

Knowledge Discovery and Integration Based on A Novel Neural Network Ensemble Model*

Yong Wang, Hong-Jie Xing

NLPR, Institute of Automation, Chinese Academy of Sciences, Beijing 100080, China
Beijing Graduate School, Chinese Academy of Sciences, Beijing 100080, China
{wangyong,hjxing}@nlpr.ia.ac.cn

Abstract

This article explores the utility of neural network ensembles in knowledge discovery and integration. A novel neural network ensemble model KBNNE (Knowledge-Based Neural Network Ensembles) integrating KDD (Knowledge Discovery in Database) techniques and neural network modeling algorithms by “parallel operations” is proposed. Through balancing the relative importance of knowledge learned by induction and deduction, KBNNE can avoid the knowledge loss and enhance the “transparency” of neural network models. The effectiveness of the proposed model is demonstrated through computer simulations on simple artificial problems and an actual modeling problem.

1. Introduction

The Knowledge Grid is supposed to be a comprehensive computational mechanism that can synthesize knowledge from scientific data through mining and reference methods and enable search engines to make references, answer questions, and draw conclusions from that data [1], but there are still five unsolved issues in its research, one of which is knowledge capture and representation [2, 3]. Artificial Neural Networks (ANNs) have been widely applied in KDD as powerful tools to discover knowledge, but it is hard for one to understand the connectivity among the units in the networks and the nonlinear transformation computed in these units, which is often regarded as “black box” [4]. In many practical modeling problems, incorporating prior domain knowledge in ANNs can overcome this dilemma. Prior domain knowledge consists of information about the data that is already available either from a domain expert or through some knowledge discovery process [5]. But the domain

knowledge obtained from the experts is often constrained to be effective in special cases. Moreover, the credibility of ANNs cannot always be guaranteed with insufficient domain knowledge. An input-output example, either obtained from data sets or created virtually [6, 7], reflects the environment in which the network is embedded. So data may be considered as a special case of domain knowledge, and mining domain knowledge from data with KDD techniques is more generally used than domain experts in neural network modeling process.

On the other hand, ANNs are power tools for KDD. Furthermore, Data preprocessing (or data transformation) and data mining are two basic steps in the KDD process [8]. In soft computing framework, both of their methodologies include fuzzy logic, neural networks, genetic algorithms, rough sets, and so on [9]. So knowledge discovery and neural networks can be integrated together.

Applying ANNs to knowledge discovery is an inductive learning process and incorporating knowledge in ANNs is a deductive learning process. The existed model, e.g. Explanation-Based Neural Network [10], integrates them in “serial operation” manner. However, the latent knowledge may be lost during the transformation from the inductive learning process to the deductive learning one. Thus, this paper proposes a novel model of integrating them by “parallel operations” and balancing the relative importance of knowledge learned through induction and deduction (Fig. 1 shows the difference). The proposed model is also quite different from selective ensembles [11], clustering ensembles [12], and mixture of experts [13], in all of which a finite number of neural networks are trained firstly and then only part of them are integrated according to some principles. This model generates the competent neural networks and integrates them all based on the learned knowledge, so it is called Knowledge-Based Neural Network Ensembles (KBNNE). The KBNNE model is expatiated in section 2. In section 3 experiments with artificial data and actual model-

*This work is supported by Natural Science Foundation of China (#60275025, #60121302).

ing application are performed. Some guidelines for further research are finally outlined in Section 4.

2. KBNNE

Technically speaking, in the models of transferring knowledge across different learning processes involves the problem of knowledge transformation and knowledge loss where there are more than one rule.

Consider a simple example - classifying the hairtail from the tunny. Four facts can be drawn through data statistic analysis as follows:

- Fact 1: on average, annual egg laying amount of the hairtail is about 25,000~35,000;
- Fact 2: on average, annual egg laying amount of the tunny is about 5,000,000;
- Fact 3: on average, annual output of the hairtail is about 1,000,000 tons;
- Fact 4: on average, annual output of the tunny is about 3,000,000 tons.

Then, two rules may be obtained from the above facts:

- Rule 1: on average, annual egg laying amount of the hairtail is smaller than that of the tunny;
- Rule 2: on average, annual output of the hairtail is smaller than that of the tunny.

In the models of integrating induction and deduction by “serial operations”, the two rules may be incorporated in a single neural network. In other words, a single neural network may be trained to represent the two rules. Knowledge is transferred from induction learning process to deduction learning process. Because of the training process of ANNs is hard to understand, no one can guarantee the trained neural network represents the two rules exactly. The real information that the neural network encodes may be that on average the survival probability of the tunny is lower than that of the hairtail, which is a right rule, or that on average the hairtail is heavier than the tunny, which is a wrong rule. So by “serial operations” the deducted rules may be obscure because of knowledge transformation and knowledge loss.

