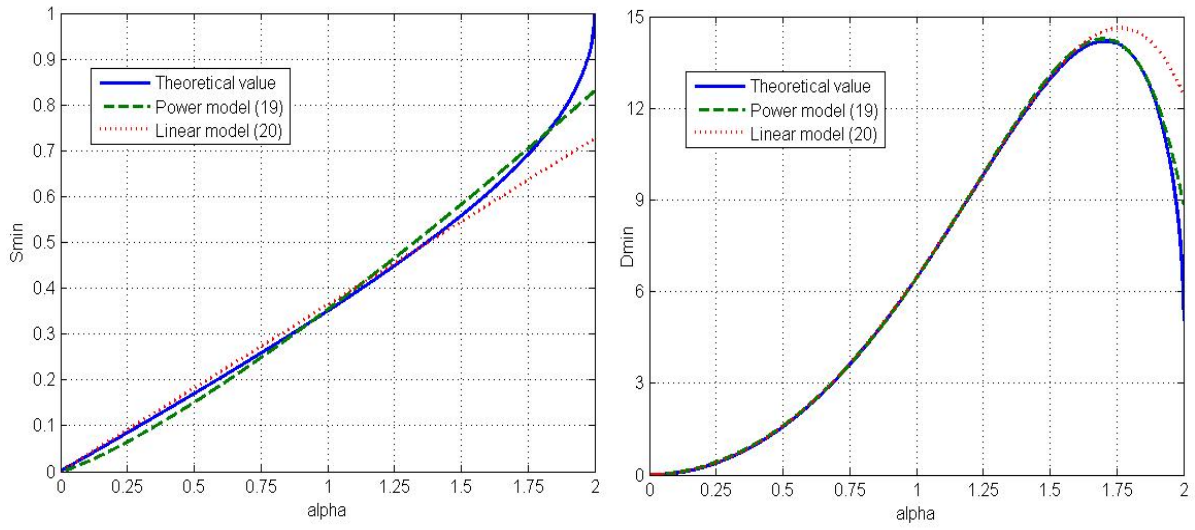


Figure 1. Asymptotic variance of estimates $D_0(\alpha, s)$ by model (12)



a b

Figure 2. Dependencies $s_{\min}(\alpha)$ (a), $D_{0,\min}(\alpha)$ (b), obtained by numerical minimization (13) on s, and by using models (14)-(15)

According to these graphs, function $D_{0,\min}(\alpha)$ peaks at $\alpha^* \approx 1.707$, which corresponds to the value $s_{\min} \approx 0.665$. Thus, the neighborhood of α^* are the least favorable values of stability index to their evaluation by the proposed method.

Dependence (13) can be approximated by

$$s_{\min}^{pow} = 0.35281 \cdot \alpha^{1.2332}, \tag{14}$$

$$s_{\min}^{linear} = 0.3630 \cdot \alpha. \tag{15}$$

From (15) it follows that neighborhood of $s / \alpha \approx 0.3630$ (which corresponds to y_0=0.41007, $x_0 = 2.75482$) is the most reasonable point for Taylor series expansion of (6). That is cause numerical values of coefficients of models (7) – (8) obtained in [Shergin, 2014].

Stability index estimation algorithm

The analysis found that for a given value of the fractional moment order s the stability index α estimators has the form (9) – (10) (with notations (3), (5) and coefficients (7) – (8)). Variance of this estimates will be the least while $s = s_{\min}^{pow}(\alpha)$ (14). Thus, a simple iterative procedure is proposed

$$s^{(m+1)} = s_{\min}^{pow}\left(\hat{\alpha}(n, s^{(m)})\right), \tag{16}$$

where functions $s_{\min}^{pow}(\hat{\alpha})$ and $\hat{\alpha}(n, s^{(m)})$ are calculated according to (14) and (9) or (10) respectively.

An exit condition for the loop (16) has the form $\left|s^{(m+1)} - s^{(m)}\right| \le tol$. Values, equal to 0.25 and 10^{-4} were used as $s^{(0)}$ and tol respectively.

It was performed comparative analysis of the efficiency among the proposed method of estimating the stability index and well known methods based on quantile parameter estimates and regression parameter estimates STABCULL and STABREG [MFE Toolbox for MATLAB]. Plot of the error variance of the stability index estimates on the sample length is shown on Figure 3.

As one can see, error variance provided by all of these methods is about the same. However proposal method is much simpler, about 12 times faster than STABREG and substantially less memory required.

Basic definitions and characteristics of fractal stochastic processes

Stochastic processes that exhibit fractal properties can be divided into two groups: self-similar (monofractal) and multifractal. Monofractal processes are homogeneous in the sense that their scaling characteristics remain constant at any range scale. Monofractal processes have the single scaling exponent. Multifractal processes can be expanded to ranges with different local scaling properties. Multifractal processes have the spectrum of scaling exponents. Consider the basic concepts of self-similar and multifractal random processes [Feder, 1991; Calvet, 1997; Reidi, 2002; Kantelhardt, 2008].

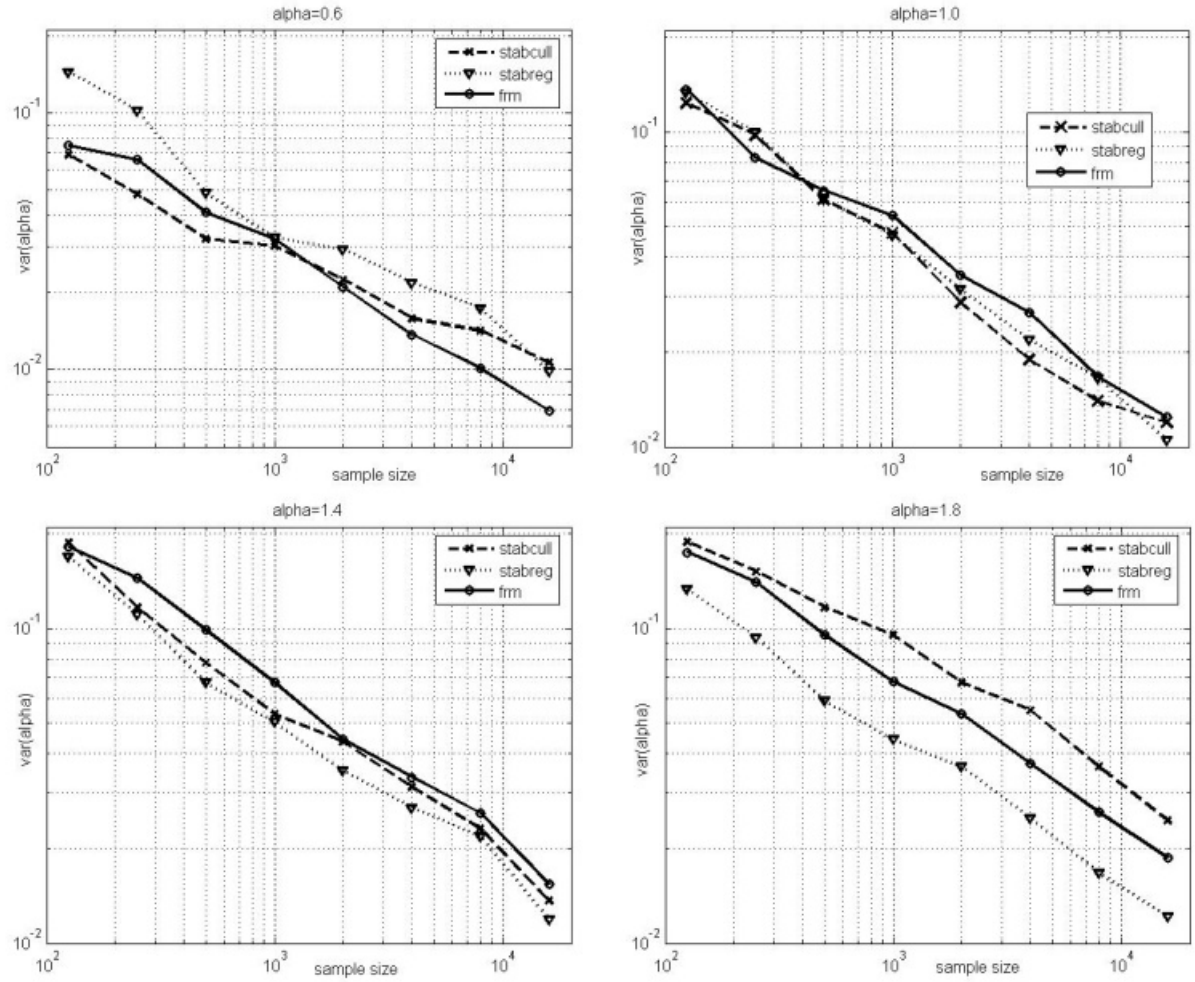


Figure 3. Error variance of the stability index estimates on the sample length via proposal method (Frm) and widely used Stabreg and Stabcull

Stochastic process $X(t)$, $t \geq 0$ with continuous real-time variable is said to be self-similar of index H, $0 < H < 1$, if for any value $a > 0$ processes $X(at)$ and $a^{-H}X(at)$ have same finite-dimensional distributions:

$$\text{Low}\{X(t)\} = \text{Low}\{a^{-H}X(at)\}. \tag{17}$$

The notation $\text{Low}\{\cdot\}$ means finite distribution laws of the random process. Index H is called Hurst exponent. H is a measure of self-similarity of a stochastic process. Ordinary moments of self-similar process can be can be expressed by

$$\mathcal{M}\left[|X(t)|^q\right] = \mathcal{M}\left[|t^H X(1)|^q\right] = t^{qH}\mathcal{M}\left[|X(1)|^q\right] = C(q)\cdot t^{qH}, \tag{18}$$

where value $C(q) = \mathcal{M}\left[|X(1)|^q\right]$.

In contrast to the self-similar processes (17) multifractal processes have more varied scaling behavior:

$$\text{Law}\{X(at)\} = \text{Law}\{\mathcal{M}(a)\cdot X(t)\},\ a>0, \tag{19}$$

where $\mathcal{M}(a)$ is random function that independent of $X(t)$.

