Wrocław University of Technology

Multiple classifier system with radial basis weight function

Konrad Jackowski

HAIS 2010

Agenda

- Model of Radial Basis Multiple Classifier System (*RBMCS*)
- *RBMCS* Learning algorithm
- Experiments
- Conclusion

General classification model

Object

Classifier
Ψ

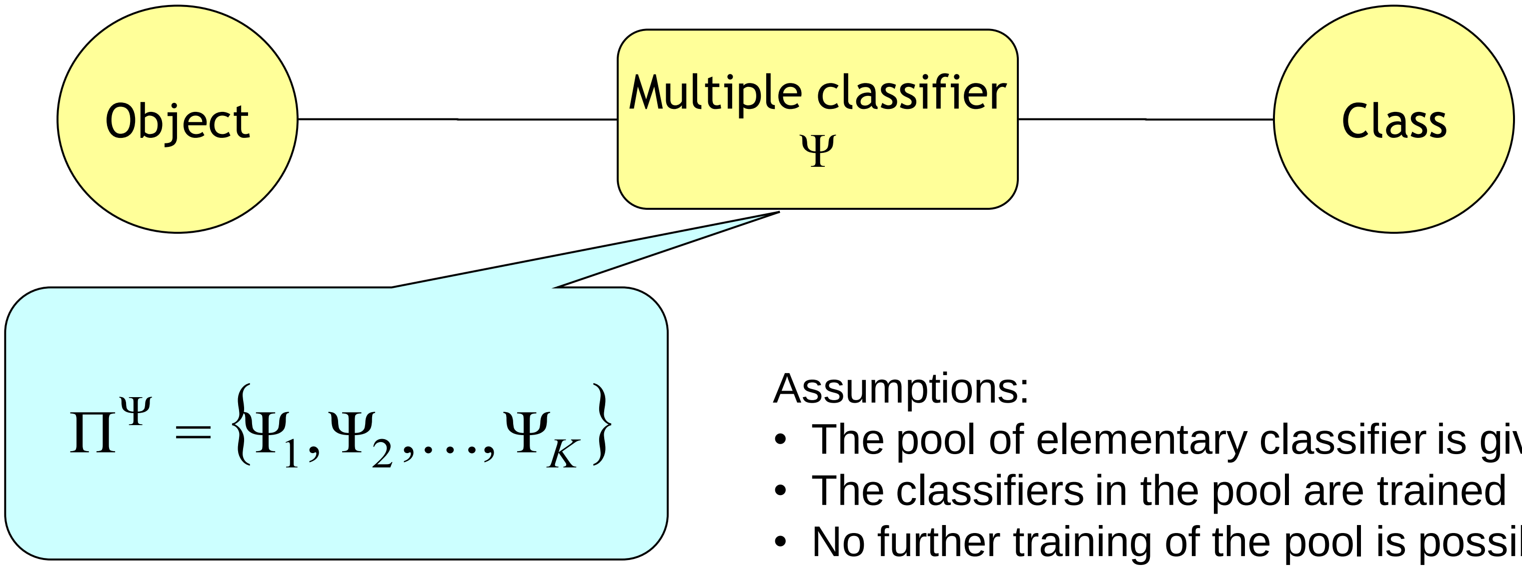
Class

$$x = \left[x^{(1)}, \ldots, x^{(d)}\right]$$

$$i \in \boldsymbol{M} = \{1, \ldots, M\}$$

$$\Psi(x) = i$$

Multiple Classifiers Systems
Fusion of elementary classifiers

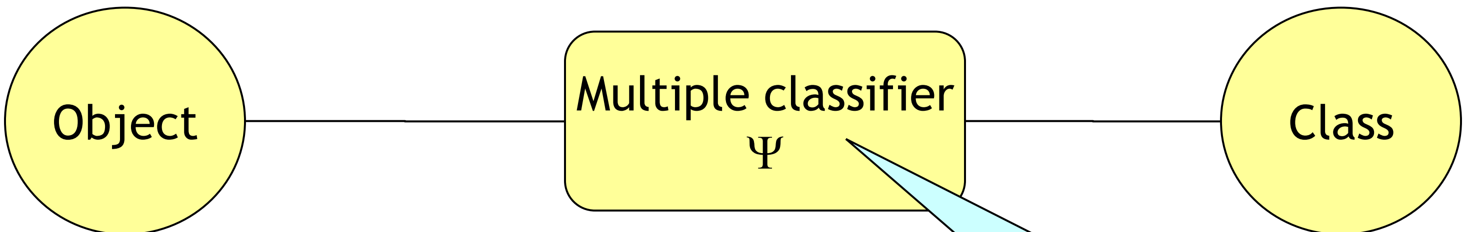


$$\Pi^{\Psi} = \{\Psi_1, \Psi_2, \ldots, \Psi_K\}$$

Assumptions:
- The pool of elementary classifier is given
- The classifiers in the pool are trained
- No further training of the pool is possible

