Optimal Threshold Estimation Using Prototype Selection

Uri Lipowezky and Victor Shenkar

Tiltan System Engineering Ltd.
35 Hayarkon Street, Beney – Beraq 51204, Israel
Tel 972 - 3 - 5780711
Fax 972 – 3 –5780714
Uri@tiltan-se.co.il

Abstract. A technique is proposed for choosing the thresholds for a number of object detection tasks, based on a prototype selection technique. The chosen prototype subset has to be correctly classified. The positive and negative objects are introduced in order to provide the optimization via empirical risk minimization. A Boolean function and its derivatives are obtained for each object. A special technique, based on the fastest gradient descent, is proposed for the sum of Boolean functions maximization. The method is applied to the detection task of house edges, using its images in aerial photos. It is shown that proposed method can be expanded to solving of a wide range of tasks, connected to the function optimization, while the function is given in vertices of a 2^n single hyper - cube.

Keywords: Prototype selection, Sum of Boolean function optimization, Edge detection.

1 Introduction

A wide variety of image processing tasks such as images compression [17], image matching for motion estimation and 3D modeling [2], object decipherment [14], edge detection [4] and so on, demands in its final stage a threshold selection procedure to distinct between correct and incorrect results. Usually these thresholds are supposed to be pre-defined and no techniques are given for their correct choice, in spite of strong influence of these thresholds on the final results. It was shown in previous works [14], that correct parameter estimation can be achieved using two sets of samples: a set of positive samples (objects) $E = \{e_1, e_2, ..., e_m\}$ and set of negative samples (anti – objects) $A = \{a_1, a_2, ..., a_p\}$. Sometimes [17] it can be found an automatic procedure for object and anti – object detection, however in common case samples are pointed out manually. Every object e_i, a_j, $i = 1,...,m$, $j = 1,...,p$ can be represented by n – dimensional vector of features $\varphi_1, \varphi_2, ..., \varphi_n$, where $\varphi_i \in R_1$ is supposed to be a real value from limited interval $[\varphi_i^{min}, \varphi_i^{max}]$. Without loss of generality, let us assume that for each i $E[\varphi_i(E)] > E[\varphi_i(A)]$, where $E[\varphi_i(E)]$ and $E[\varphi_i(A)]$ are average values

F.J. Ferri et al. (Eds.): SSPR&SPR 2000, LNCS 1876, pp. 737-746, 2000.

of feature φ_i distribution for samples of an object and an anti - object. Under this assumption thresholds $t_i = \min_{j=1,..,m} \varphi_i(e_j) i = 1,.., n$ provide correct classification for all objects and misclassification for anti-objects with values $\forall_{i=1,...,n}[\varphi_i \geq t_j]$. In the other words object $e_i = (X_{i1}, X_{i2}, ..., X_{in})$, where X_{ij} is value of feature φ_j, $j = 1,..., n$ for object e_i, will be correctly classified if for each feature φ_j value X_{ij} it exceeds or equals to the threshold t_j, i. e., $\forall_j[X_{ij} \geq t_j]$. To simplify the expression let us introduce a function similar to the Kroenecker notation:

$$\delta(X, t) = 1 \text{ if } X \geq t \quad \text{and } 0 \quad \text{oterwise}. \tag{1}$$

Using (1) the expression for true object classification can be rewritten as

$$f_i = \prod_{j=1}^{n} \delta(X_{ij}, t_j) \tag{2}$$

(2) becomes one for true classification and zero for misclassification.

The problem is to define optimal thresholds $t_j, j = 1,.., n$ in the sense of empirical risk minimization [20], i. e., to minimize functional

$$\psi = \sum_{i=1}^{m} \left[1 - \prod_{j=1}^{n} \delta(X_j(e_i), t_j) \right] + \sum_{i=1}^{p} \prod_{j=1}^{n} \delta(X_j(a_i), t_j), \tag{3}$$

where $X_j(e_i)$ and $X_j(a_i)$ are values of feature φ_j for a object and an anti-object. The first sum of (3) expresses the misclassification rate and it changes in the points $t_j = X_j(e_i)$. When t_j is increased the sum is reduced and vice versa. The second sum of (3) expresses the false alarm rate. It changes in the points $t_j = X_j(a_i)$ and it decreases when t_j is increased, so if there exists a situation $X_j(e_{i1}) < X_j(a_{i1}) \leq X_j(a_{i2}) \leq ... X_j(a_{ik}) < X_j(e_{i2})$ there is no reason to choose any one of $X_j(a_{ik})$ in order to minimize (3). Thus, it is enough to considerate only values $X_j(e_i)$. If for any feature φ_j value $t_j > X_j(e_i)$ is chosen e_i will not be correctly classified. Hence, the following approach can be used to threshold selection. There has to be selected a subset $E' \subseteq E$ of objects E that should be correctly classified using thresholds