In case of self-similar process $\mathcal{M}(a) = a^H$. Hurst exponent of multifractal processes is a random function of the argument a: $H(a) = \log_a \mathcal{M}(a)$. Relation (19) can be reformulated as follows:

$$\text{Law}\{X(at)\} = \text{Law}\{a^{H(a)}\cdot X(t)\}. \tag{20}$$

Defining characteristic of multifractal processes: process $X(t)$ is multifractal, if the following relation holds:

$$\mathcal{M}\left[|X(t)|^q\right] = c(q)\cdot t^{qh(q)}, \tag{21}$$

where $c(q)$ is some deterministic function, $h(q)$ is generalized Hurst exponent, which is generally non-linear function. Value $h(q)$ at $q=2$ is the same degree of self-similarity H. For monofractal processes generalized Hurst exponent does not depend on the parameter q: $h(q) = H$.

There are many methods for estimating the parameters of self-similar and multifractal processes from time series. [Clegg, 2005; Kantelhardt, 2008].When estimating the Hurst exponent in practice most commonly used methods are R/S -analysis, variance-time analysis and detrended fluctuation analysis (DFA). When estimating multifractal characteristics one of the most popular methods is multifractal detrended fluctuation analysis (MFDFA) [Kantelhardt, 2002].

According to the MFDFA method, for the initial time series $x(t)$ the cumulative time series $y(t) = \sum_{i=1}^{t} x(i)$ is constructed which is then divided into N segments of length τ, and for each segment $y(t)$ the following fluctuation function is calculated:

$$F^2(\tau) = \frac{1}{\tau}\sum_{t=1}^{\tau}(y(t) - Y_m(t))^2, \tag{22}$$

where $Y_m(t)$ is a local m-polynomial trend within the given segment. The averaged on the whole of the time series $y(t)$ function $F(\tau)$ depends on the length of the segment: $F(\tau)\propto\tau^H$.

In the study of multifractal properties the dependence of the fluctuation function $F_q(s)$ of a parameter q is considered: $F_q(s) = \left\{\frac{1}{N}\sum_{i=1}^{N}[F^2(s)]^{\frac{q}{2}}\right\}^{\frac{1}{q}}$. If the investigated series is multifractal and has a long-term dependence, the fluctuation function is represented by a power law

$$F_q(s)\propto s^{h(q)}, \tag{23}$$

where $h(q)$ is generalized Hurst exponent. For monofractal time series the fluctuation function $F_q(s)$ is the same for all segments, and the generalized Hurst exponent does not depend on the parameter q: $h(q) = H$. For multifractal series $h(q)$ s a nonlinear function.

Basic definitions and characteristics of ordinary Levy motion

Consider the basic concepts of self-similar and multifractal random processes [Cont, 2004]. A stochastic process $X(t),\ t \geq 0$ with real values is called Levy process, if it possesses the following properties:

- Process is right-continuity and left limits;
- Process starts at zero ($X_0 = 0$);
- At every time interval $t_0,...,t_n$ increments $X(t_0), X(t_1) - X(t_0),..., X(t_n) - X(t_{n-1})$ are independent random variables;
- Increments are stationary;
- Stochastic continuity is performed : $\forall \varepsilon > 0 \ \lim\limits_{h \to 0} P\left(\left|X(t+h) - X(t)\right| \geq \varepsilon\right) = 0.$

A stochastic process $X(t),\ t \geq 0$ with real values is called α – stable Levy process (ordinary Levy motion) , if it possesses the following properties:

- $X(t)$ is Levy process;
- For every $a > 0$, $t \geq 0$ the following relationship holds:

$$\text{Law}\{X(at)\} = \text{Law}\{a^{1/\alpha} X(t)\}. \tag{24}$$

Comparing the expressions (17) and (24), it is clear, that α – stable Levy processes have the property of self-similarity. Shown [Nakao, 2000; Oswiecimka, 2006] that such processes are multifractal. In this case the function of generalized Hurst exponent $h(q)$ takes the form:

$$h(q) = \begin{cases} 1/\alpha & q \leq \alpha, \\ 1/q & q > \alpha; \end{cases} \tag{25}$$

where α is stability index.

Obviously, the function of generalized Hurst exponent of ordinary Levy motion is completely determined by the stability index α.

Investigation Results

In this work the results of a numerical experiment are represented where realizations of ordinary Levy motion have been simulated. The length of realizations was accepted equal to 250, 500, 1000 and 2000. For every received realization, generalized Hurst exponent have been obtained using two methods: directly by MFDFA and on based of the estimating stability index by formulas (8). The obtained estimation values of generalized Hurst exponent then were averaged over a set of realizations. Value of the parameter q changed in the range $-5 \leq q \leq 5$.

Figure 4 (above) shows typical realization of ordinary Levy motion. Its increments (below) are independent stable random variables X_i with stability index $\alpha = 1.2$, ie $X_i \sim S_{1.2}(1,0,0)$. Figure 5 shows the estimates of $h(q)$, that obtained by the realizations of varying lengths by MFDFA method. The realizations of such a process are self-similar of Hurst exponent $H = \frac{1}{\alpha} = \frac{5}{6}$. Dashed line on the graph shows the theoretical values of the function $h(q)$.

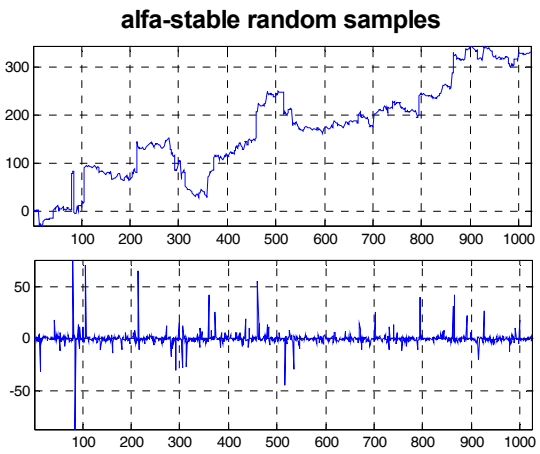


Figure 4. Levy motion realization (above), its increments (below)

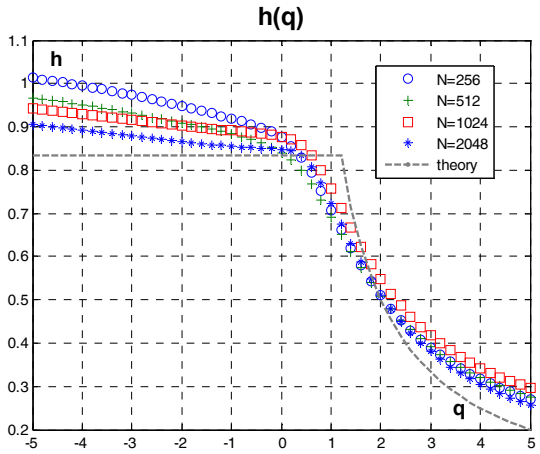


Figure 5. Function $h(q)$ of Levy motion realizations

Figure 6 shows the estimation results of generalized Hurst exponent $h(q)$ by MFDFA method (left) and on based of the estimating stability index using the method of fractional moments (right). The length of realizations in these cases was equal to 5000 values. Continuous line on the graphs shows the theoretical values of the function $h(q)$

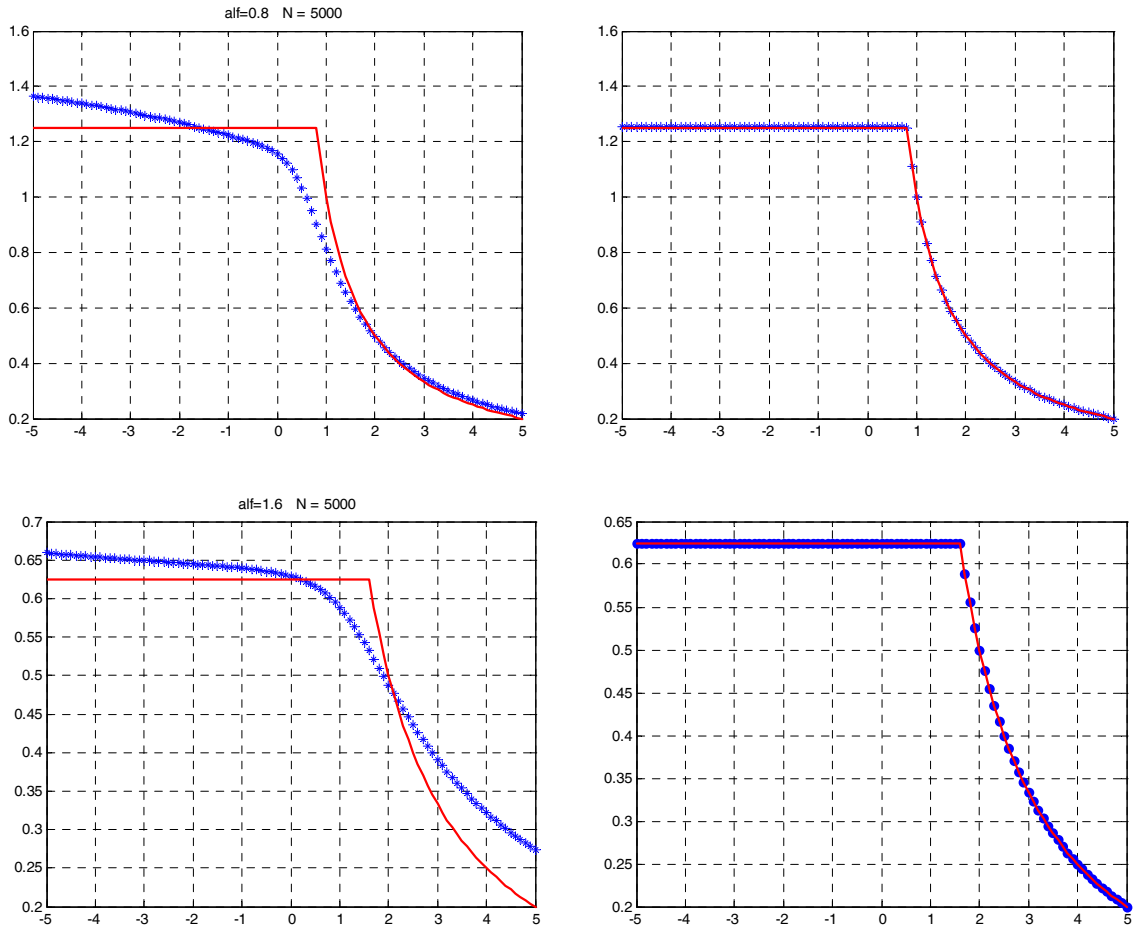


Figure 6. Estimates of $h(q)$ for Levy motion realizations of α=0.8 (above) and α=1.8 (below) by MFDFA (left) on based of the estimating α (right).