The goal
- To exploit knowledge of the classifiers in the pool
- To recognize local competence of the classifiers
- To design multiple classifier not worst than the best one out of the pool

Multiple Classifiers Systems
Fusion of elementary classifiers



$$\Psi = \arg\max_{i \in \boldsymbol{M}} \sum_{k=1}^{K} \delta\left(\Psi_k(x), i\right)$$

$$\delta(l,i) = \begin{cases} 0 & \Leftrightarrow l \neq i \\ 1 & \Leftrightarrow l = i \end{cases}$$

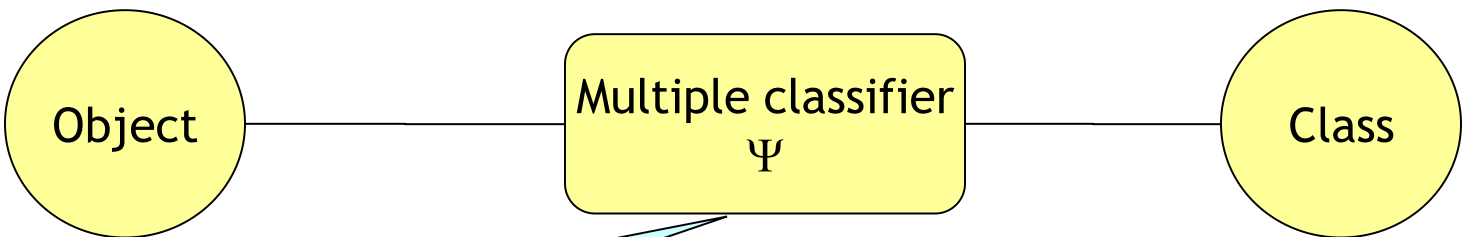
Fusion of classifiers' responses

Majority voting

+ no MCS training

- neglecting of the support level given to the decision

Multiple Classifiers Systems
Fusion of elementary classifiers



$$\Pi^{\Psi} = \{\Psi_1, \Psi_2, \ldots, \Psi_K\}$$

$F_k(1, x)$

$\vdots$

$F_k(M, x)$

$$\Psi_k(x) = \arg\max_{i=1}^{M} F_k(i, x)$$

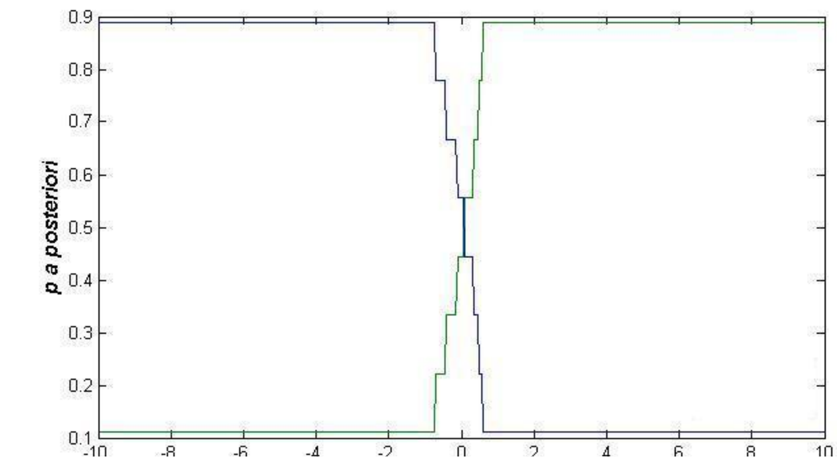
$F_k(i, x)$ - discriminating function of *k-th* elementary classifier

- Support given to *i*-th class
- Posterior probability estimator of *i*-th class