$$t_j = \min_{e_i \in E'} X_j(e_i) \quad j = 1,..., n \tag{4}$$

and it has to minimize sum (3). Using this definition the task of threshold selection can be considered as a task of prototype selection (PS).

The PS task has a rather long history. In the works of P. E. Hart [9] and G. W. Gates [8] the PS task was defined as a task of selection the most representative prototypes for $k - NN$ classification. To overcome the problems of classification quality

a hypothesis of local statistical compactness was proposed [6]. There is a number of algorithms in this framework [11], [3], [13]. Some of them e. g., [13] allow combination of features (genes) to generate new prototypes.

Other approaches can be borrowed from the feature selection task. In expression (3) positive samples e_i can be treated as a feature set and negative samples can be used for feature selection. All of approaches in the framework of feature selection can be divided into optimal and sub-optimal approaches. The optimal approach involves exhaustive search [10] and mathematical programming approach [7]. The exhaustive search of all available prototype combinations demands $2^m L$ recognition operations, where m is the initial number of prototypes and L is the test sample volume. Evidently it is impractical in case of (3) minimization. The mathematical programming approach demands $\alpha 2^m L$ operations, where $\alpha \in [0.008...0.04]$ also is not applicable for (3) optimization, since $L = m + p$ can be very large.

A sub-optimal approach supposes an existence of a finite set of optimal vertices in a m-dimensional hyper-cube, where each vertex corresponds to a combination of the prototypes. Let us assume, that binary vector $b = \{b_1,..., b_m\}$ corresponds to the initial prototype set E. If $b_i = 1$ prototype e_i is included into optimal subset E'. Using random generation of vector b and it substitution into (3) one can choose an optimal E', related to the minimum (3). If there are N_{max} optimal subsets E' with the same value (3) the probability to reach one of them is $p = N_{max} / 2^m$ and hence [15] it is enough

$$T = -2^m \ln(1-\eta) / N_{max} \qquad (5)$$

runs of a search algorithm to reach a global minimum with probability η. There are two main directions in the framework of the sub-optimal approach, yielding a single solution of the problem. They are deterministic and stochastic approaches [10]. The stochastic approach [1] is based on a punishment and encouragement technique for different subsets. It is similar to genetic approach [13] and actually converges to an adaptive random search. The family of deterministic algorithms [12] can be represented for (3) optimization by forward and backward methods, which are based on sequential deleting/addition of an appropriate prototype.

Sometimes as for 1-NN classification [15] functional (3) can be expressed analytically in respect to probabilities of prototypes present in the optimal subset, and the optimal subset can be found using classical optimization procedures [3]. The purpose of the work is to show that this approach is also possible for (3) minimization and so one can derived an algorithm that combines the advances of the above mentioned approaches.

In Section 2 an illustrative example for $n = 1$ is given. In Section 3 an algorithm for the common n-dimensional case is developed. Section 5 includes experiments and conclusions.

2. One-Dimensional Case

Let us consider an illustrative example of threshold selection. Let $n = 1$ be the number of features, $E = \{e_1,..., e_m\}$ is an initial prototype set, where prototypes are sorted up so that $\varphi(e_m) > \varphi(e_{m-1}) > ... > \varphi(e_1)$ and $A = \{a_1,..., a_p\}$ is a set of negative objects which are also sorted up. If threshold t is chosen using (4) e_1 is recognized correctly if and only if $e_1 \in E'$. Prototype e_2 can be classified correctly if e_1 or e_2 are included into E', since $\varphi(e_1) \le \varphi(e_2)$ and so on e_i can be recognized if one of $e_j, j \le i$ is included into E'. Using a set of Boolean variables $b = \{b_1,..., b_m\}$, where each b_i is associated with e_i the condition for e_i correct classification can be written as $f(e_i) = b_1 \vee b_2 \vee ... b_i = \bigvee_{j=1}^{i} b_j$ and for negative object $f(a_i) = \bigvee_{\varphi(e_j) \le \varphi(a_i)} b_j$. Since the aim is to recognize e_i and not to recognize a_i functional (3) can be rewritten as

$$\psi(b) = \sum_{i=1}^{m} \left(\bigvee_{j=1}^{i} b_j = \text{true} \right) + \sum_{i=1}^{p} \left(\neg \bigvee_{\varphi(e_j) \le \varphi(a_i)} b_j = \text{true} \right) \tag{6}$$