Obviously, the evaluation function of the generalized Hurst exponent on based of the estimating stability index provides much more accurate results. However, the use of such an approach is possible only under condition of acceptance of the hypothesis that the process under study is ordinary Levy motion.

Conclusion

The problem of estimating the stability index (alpha) of $S\alpha S$-distributions via fractional order moments was considered. The required estimate was obtained. The consistency and asymptotic unbiasedness of this estimate were proved, and their asymptotic variance was estimated. As it was found, for any admissible $0 < \alpha \le 2$ there exists a fractional order moment value $s_{\min}(\alpha)$, which minimize the asymptotic variance of estimates of α. Dependence $s_{\min}(\alpha)$ was obtained and approximated in a polynomial form.

The provided numerical modeling has fully justified all of the results. It was performed comparative analysis of the efficiency among the proposed method of estimating the stability index and wide known methods based on quantile parameter estimates and regression parameter estimates. Proposal method is much simpler, far faster and substantially less memory required.

Estimation of generalized Hurst exponent from time series of the ordinary Lévy process was performed. Multifractal fluctuation analysis method and evaluation based on stability index estimation were compared. The results of numerical modeling showed that proposed method for estimating the fractal properties of the ordinary Lévy process, based on stability index estimation via fractal order moments is a much more accurate.

Bibliography

[Borak, 2010] S. Borak, A. Misiorek, R.Weron. Models for heavy-tailed asset returns. SFB649DP2010-049, Sonderforschungsbereich 649, Humboldt University, Berlin, Germany, 40.

[Calvet, 1997] L. Calvet. Large Deviationsand the Distribution of Price Changes / L. Calvet, A. Fisher, B.B. Mandelbrot // Cowles Foundation Discussion Paper – 1997. –N.1165. –P. 1-30.

[Chenyao, 1999] D.Chenyao, S.Mittnik, T.Doganoglu. Computing the probability density function of the stable paretian distribution, Mathematical and Computer Modelling, 29, 235-240.

[Clegg, 2005] R.G. Clegg. A practical guide to measuring the Hurst parameter. R. G. Clegg. Computing science technical report (№ CS–TR–916), 2005.

[Cont, 2004] R. Cont, P. Tankov P. Financial modelling with jump processes, Chapman & Hall/CRC Press.2004, 527 p.

[Dufour, 2010] J-M.Dufour, J-R.Kurz-Kim. Exact inference and optimal invariant estimation for the tail coefficient of symmetric alpha-stable distributions. Journal of Empirical Finance, Vol.17(2), 180-194.

[Fama, 1971] E.Fama, R.Roll. Parameter estimates for symmetric stable distributions. Journal of the American Statistical Association, 66, 331-338.

[Feder, 1988] J. Feder. Fractals. J. Feder. Plenum, New York, 1988.

[Garcia, 2011] R.Garcia, E.Renault, D.Veredas. Estimation of stable distributions with indirect inference. Journal of Econometrics, 161, 325-337.

[Gnedenko, 1954] B.V.Gnedenko, A.N.Kolmogorov. Limit distributions for sums of independent random variables. Addison-Wesley.

[Hill, 1975] B.M.Hill. A simple general approach to inference about the tail of a distribution, Annals of Statistics, 3, 1163-1174.

[Kantelhardt, 2002] J.W. Kantelhardt, S.A. Zschiegner, A. Bunde, S. Havlin, E. Koscielny-Bunde, H.E. Stanley Multifractal detrended fluctuation analysis of non-stationary time series . Physica A. – 2002. – № 316. – P. 87–114.

[Kantelhardt, 2008] J.W. Kantelhardt. Fractal and Multifractal Time Series. – 2008 [Электронный ресурс]: http://arxiv.org/abs/0804.0747

[Kirichenko, 2011] L. Kirichenko. Comparative analysis of statistical properties of the Hurst exponent estimates obtained by different methods L. Kirichenko, T.Radivilova. Information Models of Knowledg / ed. K. Markov, V. Velychko, O. Voloshin. – Kiev–Sofia: ITHEA. – 2010. – P. 451–459.

[Koutrouvelis, 1980] I.A.Koutrouvelis. Regression-type estimation of the parameters of stable laws, Journal of the American Statistical Association, 75, 918-928.

[McCulloch, 1986] J.H.McCulloch. Simple consistent estimators of stable distribution parameters. Communications in Statistics, Computation and Simulation, 15, 1109– 1136.

[MFE Toolbox for MATLAB] Retrieved from http://ideas.repec.org/c/wuu/hscode/zip00001.html

[Nakao, 2000] H. Nakao Multi-scaling properties of truncated Levy flights. Phys. Lett. A. – 2000. –V.266 –P. 282-289

[Nolan, 2001] J. P.Nolan. Maximum likelihood estimation of stable parameters. In O. E. Barndorff-Nielsen, T. Mikosch, and S. I. Resnick (Eds.), Levy Processes: Theory and Applications, Boston: Birkhauser, 379-400.

[Nolan, 2009] J. P.Nolan. Stable distributions models for heavy tailed data. Boston: Birkhauser Unfinished manuscript, Chapter 1. Retrieved from http: //academic2.american.edu/~jpnolan/stable/chap1.pdf.

[Oswiecimka, 2006] P. Oswiecimka, J. Kwapin, S.Drozdz. Wavelet versus detrended fluctuation analysis of multifractal structures / // Physical Review E: Statistical, Nonlinear, and Soft Matter Physics. - Vol. 74. -2006. -P. 161-203.

[Reidi, 2002] R.H.Riedi. Multifractal processes, in Doukhan P., Oppenheim G., Taqqu M.S. (Eds.), Long Range Dependence: Theory and Applications: Birkhuser. -2002. -P. 625–715,

[Shergin, 2013] V.L.Shergin. Estimation of the stability factor of alpha-stable laws using fractional moments method, Eastern-European Journal of Enterprise Technologies, 6, 25-30.

[Shergin, 2014] V.L.Shergin. Approximation an estimate of the SαS-distributions stability factor, Eastern-European Journal of Enterprise Technologies, 1, 34-38.

[Uchaikin, 2008] V. V. Uchaikin. Fractional derivatives method. Ulyanovsk, Russia: Artishok, 512.

[Zolotarev, 1986] V.M.Zolotarev. One-dimensional stable distributions. American Mathematical Society.

Authors' Information

Lyudmyla Kirichenko *– Doctor of Technical Sciences, Professor, Kharkiv National University of Radioelectronics; 14 Lenin Ave., 61166 Kharkiv, Ukraine; e-mail: ludmila.kirichenko@gmail.com.*

Major Fields of Scientific Research: Time series analysis, Stochastic self-similar and multifractal processes, Wavelets , Chaotic systems

Vadim Shergin *– Ph. D., Associate professor, Kharkiv National University of Radioelectronics; 14 Lenin Ave., 61166 Kharkiv, Ukraine; e-mail: sherginvl@mail.ru.*

Major Fields of Scientific Research: Time series analysis, Stochastic self-similar processes, Mathematical statistics, Chaotic systems

BUILDING PRECONDITIONERS USING BASIS MATRIX METHOD

Volodymyr Kudin, Vsevolod Bohaienko

***Abstract**: New class of preconditioners for iterative algorithms of sparse linear systems solution built using basis matrix method and incomplete decomposition methodology has been proposed. Algorithms with static and dynamic restriction set along with additional refinement procedures have been presented. Results of developed algorithms' testing carried out on matrices from Tim Davis Matrix Collection have been given. Basing on received results, matrix classes, applying proposed preconditioners on which resulted in iterative algorithms speed-up and/or accuracy increase, have been identified*

***Keywords**: linear systems, sparse matrices, poor conditioned matrices, preconditioners, basis matrix method.*

***ACM Classification Keywords**: H.4.2 Information Systems Applications: Types of Systems: Decision Support.*

Introduction

Matrix computations, especially linear systems solution problems, appear while doing mathematical modelling of most physical processes. In many cases, such linear systems have ill-conditioned large sparse matrices and are usually solved by iterative methods such as conjugate gradients (CG) or stabilized biconjugate gradients (BiCGstab) [Saad, 2003].

Preconditioning [Ke Chen, 2005] – methods based on multiplying linear systems' matrix by another matrix that results in condition number decrease, is a main technique used to achieve better convergence rate and/or accuracy of iterative methods. While inverse matrix is an ideal preconditioner, most preconditioners are built as its approximation. Most common methods are incomplete decompositions (e.g. incomplete LU decomposition) and incomplete inversions (e.g. polynomial preconditioners), among others multigrid and wavelet preconditioners can be distinguished.

It's worth noting that efficiency of preconditioners usage in most cases can't be theoretically proved, so there is a problem of experimental finding of matrix classes for which given preconditioner is efficient in sense of computations speed-up or solution accuracy improvement.

Preconditioners built upon basis matrix method

Incomplete decomposition is one of preconditioner building methods which consists in a restriction of a set of matrix elements changing in decomposition process, e.g. LU [Saad, 2003], QR [Ke Chen, 2005], or while applying inversion procedure, e.g. Gram-Schmidt method [Qiaohua Liu, 2013]. This yields to a matrix, which approximates inverse matrix and can be used as a preconditioner.