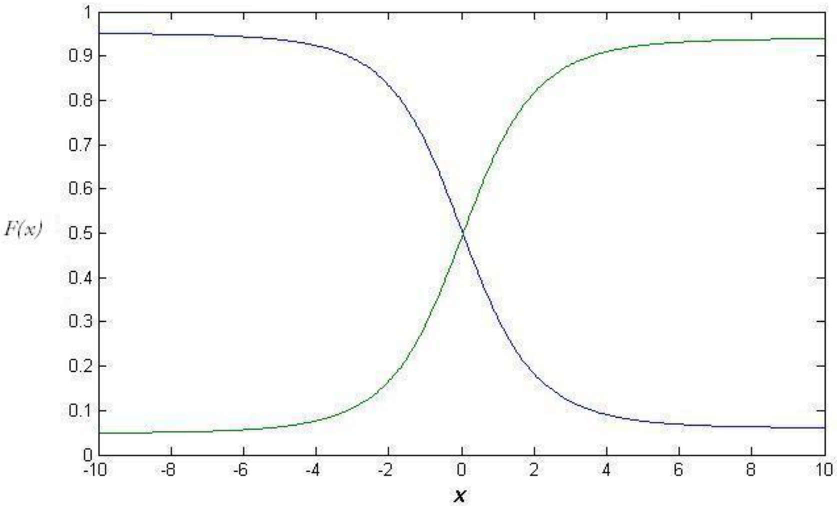
Multiple Classifiers Systems
Fusion of elementary classifiers

Exemplary discriminating function

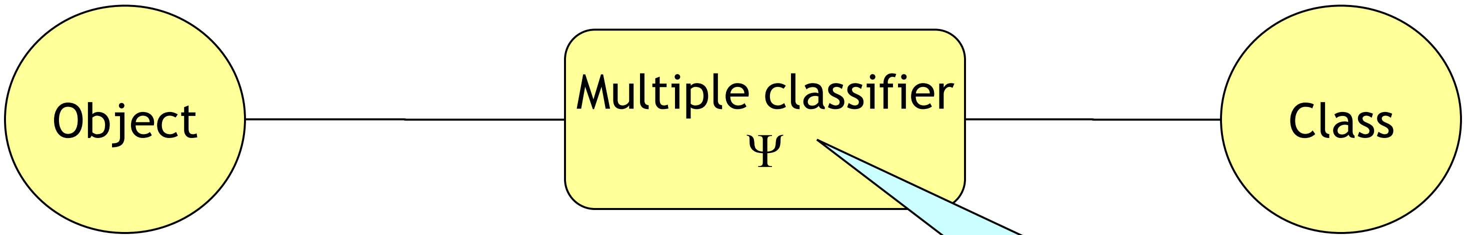
- Posterior probability estimator of k-NN classifier



- Sigmoidal transition function of nonlinear neuron



Multiple Classifiers Systems
Fusion of elementary classifiers



Fusion of discriminating functions

+ taking into consideration value of support given to the decision

+ adjustment classifier's contribution to final decision depending on their competence

$$\Psi(x) = \arg\max_{i=1}^{M} F(i, x)$$

$$F(i, x) = \sum_{k=1}^{K} \varphi_k \; F_k(i, x)$$

Multiple Classifiers Systems
Evaluation of a classifier competence

Object

$$\Psi(x) = \arg\max_{i=1}^{M} \sum_{k=1}^{K} \varphi_k(x)\, F_k(i, x)$$

Class

φ_k Competence coefficient/function of the *k-th* classifier - assumption

- Positive real value
- Straight proportional to respective classifier competence
- <u>Depends on value of *x*</u>

RB MCS
Multiple classifier with radial basis function

Object

$$\Psi(x) = \arg\max_{i=1}^{M} \sum_{k=1}^{K} \varphi_k(x)\, F_k(i, x)$$

Class

φ_k Competence coefficient/function of the *k-th* classifier - assumption

- Positive real value
- Straight proportional to respective classifier competence
- Depends on value of x
  - Reach its maximum at one point (representation point)
  - The longer the distance from the representation point the smaller value returned