Let us assume that $0 \le p_i \le 1$ is a fuzzy variable [19] associated with e_i. If a lot of subsets E' are obtained using a random generation, then p_i means a probability that prototype e_i belongs to the optimal prototype subset. Hence, $q_i = 1 - p_i$ means a fuzzy variable associated with $\overline{e_i}$. Since prototypes are supposed to be independent (6) can be rewritten using q_i as follows

$$\psi(Q) = \sum_{i=1}^{m} \left(1 - \prod_{j=1}^{i} q_j \right) + \sum_{i=1}^{p} \prod_{\varphi(e_j) \le \varphi(a_i)} q_j \tag{7}$$

The problem is to find maximum of the analytical function $\psi(Q)$, where $Q = \{q_1,..., q_m\}$. There are some properties of (7) that makes the problem easier. It follows from (7) that every q_i is represented in each multiplication not more than once. Therefore for every q_i

$$\frac{\partial^2 \psi(q_1, q_2,..., q_m)}{\partial q_i^2} = 0 \tag{8}$$

Such a function can be called poly-line function [15]. Generally speaking, there is no problem to calculate derivatives of the poly-line function. For instance, in case (7) the expression for derivative is

$$\frac{\partial \psi}{\partial q_k} = -\sum_{i=k}^{m} \prod_{j=1}^{k-1} q_j \prod_{j=k+1}^{i} q_j + \sum_{\varphi(a_i) \geq \varphi(e_k)\, \varphi(e_j) \leq \varphi(a_i);\, j \neq k} \prod q_j \quad (9)$$

So calculations of the gradient almost do not demand additional operations and optimization based the fastest gradient descent [5] can be used. From (9) it follows that $-m \leq \partial\psi / \partial q_k \leq p$ that fulfils the Lipschiz condition what is necessary for the existence of the maximum (7). Let us prove that poly-line function can be represented as a convex combination of its values in the vertices of 2^m single hyper-cube as follows:

$$\psi(q_1,...,q_m) = \sum_{i=0}^{2^m-1} \lambda_i \psi(B_m(i)), \quad (10)$$

where $0 \leq \lambda_i \leq 1$, $\sum_{i=1}^{2^m-1} \lambda_i = 1$, $\lambda_i = \prod_{j=1}^{m} y_j(i)$ and $B_m(i)$ is binary m-dimensional decomposition of the integer value i, where i is one of the vertices and each bit $j = 1,..., m$ of the decomposition with respect to the associated variable q_j. Variable $y_j(i) = q_j$ if bit j in $B_m(i)$ is equal to one and $y_j(i) = 1 - q_j$ otherwise.

Let us consider the following induction. If $m = 1$ poly-line function (10) can be represented as $\psi(q) = aq + b$ or using decomposition $\lambda_0 = 1 - q$, $\lambda_1 = q$ so $\lambda_0 + \lambda_1 = 1$ and $\psi(q) = (a + b)q + b(1 - q)$. Let us assume that it is correct for all $m = 1,2,..,d$ and let us try to prove that it is correct also for $m = d + 1$. From (8) it follows that

$$\psi(q_1,..,q_d,q_{d+1}) = \psi_1(q_1,..,q_d)q_{d+1} + \psi_2(q_1,..,q_d), \quad (11)$$

where ψ_1, ψ_2 are poly-line functions that can be represented in form (10). Substitution (10) into (11) gives

$$\psi(q_1,..,q_d,q_{d+1}) = \sum_{i=0}^{2^d-1} \lambda_i q_{d+1} [\psi_1(B_d(i)) + \psi_2(B_d(i))] + \sum_{i=0}^{2^d-1} \lambda_i (1 - q_{d+1}) \psi_2(B_d(i)) \quad (12)$$

Thus, $\psi(q_1,...,q_d,1) = \psi_1 + \psi_2$, $\psi(q_1,...,q_d,0) = \psi_2$ and $\sum_{i=0}^{2^d-1} \lambda_i q_{d+1} + \lambda_i (1 - q_d) = \sum_{i=0}^{2^d-1} \lambda_i = 1$. Therefore all values in the hyper-cube are limited within values in its vertices and so a global maximum of (10) can be found only in a vertex of hyper-cube, hence for optimization algorithm it is enough to consider only values $q_i \in \{0,1\}$.