Such method can also be applied to algorithms of basis matrix method [Kudin, 2007]. Consider linear system $Ax = b$ where $\dim A = [n,n]$, $\dim x = \dim b = [n,1]$. Basis matrix methods' algorithm which results in matrix A^{-1}, inverse of A, may be stated as follows: Let $A_{\sigma}^{(i)}$, $\dim A_{\sigma}^{(i)} = [n,n]$ be a basis matrix on i-th iteration of algorithm, $A_{\sigma}^{(i)}[j]$ be a column of matrix $A_{\sigma}^{(i)}$, a_i be a row of matrix A, $A_{\sigma}^{(0)} = A_{\sigma}^{0}$. When i-th iteration of algorithm can be written as follows:1) $v = diag\left(AA_{\sigma}^{(i-1)}\right)$; 2) $k = \max_i |v_i|$; 3) $\alpha = a_k A_{\sigma}^{(i-1)}$;

4) $A_{\delta}^{(i)}[k] = A_{\delta}^{(i-1)}[k] / \alpha_k$; 5) $A_{\delta}^{(i)}[l] = A_{\delta}^{(i-1)}[l] - \alpha_l A_{\delta}^{(i)}[k],\ l \neq k$;

After execution of n iterations, algorithm results in $A_{\delta}^{(n)} = A^{-1}$.

Algorithm of incomplete basis matrix method can be obtained from aforementioned algorithm by changing step 5 as follows: 5a) $A_{\delta}^{(i)}[l]_j = A_{\delta}^{(i-1)}[l]_j - \alpha_l A_{\delta}^{(i)}[k]_j,\ l \neq k,\ (l,j) \in R$, where R is a set of matrix elements' indices. Here, after execution of n iterations, algorithm results in $A_{\delta}^{(n)} \approx A^{-1}$.

Definition of set R is needed to obtain a particular algorithm. By analogy with incomplete LU decomposition algorithms [Saad, 1994], consider following incomplete basis matrix methods' algorithms: IBMM0, where R is a set of matrix A non-zero elements, and IBMM1, where R is a set of matrix AA^T non-zero elements.

Another variant of incomplete basis matrix methods' algorithm (also by analogy with other incomplete decomposition algorithms) is an algorithm with dynamically changing R. Let R be restricted in such way that (condition1) number of (l,j) elements in it can't be bigger that number of non-zero elements in l-th row of matrix A. This can be taken into account by changing step 5 as follows:

5б) If $(l,j) \in R$ or $(l,j) \notin R$, but condition 1 is met only for column l, then do following transformation: $A_{\delta}^{(i)}[l]_j = A_{\delta}^{(i-1)}[l]_j - \alpha_l A_{\delta}^{(i)}[k]_j,\ l \neq k,\ R = R \cup (l,j)$. Here element will be changed and added to the set if set does not contain it;

If $(l,j) \notin R$ and condition 1 is not met, then transformation $A_{\delta}^{(i-1)}[l]_m = 0, A_{\delta}^{(i)}[l]_j = A_{\delta}^{(i-1)}[l]_j - \alpha_l A_{\delta}^{(i)}[k]_j,\ R = \left(R - (l,m)\right) \cup (l,j)$ must be done if $\exists m : \left|A_{\delta}^{(i-1)}[l]_m\right| < \left|A_{\delta}^{(i-1)}[l]_j - \alpha_l A_{\delta}^{(i)}[k]_j\right|$: element will be changed and added to the set if column contains an element with lower absolute value. That element at the same time will be set to zero and removed from set R. We'll call such algorithm IBMMd.

Consider additional procedures which extend IBMM0, IBMM1 and IBMMd algorithms.

Condition $a_k A_{\delta}^{(i)}[l] = \begin{cases} 1, & l = k \\ 0, & l \neq k \end{cases}$ is met on every iteration of basis matrix method, but not during incomplete transformation. To make this condition met in IBMM0 and IBMM1 algorithms' iterations, following step must be added:

6) If $(l,j) \notin R$, $a_{kj} \neq 0$ and $\exists m = \max_j \left|a_{kj}\right|, A_{\delta}^{(i-1)}[l]_j \neq 0$ then transformation $A_{\delta}^{(i)}[l]_m = A_{\delta}^{(i)}[l]_m - \alpha_l A_{\delta}^{(i)}[k]_j a_{kj} / a_{km}$ must be done.

In the case of IBMMd algorithm, step 6 must be applied when $(l,j) \notin R$, condition 1 is not met and an element with lower absolute value does not exist in a column, or while substituting a non-zero element:

6а) If $(l,j) \notin R$, condition 1 is not met, $\neg\exists m : \left|A_{\delta}^{(i-1)}[l]_m\right| < \left|A_{\delta}^{(i-1)}[l]_j - \alpha_l A_{\delta}^{(i)}[k]_j\right|$ and $\exists m = \max_j \left|a_{kj}\right|, A_{\delta}^{(i-1)}[l]_j \neq 0$, then do following transformation: $A_{\delta}^{(i)}[l]_m = A_{\delta}^{(i)}[l]_m - \alpha_l A_{\delta}^{(i)}[k]_j a_{kj} / a_{km}$.

6б) If $(l,j) \notin R$, condition 1 is not met, $\exists m : \left|A_{\delta}^{(i-1)}[l]_m\right| < \left|A_{\delta}^{(i-1)}[l]_j - \alpha_l A_{\delta}^{(i)}[k]_j\right|$ and $\exists p = \max_j \left|a_{kj}\right|, A_{\delta}^{(i-1)}[l]_j \neq 0,\ j \neq m$, then do following transformation $A_{\delta}^{(i)}[l]_p = A_{\delta}^{(i)}[l]_p - \alpha_l A_{\delta}^{(i)}[k]_m a_{km} / a_{kp}$.

We'll designate algorithms with correction step 6 as IBMM0+c, IBMM1+c, IBMMd+c.

Taking into consideration efficiency of Jacobi preconditioner while solving many linear systems, it can be combined with incomplete basis matrix methods' algorithms. In such case, left preconditioner will take the following form: $\left[diag A_{\acute{o}}^{(n)} A \right]^{-1} A_{\acute{o}}^{(n)}$. We'll designate such algorithms as "+rs".

Refinement procedure [Bohaienko, 2009] consisting in iterative algorithm execution with $A_{\acute{o}}^{(0)} = A_{\acute{o}}^{(n)}$, can be also applied to incomplete basis matrix methods' algorithms.

Testing of preconditioners efficiency

Efficiency of matrices, generated by developed algorithms, as left preconditioners has been tested on matrices from *Tim Davis Matrix Collection* (http://www.cise.ufl.edu/research/sparse/matrices/index.html) while solving corresponding linear problems using Bicgstab algorithm. Number of iterations after which given accuracy was achieved is given in Table 1. Maximal iteration number was set to 1000. All elements of vector b was equal to 1, and $A_{\acute{o}}^{0} = I$. Here information about matrix is given to the right of its' name, nnz(A) is a number of non-zero elements and cond(A) is a condition number.

Table 1. Number of iterations after which accuracy ε was achieved

			1	1	1	1	1	1	1	2	2	2	2	2	2	Number of refinement iteration
			-	-	-	-	-	-	+	-	-	-	-	+	+	Correction steps
	-	+	-	-	-	+	+	+	-	-	-	+	+	-	+	Jacobi preconditioner
Matrix and $\log_{10} \varepsilon$	-	-	0	1	d	0	1	d	0	0	1	0	1	1	1	Algorithm: 0 – IBMM0 1 – IBMM1 d - IBMMd
bfwa62	Collection *Bai*, dimA=[62,62], nnz(A)=450, cond(A)=553, electro-dynamical problem															
-2	26	20	57	-	-	-	-	-	-	-	-	-	-	-	-	
-4	35	23	61	-	-	-	-	-	-	-	-	-	-	-	-	
-6	37	31	-	-	-	-	-	-	-	-	-	-	-	-	-	
-8	42	33	-	-	-	-	-	-	-	-	-	-	-	-	-	
-10	44	34	-	-	-	-	-	-	-	-	-	-	-	-	-	
bfwa398	Collection *Bai*, dimA=[398,398], nnz(A)=3678, cond(A)=2993, electro-dynamical problem															
-2	72	54	269	-	226	313	-	195	-	-	-	-	-	-	-	
-4	103	78	295	-	230	317	-	213	-	-	-	-	-	-	-	
-6	107	87	304	-	266	317	-	245	-	-	-	-	-	-	-	
-8	108	89	-	-	286	-	-	281	-	-	-	-	-	-	-	
-10	118	90	-	-	-	-	-	-	-	-	-	-	-	-	-	
olm100	Collection *Bai*, dimA=[100,100], nnz(A)=396, cond(A)=15275, hydrodynamic problem															
-2	82	-	73	-	-	-	-	-	-	56	-	-	-	35	58	
-4	83	-	79	-	-	-	-	-	-	56	-	-	-	35	61	
-6	-	-	-	-	-	-	-	-	-	-	-	-	-	35	61	
poli	Collection *Grund*, dimA=[4008,4008], nnz(A)=8188, cond(A)=311, economical problem															
-6	17	17	8	6	-	8	7	-	-	8	8	8	10	-	-	
-10	20	20	9	9	-	9	9	-	-	9	9	9	12	-	-	
-14	-	-	-	-	-	-	-	-	-	-	-	9	12	-	-	

A more detailed look has been taken on developed algorithms' efficiency when applying them to matrices from *Averous* collection: *epb0*, *epb1*, *epb2*, *epb3*. Characteristics of these matrices which arise from thermodynamical problems are given in Table 2.

Table 2. Characteristics of *Averous* collection matrices

Matrix	Number of rows	nnz(A)	cond(A)	Sparsity pattern
epb0	1794	7764	64165	
epb1	14734	95053	5940	
epb2	25228	175027	2618	
epb3	84617	463625	-	

Data, same as in Table 1, are given in Table 3.