$$\varphi_k(x) = \exp\left(-\beta_k\ d(x, C_k)^2\right)$$

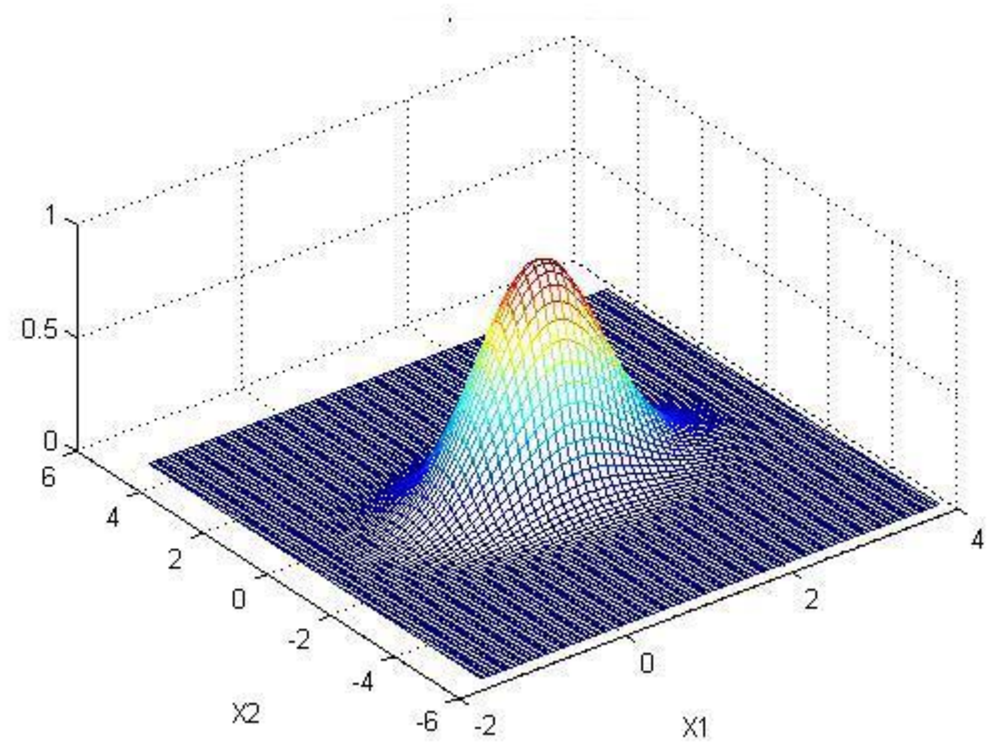
where

$$C_k = \left[c_k^{(1)}, c_k^{(2)}, \ldots, c_k^{(d)}\right]^T \in R^d$$

representation point

RB MCS
Multiple classifier with radial basis function

$$\varphi_k(x) = \exp\left(-\beta_k \; d(x, C_k)^2\right)$$



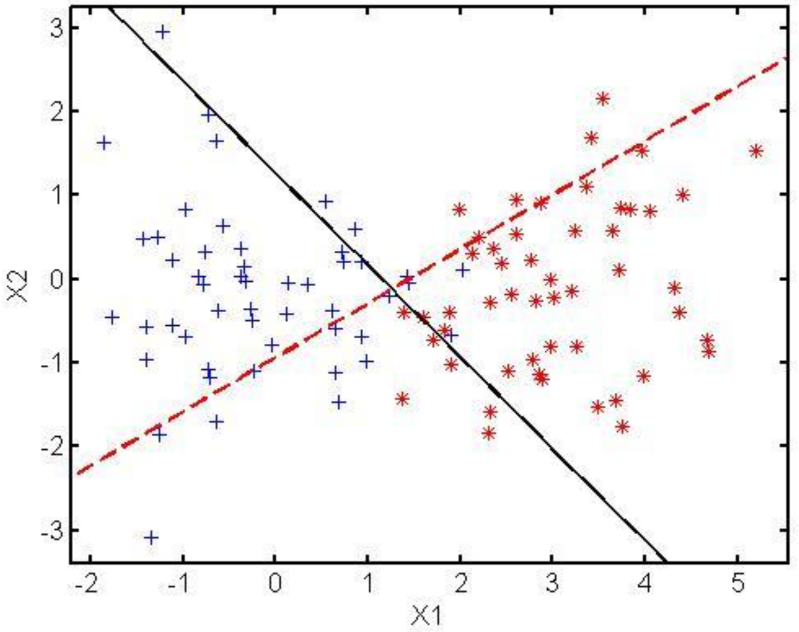
RB MCS
Example

Pattern recognition task

- Number of classes – 2
- Number of attributes – 2
- Prior probabilities of the classes are equal
- Example objects in the classes drawn from normal density function