The main corollary following from this property is that a variety of tasks that are connected with optimization on the 2^m hyper-cube, when the values of the functional could be found in every vertex can be solved using poly-line function optimization. However this corollary does not show how this poly-line function and its derivatives

can be found except of trivial case (10) when all 2^m values are known. In most of the pattern recognition tasks functional (3) can be represented as sum of Boolean functions as (6). Let us prove that if Boolean function $F(x_1,...,x_n)$ is defined on all of 2^n sets of variables $x_i \in \{0,1\}$, there exists a poly-line function $\psi(p_1,...,p_n)$, where p_i is the probability that x_i belongs to the optimal subset.

Indeed every Boolean function $F(x_1,...,x_n)$ can be represented using its disjunctive normal form $F(x_1,...,x_n) = x_i A \vee \overline{x}_i B \vee C$, where A, B and C do not depend on x_i. The expectation of $\psi(p_1,...,p_n)$ respected to $F(x_1,...,x_n)$ is

$$\psi(p_1,...,p_n) = 1-(1-\Pr(C))(1-\Pr(x_i A \vee \overline{x}_i B \mid C)). \quad (12)$$

Since events $x_i A$ and $\overline{x}_i B$ are independent (12) can be rewritten as

$$\psi(p_1,...,p_n) = 1-[\Pr(C)]\{p_i[\Pr(A \mid C)-\Pr(B \mid C)]-1-\Pr(B \mid C)\}. \quad (13)$$

and hence, $\partial^2\psi/\partial p_i^2 = 0$. Since x_i is an arbitrary variable (13) is correct for any $i = 1,..,n$. Formula (17) gives the technique to solve task (3) in common case.

3. Common n-Dimensional Case

Let us consider once more expression (2). Object e_i will be correctly classified if $f_i = \underset{j=1}{\overset{n}{\&}} \underset{x_{jk} \le X_{ji}}{\vee} e_k$ is true. Using disjunctive form the same expression can be written also for an anti-object as follows:

$$\overline{f}_i = a_i = \underset{j=1}{\overset{n}{V}} \underset{x_{jk} \le X_{ji}}{\&} \overline{e}_k \quad (14)$$

If it is supposed that $f_i, a_i = 1$ if (14) is true and zero if it is false, the maximum of $\psi = \sum_{i=1}^{m} f_i + \sum_{i=1}^{p} a_i$ is equal to minimum of (3). Any one of expressions (14) can be simplified using common variable extraction: $ab \vee ac = a(b \vee c)$ and group elimination: $ab \vee a = a$ and hence, (14) can be represented as

$$\overline{f}_i = a_i = \underset{e_j \in G_0}{\&} \left(\underset{k=1}{\overset{m}{\vee}} \underset{e_j \in G_k}{\&} e_j \right), \quad (15)$$

where G_0 is common group, $m \le n$ is reduced number of groups and G_k is a partial group.

Let us find derivatives of (15). Since inclusion of any e_i does not depend on inclusion exclusion of any other $e_j, j \ne i$ into the optimal subset, it can be written:

$$1-f_i = a_i = \prod_{j\in G_{0\setminus}} (1-p_j)\Pr\left(\bigvee_{k=1}^{m} \mathop{\&}_{e_l\in G_k} \bar{e}_l\right) \tag{16}$$

from (16) is follows that for any $e_j \in G_0$

$$\frac{\partial f_i}{\partial p_j} = -\frac{\partial a_i}{\partial p_j} = \mathop{\&}_{e_k\in G_0, k\neq j} \bar{e}_j\left[\bigvee_{k=1}^{m} \mathop{\&}_{e_l\in G_k} \bar{e}_l\right] \tag{17}$$