Table 3. Number of iterations after which accuracy ε was achieved for *Averous* collection matrices

			1	1	1	1	1	1	2	2	2	Number of refinement iteration
			-	-	-	-	-	-	-	-	-	Correction steps
	-	+	-	-	-	+	+	+	-	-	+	Jacobi preconditioner
Matrix and $\log_{10}\varepsilon$	-	-	0	1	d	0	1	d	0	1	0	Algorithm: 0 – IBMM0 1 – IBMM1 d - IBMMd
epb0												
-4	-	918	-	-	-	663	-	-	490	-	490	
-6	-	992	-	-	-	675	-	-	526	-	518	
-8	-	-	-	-	-	772	-	-	548	-	528	
-10	-	-	-	-	-	815	-	-	585	-	567	
-12	-	-	-	-	-	838	-	-	588	-	575	
epb1												
-4	358	321	231	210	125	187	199	133	255	272	279	
-6	426	407	274	260	160	233	228	166	383	385	399	
-8	517	488	343	303	198	295	262	201	434	505	423	

-10	541	517	363	348	203	299	325	212	466	518	430	
-12	573	552	405	352	211	336	331	221	490	664	488	
epb2												
-4	192	111	91	96	-	85	95	-	96	-	111	
-6	259	128	116	118	-	111	119	-	139	-	148	
-8	315	166	133	132	-	133	130	-	181	-	176	
-10	461	191	162	159	-	157	175	-	198	-	231	
epb3												
7	120	99	105	-	-	-	-	-	103	-	-	
5	291	258	428	-	-	-	-	-	542	-	-	
3	845	297	511	-	-	-	-	-	566	-	-	
1	-	616	569	-	-	-	-	-	661	-	-	

Change of residual logarithm while solving linear system with *epb0* matrix applying different preconditioners is presented on Figure 1.

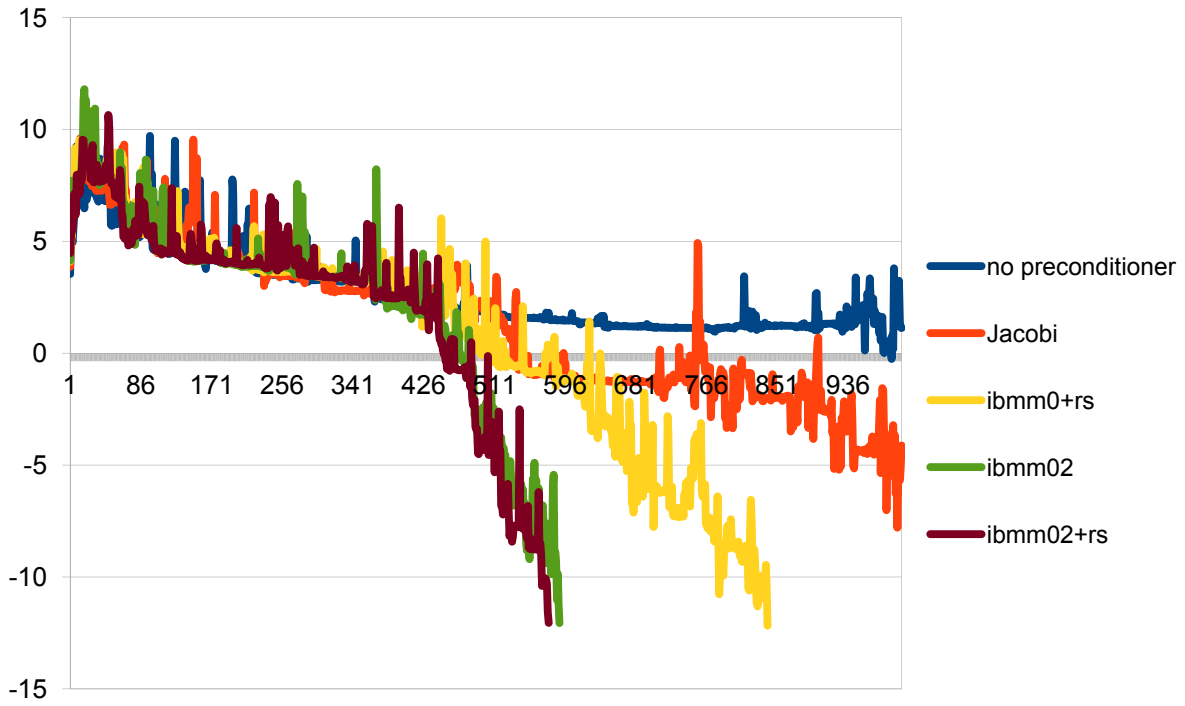


Figure1. $\log_{10}\varepsilon$ while solving linear system with epb0 matrix depending on iteration number

Conclusions

According to numerical experiments' results, different modifications of developed algorithms are efficient while using them with different matrices.

It's worth noting that, in general, they are more efficient when applying to matrices with bigger conditional number. Particularly, when using them with relatively poor conditioned matrices olm100 and epb0, several orders of magnitude increase of accuracy was achieved and applying to epb1 matrix yields to twice less number of iterations needed to achieve given accuracy. On the other side, no positive effects were observed when applying them to relatively good conditioned matrices bfwa62, bfwa398, epb2 and epb3.

Reduction of needed number of iterations along with accuracy increase was observed upon applying incomplete basis matrix methods' preconditioners too relatively good conditioned matrix *poli* which arises from economical modelling problem. Taking this feature into consideration, similar matrices could be used as an object of further research.

Acknowledgements

The work is published with the financial support of the project ITHEA XXI Institute of Information Theory and Applications FOI ITHEA Bulgaria www.ithea.org Association and the creators and users of intellectual systems ADUIS Ukraine www.aduis.com.ua.

Bibliography

[Bohaienko, 2009] Vsevolod A. Bohaienko, Vladimir I. Kudin, Vasiliy V. Skopetsky Analysis of Computational Schemes for Geohydrodynamics Processes Modeling // Journal of Automation and Information Sciences - Volume 41, 2009 Issue 8 – pp. 1-12.

[Ke Chen, 2005] Ke Chen Matrix Preconditioning Techniques and Applications. - Cambridge University Press, 2005. - 592 p.

[Kudin, 2007] Kudin V.I., Liashko S.I., Iatsenko Iu.P., Kharytonenko N.V. Metod shtuchnykh bazysbyh matryts'// Dopovidi NAN Ukraiiny, 9, 2007, pp.30-34, in ukrainian

[Saad, 1994] Saad Y. ILUT: a dual threshold incomplete LU factorization // Numer. Linear Algebra Appl. 1(4), 387–402 [Qiaohua Liu, 2013] Qiaohua Liu, Fudao Zhang Incomplete hyperbolic Gram–Schmidt-based preconditioners for the solution of large indefinite least squares problems // Journal of Computational and Applied Mathematics 250 (2013) 210–216

[Saad, 2003] Saad Y. Iterative methods for sparse linear systems, 2 edition. - Society for Industrial and Applied Mathematics, 2003. – 528 p.

Authors' Information

Volodymyr Kudin *– Doctor of Science, senior researcher, Taras Shevchenko National University of Kyiv; e-mail:V_I_Kudin@mail.ru*

Scientific interests: optimization methods, systems analysis, methods of computations, decision making, mathematical economics.

Vsevolod Bohaienko *– PhD, senior researcher, V.M. Glushkov Institute of Cybernetics of NAS of Ukraine; e-mail: sevab@ukr.net*

Scientific interests: mathematical modeling in ecology and energetic, parallel algorithms, linear algebra, optimization methods.

ИНФОРМАЦИОННАЯ ТЕХНОЛОГИЯ РАСПОЗНАВАНИЯ ЗДАНИЙ НА СПУТНИКОВЫХ ИЗОБРАЖЕНИЯХ С ПОМОЩЬЮ НЕЧЕТКИХ НЕЙРОННЫХ СЕТЕЙ

Светлана Надеран

Аннотация: *в статье рассматривается разработанная информационная технология распознавания зданий на спутниковых изображениях, в основе которой лежит объектно-ориентированный подход к анализу спутниковых изображений и технологии мягких вычислений.*

Ключевые слова: *распознавание зданий на спутниковых изображениях, сегментация изображений, нечеткий классификатор NEFCLASS, генетические алгоритмы.*

ACM Classification Keywords: *1.4.7 Computing Methodologies - Image Processing and Computer Vision – Feature Measurement - Size and shape. G.1.6 Mathematics of Computing – Numerical Analysis – Optimization-Gradient methods. I.4.8 Computing Methodologies - Image Processing and Computer Vision – Scene Analysis - Object recognition. I.5.1 Computing Methodologies - Pattern Recognition - Models - Neural nets.*

Вступление

В связи с возрастающими возможностями получения спутниковых данных сверхвысокого разрешения усиливается необходимость в разработке методов их автоматического анализа. Рассмотрим некоторые актуальные на сегодняшний день задачи, для которых информационная технология распознавания зданий на спутниковых изображениях является эффективным решением:

Для сбора, хранения, анализа и графической визуализации пространственной информации используется ГИС. Ручная оцифровка и ввод пространственных данных в базу данных ГИС имеет недостатки в виде больших временных и материальных затрат, а также в виде высокого риска допущения ошибок, по причине человеческого фактора. Информационная технология распознавания зданий на спутниковых изображениях позволит сократить временные и материальные затраты на обновление базы данных ГИС, и повысить точность вводимых данных.

Выявление объектов самовольного строительства является актуальной задачей для городских администраций, не имеющей оптимального решения и приводящей к серьезным социально-экономическим проблемам. Автоматизированная система распознавания объектов незаконного строительства позволит оперативно выявить и устранить нарушения в вопросах землепользования.

В чрезвычайных ситуациях из-за невозможности доступа непосредственно к пострадавшей зоне наиболее эффективным методом для маркировки зданий, понесших ущерб и оперативного принятия решений для проведения спасательных операций является анализ разрушений, на основе данных предварительного дистанционного зондирования и данных ДЗЗ после стихийного бедствия. Предлагаемая информационная технология позволит оценить нанесенный ущерб в результате стихийных бедствий, сократив при этом временные затраты необходимые для визуального анализа.