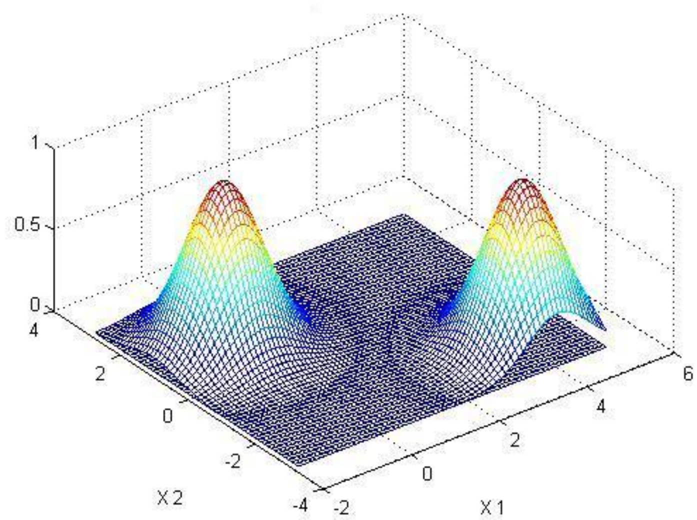
Elementary classifier pool

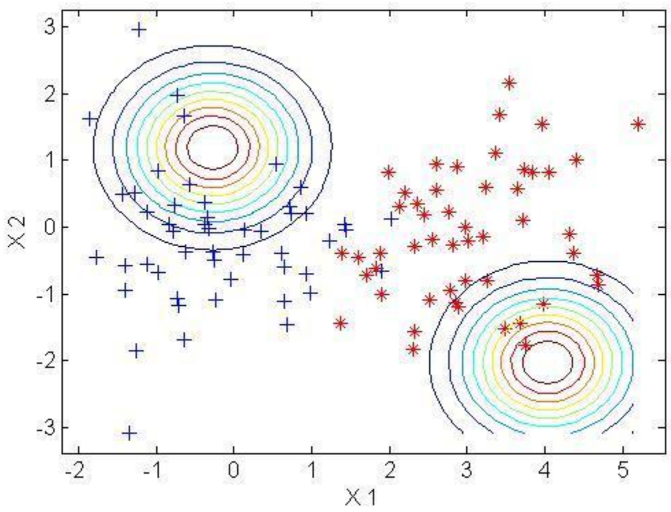
- Classifier 1 (solid line)
  13 % of misclassifications
- Classifier 2 (dot line)
  18 % of misclassifications



RB MCS
Example

	Misclassification rate	Representation points	Beta	Misclassification rate of RB MCS
Classifier 1	0.13	$C_1 = \begin{bmatrix} -0.32 \\ 1.34 \end{bmatrix}$	1.00	0.04
Classifier 2	0.18	$C_2 = \begin{bmatrix} 4.12 \\ -1.89 \end{bmatrix}$	0.98	





RB MCS training

The goal of training process – minimizing of misclassification rate

$$Q_e(\Psi) = \frac{1}{N}\sum_{n=1}^{N} L(\Psi(x_n), j_n)$$

where *L(i,j)* is a loss function

x_n, j_n *n-th* sample drawn from learning set DS along with class index

$$DS = \{(x_1, j_1), (x_2, j_2), \ldots, (x_N, j_N)\}$$

$$Q_e(\Psi) = \frac{1}{N}\sum_{n=1}^{N} L\left(\arg\max_{i=1}^{M}\left(\sum_{k=1}^{K} \exp\left(-\beta_k\ d(x, C_k)^2\right) F_k(i, x)\right), j_n\right)$$

RB MCS training

RB MCS training as compound optimization task

$$Q_e(C,\beta) = \frac{1}{N}\sum_{n=1}^{N} L\left(\arg\max_{i=1}^{M}\left(\sum_{k=1}^{K}\exp\left(-\beta_k\ d(x,C_k)^2\right) F_k(i,x)\right), j_n\right)$$

Optimization of the target function is achieved by modification of

- Set of representation points $C = \{C_1, \ldots, C_K\}$
- Set of parameters $\beta = \{\beta_1, \ldots, \beta_K\}$