For any $e_j \in G_k$ expression (15) in parenthesis can be written as $R = \bigvee_{k=1}^{m} \&_{e_j\in G_k} \bar{e}_j = \bar{e}_j F_1 \vee F_2$, where $F_1 = \bigvee_{G_k\supseteq e_j} \&_{e_l\in G_k, l\neq j} \bar{e}_l$ and $F_2 = \bigvee_{G_k\not\supset e_j} \&_{e_l\in G_k, l\neq j} \bar{e}_l$. The derivative of R is $dR/de_j = R(e_j = 1) = R(e_j = 0) = F_2 - F_1 \vee F_2 = -F_1 F_2$ and finally

$$\frac{\partial f_i}{\partial e_j} = -\frac{\partial a_i}{\partial e_j} = \mathop{\&}_{e_l\in G_0} \bar{e}_l\left(\bigvee_{G_k\supseteq e_j} \mathop{\&}_{e_l\in G_k} \bar{e}_l\right)\left(\mathop{\&}_{G_k\not\supset e_j} \bigvee_{e_l\in G_k} e_l\right). \tag{18}$$

Since f_i, a_i and their derivatives could be calculated using the same groups G_k computation of the gradient $\mathrm{grad}\psi = (g_1,...,g_n)$ almost does not demand additional operations. It allows us to derive the following optimization algorithm, based on gradient descent [5].

1. For every $f_i, i = 1,..,m$ and $a_i, i = 1,..,p$ obtain formulae (15) and save them via groups $G_k, k = 0,1,..$
2. Using random generation, choose an arbitrary vertex $Q = (q_1,..,q_n)$, where $q_i \in \{0,1\}$.
3. For a chosen vertex Q calculate ψ and $\mathrm{grad}\psi = (g_1,...,g_n)$, where each derivative g_i is sum of (17), (18) for all f_i and a_i.
4. Define next vertex $P = (p_1,..,p_n)$, where $p_i = q_i$ if $q_i = 1$ and $g_i \geq 0$ or $q_i = 0$ and $g_i \leq 0$. Otherwise $p_i = 1 - q_i$. If $P = Q$ (the same vertex) the process is over and local maximum point is reached, otherwise the next step has to be done.
5. If $\psi(P) \geq \psi(Q) + \max_B |g_i|$, where B includes indices i for which $p_i \neq q_i$ as P is the next vertex. Otherwise the next vertex is $Y = (q_1,..,q_{j-1}, 1-q_j, q_{j+1},..,q_n)$, where $j = \arg\max_B |g_i|$.
6. Repeat Steps 2-5 for $N_c(\eta)$ random vertices, where η is the required probability for the reaching a global maximum, typically $\eta = 0.95$. The value of $N_c(\eta)$ can be found using results of [15] $N_c(\eta) \approx (1 + p/m)\ln(1-\eta)$.

7. Select the maximal value $E' = \arg \max_{i=1,...,N_c} \psi(E_i)$ among $N_c(\eta)$ local maxima. Following [15] the algorithm of local maximum search converges after no more than me^{γ} steps 2-5, where $\gamma = 0.577215...$ is the Euler constant. The average number of steps is $C(m) = \int_0^1 \exp\left(\sum_{i=1}^{m} \frac{\tau^i}{i}\right) d\tau$.

4. Experiments

Let us consider for example the well-known task of automatic edge detection. The purpose is to detect edges of houses and ignore edges of vegetation using aerial photos. At the first stage of image processing the convolution map $M(x,y) = \max_{\varphi,L} \left| I(x,y) * C(\varphi, L, x', y') \right|$ is obtained, where C is Canny's function [4] of scale of edge L, rotated to angle φ. Calculation of $M(x,y)$ is done using fast Fourier transform. Threshold of $M(x,y) \geq M_0$ is the first parameter. Other parameters can be extracted by applying mask $C(\varphi, L) = \arg\max M(x,y)$ to the image $I(x,y)$ in point (x,y). The second parameter is minimal contrast $C_0 = \max_{x,y \in C} I(x,y) - \min_{x,y \in C} I(x,y)$, the third is minimal median value $m_0 = \max\left(\operatorname{med}_{C(x,y)>0} I(x,y), \operatorname{med}_{C(x,y)<0} I(x,y) \right)$. The next three parameters are minimal common standard deviation $\sigma_0 = D[I(x,y) | (x,y) \in C]$, maximal one-side standard deviation $\sigma_m = \min(D[I(x,y) | C(x,y) > 0], D[I(x,y) | C(x,y) < 0])$ and relative Student relation $t = \left| E[I(x,y) | C(x,y) > 0] - E[I(x,y) | C(x,y) < 0] \right| / \sigma_m$, where $E[x]$ and $D[x]$ are expectation and standard deviation of variable x. Using this set of six parameters $n = 7$, $m = 142$ positive and $p = 180$ negative points edge image of Fig. 1 was received.