In general cases, data set $D=\{(\mathbf{x}_i, y_i)\}_{i=1}^k$ ($\mathbf{x}_i \in \mathbf{R}^n, y_i \in \mathbf{R}$) is divided into two subsets $D_S=\{(\mathbf{x}_s, y_s)\}_{s=S_1}^{S_k}$ and $D_T=\{(\mathbf{x}_t, y_t)\}_{t=T_1}^{T_k}$, where $D_S \cap D_T=\emptyset, D_S \cup D_T=D$. D_S called support set is used to train different neural networks, which are supposed to generate complete domain knowledge though in most cases it fails. The domain knowledge is often formulated as symbolic rules, such as IF ... THEN rules, which are extracted from the neural networks. The extracted rules

may not be able to express the input-output mappings in a concise way. Let U represent the domain knowledge, $L_S=\{L_{S_1}, L_{S_2} \dots L_{S_M}\}$ represents the neural networks, and $\acute{L}_S=\{\acute{L}_{S_1}, \acute{L}_{S_2} \dots \acute{L}_{S_M}\}$ represents the rules extracted. Then in general cases the following equation exists:

$$\acute{L}_S \subseteq L_S \subseteq U \iff \acute{L}_{S_1} \cup \acute{L}_{S_2} \cup \dots \cup \acute{L}_{S_M} \subseteq L_{S_1} \cup L_{S_2} \cup \dots \cup L_{S_M} \subseteq U. \quad (1)$$

D_T called training set is explained and analyzed in terms of \acute{L}_S and are supposed to refine the domain knowledge for the approximation function by generating new rules. These refined new rules are typically formulated as connectionist rules, such as neural networks. Let $L_T=\{L_{T_1}, L_{T_2}, \dots, L_{T_N}\}$ represents the new rules. Then in general cases the following equation exists:

$$\acute{L}_S \subseteq L_T \subseteq U \iff \acute{L}_{S_1} \cup \acute{L}_{S_2} \cup \dots \cup \acute{L}_{S_M} \subseteq L_{T_1} \cup L_{T_2} \cup \dots \cup L_{T_N} \subseteq U. \quad (2)$$

Because a single neural network can be interpreted as a generalization of a rule, L_T can be represented by a single neural network or by neural network ensembles consist of N different neural networks. It has been proved that the generalization error of neural network ensembles is always smaller than the weighted average of the ensemble errors [14]:

$$E < \bar{E} = \sum_{i=1}^N \omega_i E_i, \quad \sum_{i=1}^N \omega_i = 1, \quad \omega_i \geq 0. \quad (3)$$

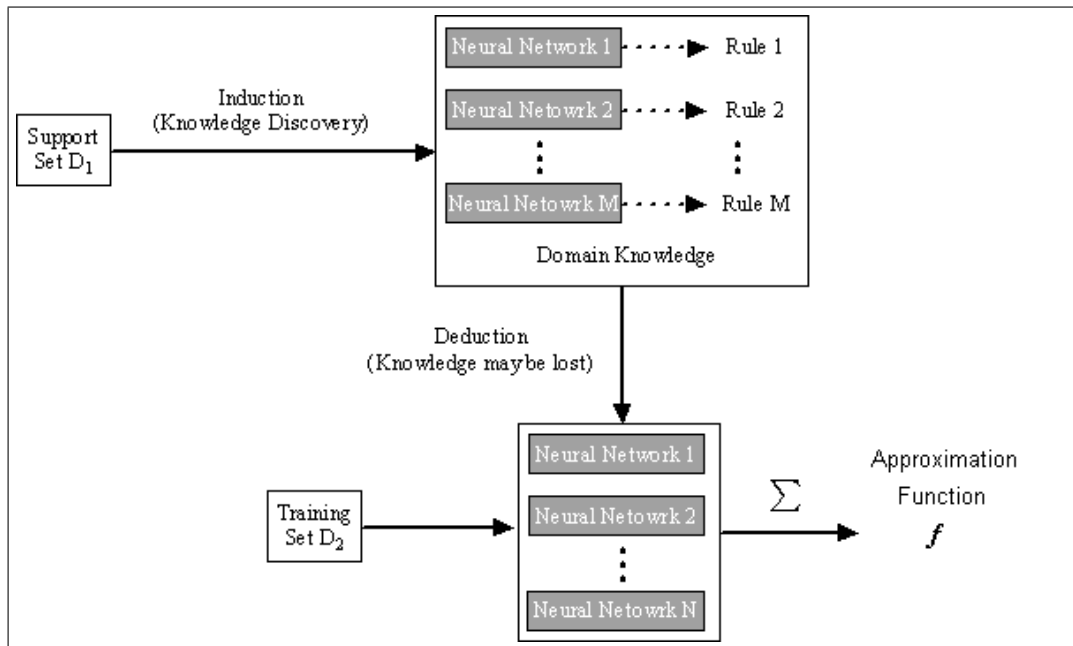
$$E = \int dx p(\mathbf{x}) \left(\sum_{i=1}^N \omega_i L_{T_i} - d(\mathbf{x}) \right)^2. \quad (4)$$