Целью данной статьи является рассмотрение разработанной информационной технологии распознавания зданий на спутниковых изображениях, в которой для сегментации используется комбинация метода роста

регионов и нечеткой кластеризации С-средних, а в качестве основы для метода распознавания применяется ННС NEFCLASS.

Описание информационной технологии

В основе информационной технологии распознавания зданий на спутниковых изображениях лежит объектно-ориентированный подход в котором анализ и классификация изображений выполняется на уров¬не объектов. В качестве признаков для классификации используются статистические, текстурные и геометрические характеристики объектов. Объектно-ориентированный подход включает в себя три этапа:

Сегментация изображений.

Формирование пространства информативных признаков.

Распознавание объектов.

Сегментация является ключевым этапом обработки данных и влияет на эффективность всех дальнейших шагов анализа спутниковых изображений по причине зависимости качества получаемого в результате работы системы распознавания решения от правильно выделенных объектов. В работе [Дьяконова С.В., Зайченко Ю.П., 2012] проведен сравнительный анализ различных методов сегментации для поиска наилучшего метода применительно к задаче распознавания зданий на спутниковых изображениях. Наилучший результат сегментации был получен с помощью комбинированного метода, заключающегося в последовательном применении метода роста регионов и нечеткой кластеризации С-средних. Для повышения качества сегментации спутниковых изображений проводится морфологическая обработка, состоящая из последовательного применения двух морфологических операций: размыкание и наращивание. Морфологическая обработка обеспечивает уменьшение количества анализируемых областей за счет слияния сегментов и удаления несущественных фрагментов с точки зрения рассматриваемой задачи.

В качестве основы для метода распознавания зданий был выбран нечеткий классификатор NEFCLASS. Для формирования базы знаний ННС NEFCLASS используется предложенный в [Дьяконова С.В., Зайченко Ю.П., 2013] набор информативных признаков, характеризующих свойства верхней конструкции зданий.

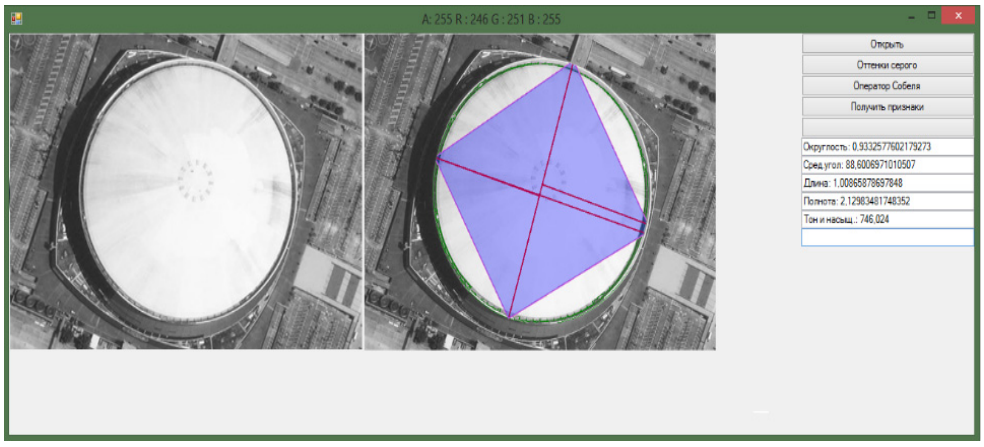


Рис.1 Пример нахождения значений информативных признаков

Для распознавания используется модификация структуры нечеткого классификатора NEFCLASS (Рис.2), позволяющая определять степень соответствия входного образца каждому из выходных классов и в

случае, если отклонение между значениями меньше ε, то ранжировать возможные решения по предпочтительности.

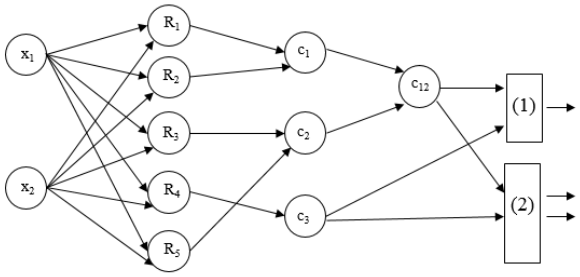


Рис.2 Архитектура модифицированной ННС NEFCLASS

Обучение ННС NEFCLASS проводится с помощью генетический алгоритм обучения, подробно рассмотренного в [Зайченко Ю.П., 2008].

Рассмотрим основные составные части информационной технологии (Рис.3). Подсистема „Интерфейс коммуникации" отвечает за прием и обработку входного пакета данных, что в свою очередь включает проверку цельности и корректности пакета. Подсистема выполняет функции по записи и выдаче геоданных, а также отвечает за приобретение электронных карт у провайдеров геоданных. Модуль „Обработчик событий" обеспечивает как синхронность, так и асинхронность выполнения событий, а также взаимодействие основных подсистем друг с другом. Подсистема „Распознавание" включает в себя модуль „ННС NEFCLASS", который реализует модифицированный нечеткий классификатор NEFCLASS, и модуль „Обработка информативных признаков", который вычисляет значения информативных признаков. Подсистема „Обучение" реализует генетический алгоритм обучения ННС NEFCLASS с помощью соответствующего модуля. Подсистема „Обработка изображений" включает в себя модуль „Grayscale", отвечающий за преобразование растровых данных из цветовой модели RGB к оттенкам серого, модуль „HSV", обеспечивающий преобразование из цветовой модели RGB в HSV, и модуль „Морфологическая обработка", реализующий морфологические операции размыкание и наращивание. Подсистема „Сегментация" реализует метод роста регионов и метод нечеткой кластеризации С-средних для сегментации растровых изображений.

ГИС БД отвечает за хранение электронных карт в виде растровых изображений и векторных данных, и за хранение информация о пространственно-распределенных объектах, которая содержится в БД в виде электронных таблиц. Пространственные данные делятся на четыре основных типа пространственных объектов: точка, линия, полигон и поверхность, все эти объекты характеризуется местоположением в пространстве и набором атрибутивных данных. Таким образом, пространственные данные позволяют описать местоположение пространственных объектов, их топологические свойства и содержательные характеристики. Хранящиеся атрибутивные данные дополняют пространственные данные с необходимой степенью детализации.

БД Конфигурации отвечает за хранение настроек параметров конфигурации подсистемы сегментации, хранение настроек ННС NEFCLASS, таких как параметры функции принадлежности, нейроны входного и выходного слоя, и база нечетких правил. В БД Конфигурации сохраняются индивидуальные настройки пользователей, в том числе персональные данные пользователей системы.

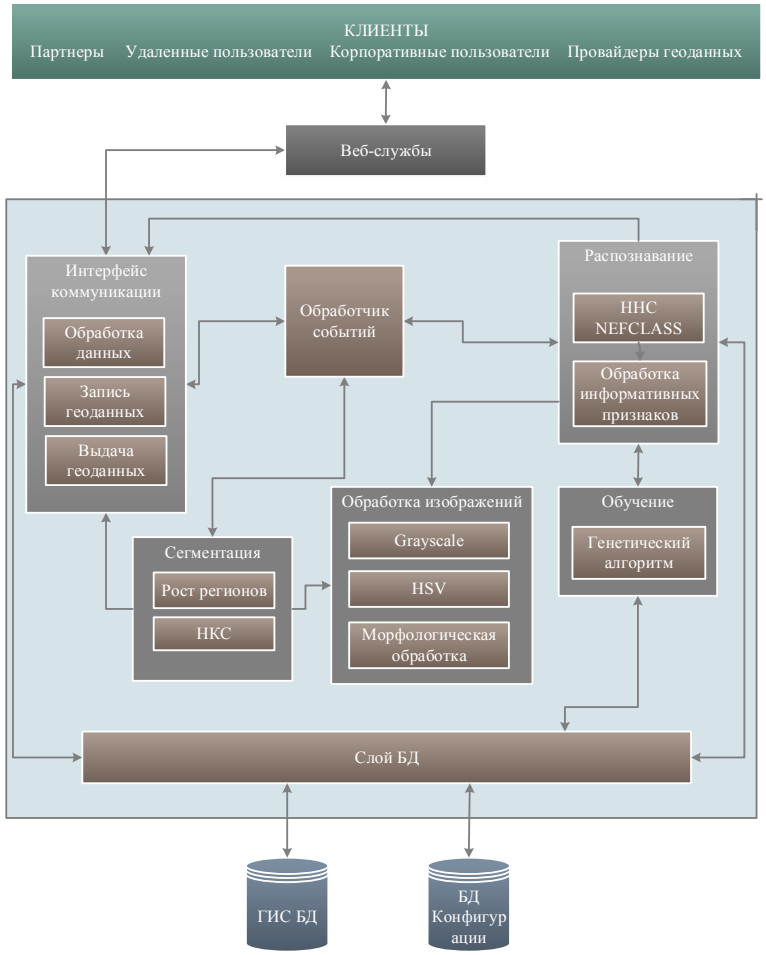


Рис.3 Структурная схема

Экспериментальные исследования

В работе использовались спутниковые изображения IKONOS и GeoEye-1. Пространственное разрешение спутниковых данных составляет 0.8 м и 0.5 м соответственно. Входное изображение представляет собой растровые данные в цветовой модели RGB, которые преобразуются в оттенки серого с последующей сегментацией с помощью комбинированного метода.

Выходам нейронной сети NEFCLASS соответствуют четыре класса $\{c_1, c_2, c_3, c_4\}$, входам – пять информативных признаками $\{x_1, x_2, x_3, x_4, x_5\}$

c_1 - здание с треугольной крышей;

c_2 - здание с круглой крышей;

c_3 - здание с плоской крышей;

c_4 - не здание;

x_1 – округлость крыши;

x_2 - среднее значение углов;

x_3 - длина крыши;

x_4 - полнота;

x_5 - среднее значение интенсивности оттенка крыши.