5. Conclusion

The further development of the maximization method of functions given on the vertices of a n-dimensional single hyper-cube [15] opens the way to decision making in a wide range of well-known pattern recognition tasks such as prototype feature selection, factor analysis, threshold estimation and so on. The only problem to be solved in this way is to obtain a correct expression for Boolean functions and their derivatives such as (15), (17), (18). Theoretically, following theorem (10) it always possible but practically the derivation of these expressions is a special and sometimes complicated task that requires deeply understanding of the being considered problem.

Fig. 1. The result of house edge detection using optimal threshold selection

Acknowledgment

The authors would like to thank Leah Bar for her help in providing practical information.

References

1. Antamoshkin, A. A., Saraev, V. M.: On Definition of Informative Subsystem of Signs in the Pattern Recognition Problem. Computers and Artificial Intell. 4 (1985) 245-252
2. Beveridge J. R., Riserman E. M.: How Easy Matching 2D Line Models Using Local Search? IEEE Trans. on Pattern Anal. and Machine Intell. 6(1997) 564-579
3. Decaestecker C.: Finding Prototypes for Nearest Neighbor Classification by Means of Gradient Descent and Deterministic Annealing. Pattern Recognition 30(1997) 281-288

4. Demigny D., Kamle T.: A Discrete Expression of Canny's Criteria for Step Edge Detector Performances Evaluation. IEEE Trans. on Pattern Anal. and Machine Intell. 19(1977) 1199-1211
5. Dennis J. E. Jr., Schnabel R. B.: Numerical Methods for Unconstrained Optimization and Nonlinear Equations. Prentice-Hall Inc., New Jersey (1983)
6. Devijer P. A., Kittler J.: Pattern Recognition: A Statistical Approach. Prentice-Hall Inc., Englewood Cliffs, N. J. (1982)
7. Foroutan I., Sklansky J.: Feature Selection for Automatic Classification of Non-Gaussian Data. IEEE Trans. on System Man and Cybernetics 17(1987) 187-198
8. Gates G. W.: The Reduced Nearest Neighbor Rule. Trans. on Inf. Theory 18(1972) 431-433
9. Hart P. E.: The Condensed Nearest Neighbor Rule. Trans on Inf. Theory 14(1968) 515-516
10. Jain A., Zongker D.: Feature Selection: Evaluation, Application and Small Sample Performance. IEEE Trans. on Pattern Anal. and Mach. Intel. 19(1997) 153-158
11. Kim J., Yu J. R., Kim S. H.: Learning of Prototypes and Decision Boundaries for a Verification Problem. Pattern Recognition Lett. 17(1996) 691-697
12. Kittler J.: Feature Set Search Algorithms. In Chen C. H. (ed.): Pattern Recognition and Signal Processing. Sijthoff and Noordhoff, Alphen van der Rijn, The Nethrlands (1978) 41-60
13. Kuncheva L. I.: Editing For the k-Nearest Neighbors Rule by a Genetic Algorithm. Pattern Recognition Lett. 16(1995) 809-814
14. Lipowezky U.: Tree-Plantation Decipherment of Panchromatic Aerial Photo Images Using Supervised Template Matching. In Proc. of 9-th. Mediterranean Electromechanical Conf. Melecon'98, vol. 1 Tel-Aviv (1998) 48-52
15. Lipowezky U.: Selection of the Optimal Prototype Subset for 1-NN Classification. Pattern Recognition Lett. 19(1998) 907-918
16. Sanchez J. S., Fla F., Ferry F. J.: Prototype Selection for the Nearest Neighbor Rule Through Proximity Graph. Pattern Recognition Lett. 18(1997) 507-513
17. Strela V., Heller P. N., Strang G., Topiwala P., Heil C.: The Application of Multi-wavelet Filterbanks for Image Processing. IEEE Trans. on Image Processing 8(1999) 548-563
18. Trier Ø. D., Jain A. K.: Goal-Directed Evaluation of Binarization Methods. IEEE Trans. on Pattern Anal. and Mach. Intel. 17(1995) 1191-1201
19. Vapnik V. N.: Estimation of Dependencies Based on Empirical Data. Springer-Verlag, Berlin Heidelberg (1982)
20. Zadeh L. A.: Fuzzy Sets as a Basis for a Theory of Possibility. Fuzzy Sets Syst. 1(1978) 3-28