Training algorithm of RB MCS

Evolutionary algorithm

- Chromosome model

$$Chromosom = \begin{cases} \{C_1, C_2, \ldots, C_K\} \\ \{\beta_1, \beta_2, \ldots, \beta_K\} \end{cases}$$

- Fitness function

$$\Phi(Chromosom) = 1 - Q_e(Chromosom)$$

- Evolutionary operators
  - mutation – introduces random changes into selected chromosomes
  - crossover – produces child as the result of exchanging data provided by parents

Training algorithm of RB MCS

Preliminary phase	
1	Setting up the parameters (of the model and the algorithm)
2	Generation of the population
3	Assessment of the population
Main phase (loop)	
4	Promoting the elite
5	Mutation
6	Crossover
7	Assessment of the population
8	Drawing members of the population
9	Assessment of the over training
10	Checking out exit condition
Ending phase	
11	Selecting the winner

Experiments

The goal

verification if the RB MCS can effectively exploit local competences of the elementary classifier and create MCS that is characterized by noticeably smaller misclassification rate

Reference point: misclassification rate of

- Elementary classifier in the pool
- MV MCS that make decision according to simple majority voting model

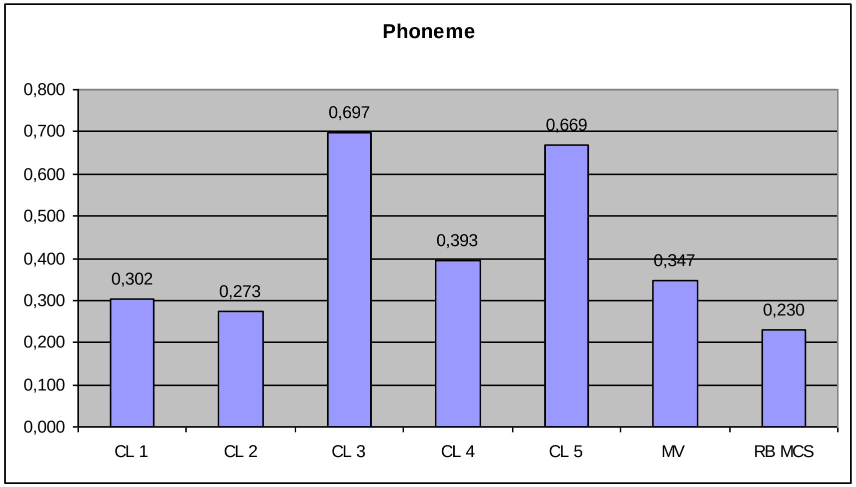
Experiments
Benchmark databases

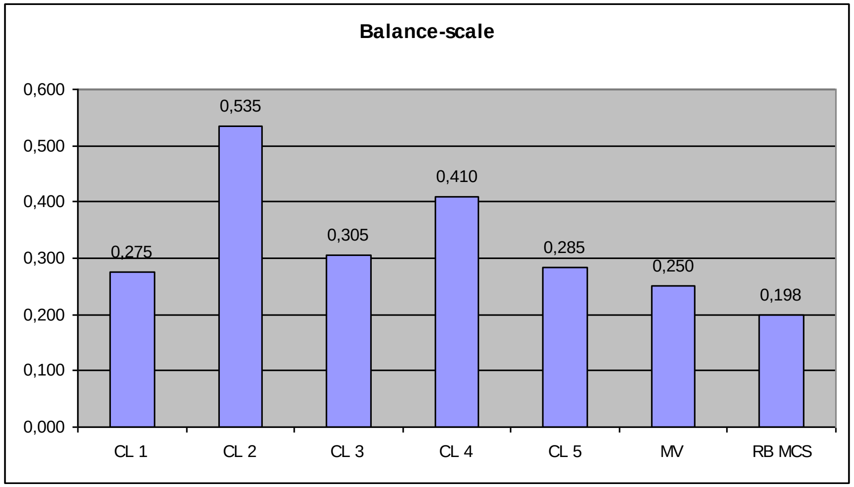
	Experiment 1	Experiment 2	Experiment 3	Experiment 4
Dataset name	Phoneme	Balance-Scale	Cone-Torus	Ecoli
Number of classes	2	3	3	8
Number of attributes	5	4	2	7
Number of instances in CL_LS	1000	104	100	104
Number of instances in RBF_MCS_LS	1000	202	400	103
Number of instances in TS	2404	200	300	129
Number of classifier in pool	5	5	5	5