Рассмотрим способ формирования базы правил. Каждое правило имеет следующий вид:

ЕСЛИ x_1 является μ_1, …, x_n является μ_n;

ТО образец (x_1 , … , x_5) принадлежит классу i;

где i название класса образа,

n количество информативных признаков.

Для обучения параметров функций принадлежности выбран генетический алгоритм обучения.

Обучающая выборка состояла из 700 образцов, выборка для тестирования включала 400 образцов. Эксперимент был поставлен следующим образом. Каждое изображение обрабатывается и сегментируется. Для генерации базы правил каждому сегменту на изображении ставится в соответствие один из заданных классов. Таким образом, создается база правил. Проводится обучение нейронной сети выбранным методом. Тестирование обученной ННС проводится с помощью выборки для тестирования. На Рис.4 представлен графический интерфейс информационной технологии.

Рис.4 Графический интерфейс информационной технологии

По результатам экспериментальных исследований было получено 12,50% ошибочно классифицированных образцов.

Заключение

В статье рассмотрена разработанная информационная технология распознавания зданий на спутниковых изображениях. Описаны основные принципы объектно-ориентированного подхода к анализу спутниковых изображений. Рассмотрены подходы к сегментации и распознаванию, которые легли в основу информационной технологии, а также основные компоненты из которых она состоит. Описаны проведенные экспериментальные исследования по результатам которых точность распознавания составила 87,50%.

Благодарности

Статья частично финансирована из проекта ITHEA XXI Института Информационных теорий и Приложений FOI ITHEA и консорциума FOI Bulgaria (www.ithea.org, www.foibg.com)

Литература

[Дьяконова С. В., Зайченко Ю.П., 2012] Дьяконова С. В., Зайченко Ю.П. Анализ методов сегментации спутниковых изображений. - Вісник НТУУ „КПІ" Інформатика, управління та обчислювальна техніка, №57, 2012. – с. 118-123.

[Дьяконова С. В., Зайченко Ю.П., 2013] Дьяконова С. В., Зайченко Ю.П. Подход к решению задачи автоматизированного обнаружения зданий на спутниковых изображениях. - Вісник НТУУ „КПІ" Інформатика, управління та обчислювальна техніка, №58, 2013. – с.51-55.

[Зайченко Ю.П., 2008] Зайченко Ю.П. Нечёткие модели и методы в интеллектуальных системах. Учебное пособие для студентов высших учебных заведений. – К.: „Издательский Дом „Слово"„, 2008. – С. 344.

Информация про автора

Светлана Владимировна Надеран *– аспирантка Национального технического университета Украины „КПИ", адрес электронной почты: sv.naderan@gmail.com*

Основные сферы научных исследований автора: *применение нечеткого классификатора NefClass к задаче распознавания зданий на спутниковых изображениях*

Automated Building Extraction System Applied On High Resolution Satellite Imagery Using Fuzzy Neural Network

Svetlana Naderan

Abstract: *The article is devoted to the development of the new Automated Building Extraction system applied on High Resolution Satellite Imagery, which is based on an object-oriented approach and soft computing.*

Keywords: *building recognition, satellite imagery, neuro-fuzzy classifier NEFCLASS, genetic algorithm.*

THE ANALYSIS OF NEURAL NETWORKS' PERFORMANCE FOR MEDICAL IMAGE CLASSIFICATION

Kateryna Malyshevska

Abstract*: The tissue type classification is presented using the neural networks. The multi-spectral images of uterine cervix were segmented using self-organizing Kohonen maps and k-means algorithm. Then, the classification of tissue types from computed segments was made using the cascade neural network, back propagating neural network, and RBF network. The basics of neural networks were briefly explained. The results were presented and analyzed, based on which, the conclusions were made.*

Keywords*: neural networks, decision making, intellectual systems, segmentation.*

ACM Classification Keywords*: I.2.1 Applications and Expert Systems - Medicine and science*

Introduction

In this work, the ability is studied for the tissue type classification from the multi-spectral images of a uterine cervix. This problem is dictated by the necessity of an early diagnosis of the medical condition, using the computer system, which will help the physician to detect the high risk areas to become malignant.

For experimental data, the information about 113 patients was used, who have been diagnosed in the hospital. The physician made the diagnosis by classifying the tissue types, taken for the biopsy. The area where the biopsy was conducted is specified on images. Images were obtained with the use of the optical system introduced in the medical university of Arizona (USA), with whose help multi-spectral electronic images of cervix tissues were made [Schoonmaker, 2007].

For each patient, a 16-channel image is made:

- Four bands of reflected polarized light with the polarizer parallel to the source light;
- Four bands of reflected polarized light with the polarizer perpendicular to the source light;
- Eight bands of fluorescence, using a 365 nm (two quad filter set).

There can be six tissue types on a uterine cervix. Three of them are considered safe (normal) and exist on a normal healthy organ: Squamous epithelium, Columnar epithelium, and Metaplasia – benign tissue transformation. However, there are also three other tissue types, which are dangerous and can be either predecessors of a cancer, or indicate its presence (CIN1 – mild dysplasia, CIN2 – moderate dysplasia, CIN3 – severe intraepithelial neoplasia, the notion that combines severe dysplasia and intraepithelial cancer) [Koss, 1989]. After the classification of each epithelium type, the classification of the precancerous and cancerous epithelium type (CIN1+CIN2+CIN3) is implemented.

The statement of the problem

Similar tissue types have similar optical properties and distinctions between different tissue types are greater, then the distinctions between the similar tissue types of different patients; therefore, using 16-channel images, it is possible to identify similar optical tissue properties. The idea is to develop the system, which can help with recognition of problematic areas in the uterine cervix, to point the doctor the exact area for biopsy.

Methods

Based on the assertion discussed before, the system was developed, which could detect the tissue type from images with the help of neural networks.

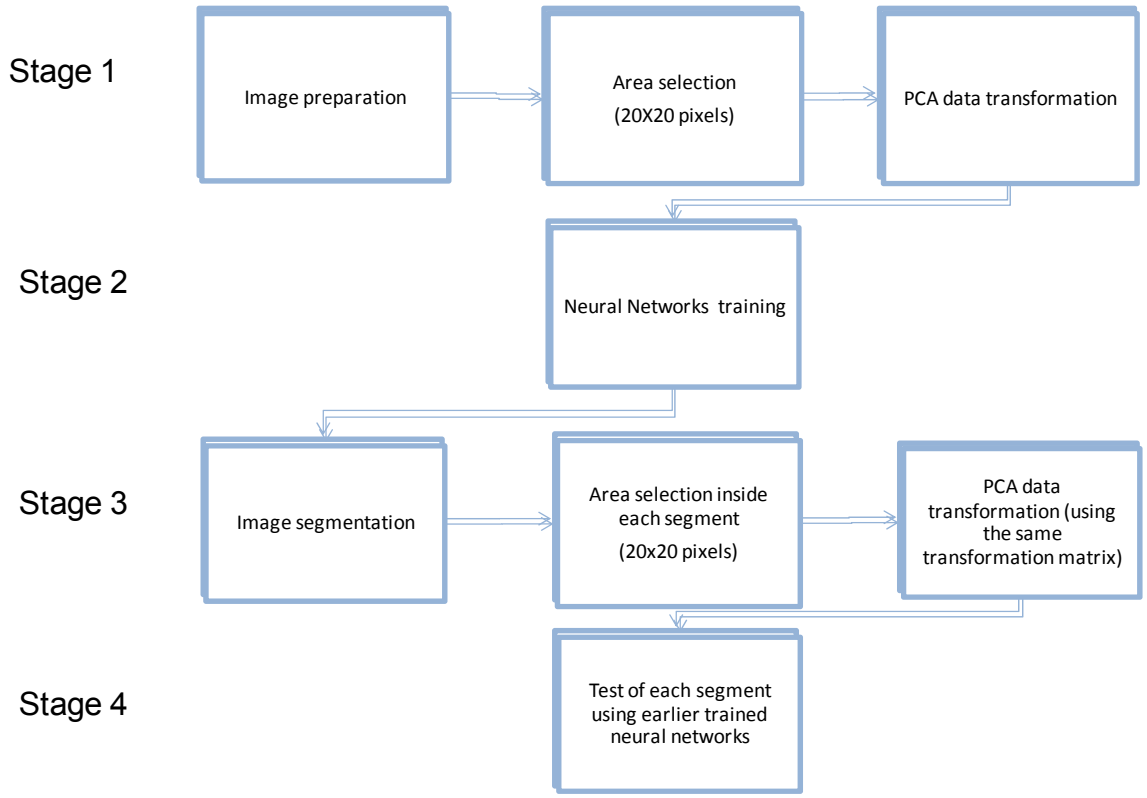


Figure 1. Information system for medical images classification

At the stage 1, initial data have been prepared. In the place where the biopsy was made, an area of 20 by 20 pixels is selected; this way, the texture pattern is taken into an account from the place where the biopsy was made. For every case, this approach generated 16*20*20 = 6400 inputs. Because of this large number, they were transformed into 14 inputs with the help of principal component analysis (PCA).

Then, at the stage 2 the neural networks were trained using the obtained data.

At the stage 3, images were segmented using self-organizing Kohonen maps and k-means clustering. Both methods were used at this step to determine their efficiency, and in the result section the work of each method is shown.

After segmenting the images, an area of 20 by 20 pixels was selected on each segment and transformed using PCA (using the transformation matrix from the first stage).

At the stage 4, each segment was tested using earlier trained neural networks.

To choose what neural network to use in the system the series of tests were implemented. The neural networks that were tested are Back propagation neural networks, Radial Basis Function Networks (RBF), and Cascade neural networks. The results of use the above mentioned networks follow.

Results

We conducted six experiments. In all experiments, we had 100 observations. Each observation corresponded to a case where biopsy results were known. In other words, inputs were part of an image where the biopsy was made and six outputs were the results of this biopsy (fraction of each epithelium type). The data set was split into the training (80) and testing (20) subsets. Also, the cross validation was used.

The first experiment was concerned with determining the fraction of the each epithelium type (all six of them). In other words, each observation consisted of a multichannel image of a given spot (inputs) with the biopsy results (six outputs). The biopsy results included the fraction of each epithelium type. Three types of neural networks were tested for their capability to correctly predict the biopsy results (the fraction of each epithelium type).

The second experiment was concerned with determining whether a given tissue had transitioned into a dangerous state. Instead of predicting the fractions of each epithelium type, we tried to predict the fraction of precancerous or cancerous states. To accomplish this, we essentially conducted the first experiment with post-processing: we used observations with full biopsy results (including the fraction of every epithelium type) and checked the results for combined fraction of three dangerous epithelium types (CIN1+CIN2+CIN3).

The third experiment was similar to the second, except we preprocessed the observations to have only one output (the fraction of combined dangerous epithelium types) and trained networks on such observations.

The results are shown in the following tables.

Table 1. The forecast results of the epithelium type (mean error)

	Cascade neural network	Back propagation neural network	RBF network
6 epithelium types	0.0479	0.0584	0.0610
CIN1+CIN2+CIN3 (before training)	0.0832	0.1089	0.0569
CIN1+CIN2+CIN3 (after training)	0.0768	0.0865	0.0446

We can see from Table 1, that RBF network demonstrates the best results for CIN1+CIN2+CIN3 (combined fraction of three dangerous epithelium types) which is important, because the goal is to find an area for further biopsy. Cascade neural networks were the best in predicting the fraction of each epithelium type (the first row of data). Also, the second and the third rows of data show that it is much more effective to train networks to predict percentage for each epithelium type and only then combine results to obtain the data for CIN1+CIN2+CIN3 ("after training").

The rest of the experiments studied the ability to correctly predict the correct class for each epithelium type. For each epithelium type, we split the range from 0 to 1 (a fraction) into four equivalence classes of the same size. Thus, our goal was not to compute the fraction as accurate as possible, but to correctly predict the class. In other words, it was important to determine whether a given tissue was in danger to become cancerous. In each observation, the inputs were the same as in the first set of experiments, but the outputs were the equivalence classes. The forth experiment studied the capability to predict a correct class for each epithelium type. The fifth and the sixth experiments studied the capability to predict the class of combined dangerous epithelium types (CIN1+CIN2+CIN3). The difference between the fifth and the sixth experiments is the same as the difference between the second and the third experiments: combining CIN1+CIN2+CIN3 either after or before the training. For each such an experiment, we counted the number of correct predictions of classes ("correct"), the number of

predictions where the predicted class was less than the correct class ("false negative"), and the number of predictions where the predicted class was greater than the correct class ("false positive"). Generally, false negative errors should be minimized because they are more dangerous than false positive errors.

The results are shown in the following tables.

Table 2. The forecast results of the 6 epithelium types

	Cascade neural network	Back propagation neural network	RBF network
False negative	0.0854	0.0987	0.1033
False positive	0.0433	0.0417	0.0508
Correct	0.8713	0.8596	0.8458

Table 2 shows the results where the neural networks try to predict the class for each of the six epithelium types. All networks demonstrate good results with cascade network being the best.

Table 3. The forecast results of the precancerous and cancerous epithelium type (CIN1+CIN2+CIN3) (before training)

	Cascade neural network	Back propagation neural network	RBF network
False negative	0.1550	0.1650	0.1350
False positive	0.0700	0.0575	0.0250
Correct	0.7750	0.7775	0.8400

Table 3 shows the results where neural networks try to predict an equivalence class for precancerous and cancerous epithelium state. The networks were trained for CIN1+CIN2+CIN3 data. RBF neural network demonstrated the best results.

Table 4. The forecast results of the precancerous and cancerous epithelium type (CIN1+CIN2+CIN3) (after training)

	Cascade neural network	Back propagation neural network	RBF network
False negative	0.1200	0.1250	0.1975
False positive	0.0925	0.0675	0.0400
Correct	0.7875	0.8075	0.7625

Table 4 shows the results where neural networks try to predict an equivalence class for precancerous and cancerous epithelium state. The networks were trained for each of 6 epithelium types and the results were summed. The Back propagation neural network demonstrated the best results in terms of the correct

classification; however, the misclassification rate in terms of false negative results is the lowest for the cascade neural network, which, in our case, is the most important.

In addition to earlier described experiments, we drew histograms to show the distribution of the differences between the predicted value for epithelium fraction and the correct one (the biopsy result). The data are for every observation in the testing set and for each epithelium type.

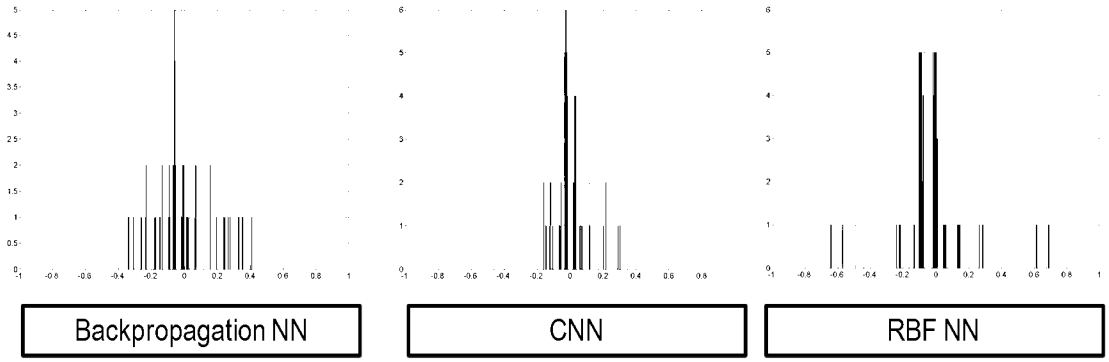


Figure 2. The histograms of forecast results for 6 types of epithelium

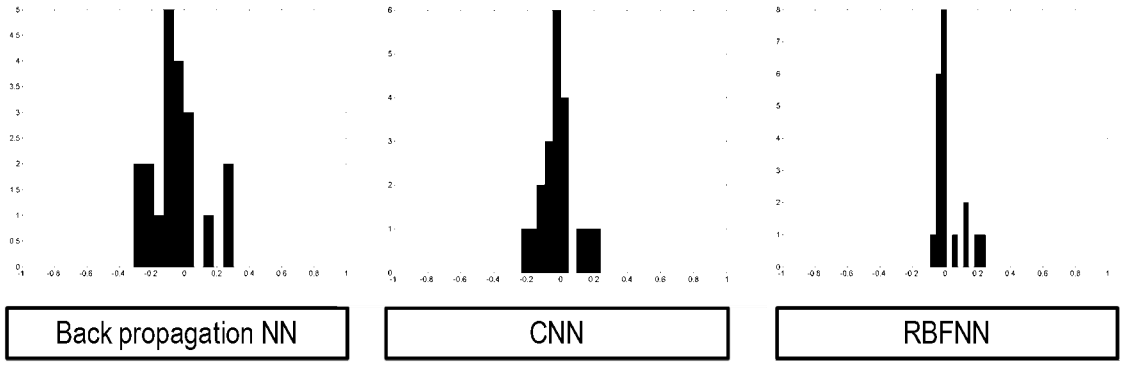


Figure 3. The histograms of forecast results for the CIN1+CIN2+CIN3 epithelium types

Figure 2 shows the histograms of forecast results for six types of epithelium with 120 data points (6 x 20). Here, the differences (errors) are concentrated in a narrow interval around zero.

Figure 3 shows the histograms of forecast results for CIN1+CIN2+CIN3 epithelium types. The data are for every observation in the testing set with the total of 20 data points. We can see that, in this case, the best results are provided by RBF neural network demonstrates the best results (data points are concentrated in a very narrow interval around zero).

Conclusion

This work studied the application of neural networks for the problem of tissue type classification in uterine cervix from multi-spectral images.

- Conducted studies showed that neural networks could accurately predict the epithelium type and determine high risk areas;

- Cascade neural networks were the most accurate in predicting the fraction of each epithelium type separately;
- RBF network demonstrated the best results for CIN1+CIN2+CIN3 (combined fraction of three dangerous epithelium types);
- It is much more effective to train networks to predict percentage for each epithelium type and only then to combine results to obtain the data for CIN1+CIN2+CIN3 ("after training") (another less effective approach was to train the networks on already combined CIN1+CIN2+CIN3 data);
- Neural networks can accurately predict an equivalence class for each of the six epithelium types with cascade neural network demonstrating the best results;
- Neural networks can accurately predict an equivalence class for precancerous and cancerous epithelium state with RBF neural network demonstrating the best results.

Bibliography

[Schoonmaker, 2007] Schoonmaker J. et al. Automatic Segmentation of Uterine Cervix for in vivo Localization and Identification of Cervical Interaepithelial Neoplasia // Apogen Technologies 2007

[Koss, 1989] Koss L.G. The complex test for cervical cancer detection. In: The Journal of the American Medical Association. – 1989

[Zaychenko, 2004] Zaychenko Yu.P. Osnovy proektuvannya intelektual'nykh system. Navchal'nyy posibnyk – K.: Vydavnychyy Dim "Slovo", 2004

[Kovacevic, 1997] Domagoj Kovacevic, Sven Loncaric Radial Basis Function-based Image Segmentation using a Receptive Field //Computer-Based Medical Systems. – 1997. Proceedings., Tenth IEEE Symposium – 126-130

Scott E. Fahlman, Christian Lebiere: The Cascade-Correlation Learning Architecture. NIPS 1989: 524-532

[Kohonen, 1997] Kohonen T. 1997 "Self-Organizing Maps"(2-nd edition), Springer, 1997.

[Starikov, 2000] Starikov A. 2000 BaseGroup 2000.

[Chubukova, 2008] Chubukova I.A. Data Mining // BINOM. Laboratoriya znaniy, 2008 g., 384 str.

Authors' Information

Kateryna Malyshevska *– NTUU 'KPI' Ph.D. student, Kyiv, Ukraine; e-mail: kate.inv@gmail.com*

Major Fields of Scientific Research: Neural networks, Intellectual systems of decision making.

TABLE OF CONTENTS