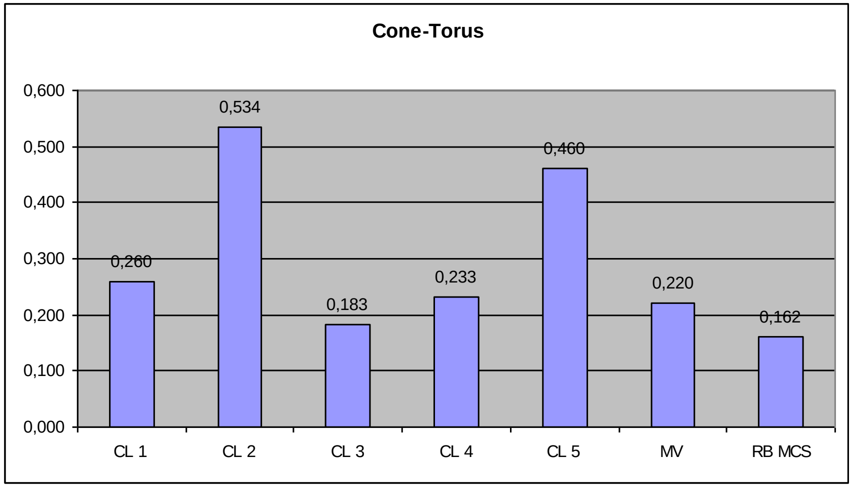
Experiments

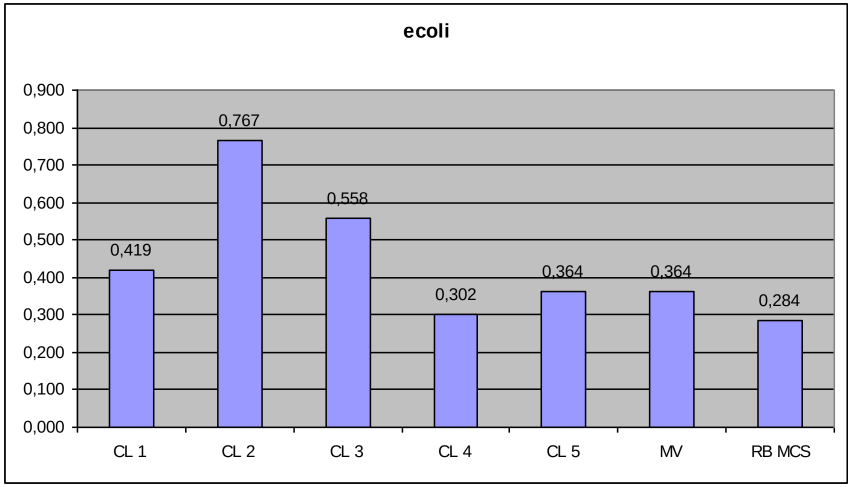
- Experiments were carried on in Matlab with GA-toolbox and PRTools
- Pool of elementary classifier consists of
  - Multilayer sigmoidal neural networks trained according to back propagation algorithm
  - Number of neurons in hidden layer was randomly selected and varied within bounds between 1 and 5
  - Elementary classifiers were under-trained (training process was interrupted after 50 epochs)
- RB MCS training
  - Population of 35 randomly generated individuals
  - The population was processes 30 times
- Each experiment was repeated 50 times

Results of the experiments









Results of the experiments

Observed facts

- RB MCSs achieved always better results than all elementary classifiers in the pool
- RB MCSs achieved always better results than majority voting based MSC
- In experiments 1, 3, 4 majority voting based MCS did not managed to overcome the best of elementary classifier
- Competence function applied allows to effectively evaluate competence of elementary classifier over the feature space
- Training algorithm proposed ensures valid adjustment of RB MCS's parameters that allows effective exploitation of the elementary classifiers

Further works

1. Extending RB MCS – applying competence function that depends on
   - Classifier
   - Value of x
   - Response of the classifier – indicated class
2. Extending representation of the competence function
3. Detailed inspection of the training effectiveness depending on parameters that control training process
4. Validation of the proposes models and training algorithms over larger number of benchmark databases