$$E_i = \int dx p(\mathbf{x}) (L_{T_i} - d(\mathbf{x}))^2. \quad (5)$$

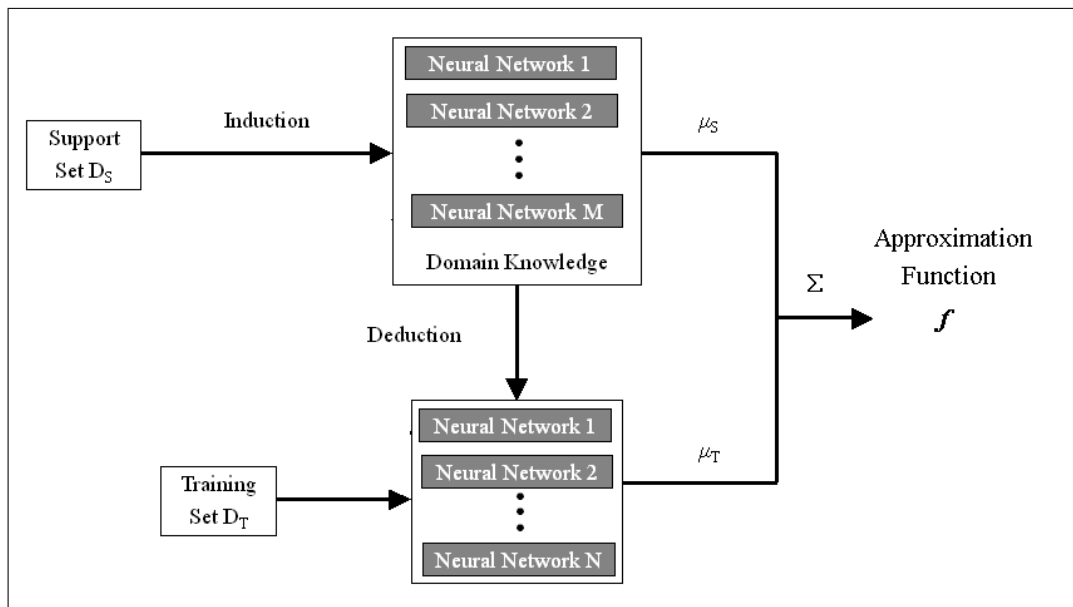
Among the above formulas, E is the generalization error of the ensemble, \bar{E} is the weighted average of the generalization errors of the individual networks, E_i is the generalization error of the network i , and w_i is the weights. So the approximation function f can be calculated by neural network ensembles (Equation 6). Fig.1 (a) shows the process.

$$f = \sum_{i=1}^N \omega_i L_{T_i}. \quad (6)$$

Though L_T are refined rules for \acute{L}_S , which are extracted rules for L_S , no direct relationship between L_T and L_S can be shown from equations (1) and (2). Furthermore, in many



(a) EBNN - serial operations



(b) KBNNE - parallel operation

Figure 1. Two different strategies of integrating induction and deduction.

cases refined rules may not always perform better than original ones. That is, $L_S \cup L_T \neq L_T$. To guarantee the completeness of domain knowledge, both the induced and refined rules should be integrated together. KBNNE integrates all the rules represented by L_S and L_T through “parallel operations” to form a neural network ensembles base on domain knowledge. Furthermore, KBNNE avoids using symbolic rules to transfer knowledge between induction and deduction. According to the knowledge represented by the results of induction, it generates L_T with transformed data. For balancing the relative importance of L_S and L_T , parameters μ_S and μ_T are employed.

So by “parallel operations” the approximation function is calculated by the following neural network ensembles, and equation (6) is a special case of equation (7). The process is illustrated in Fig.1 (b).

$$\begin{aligned} f &= \mu_S L_S + \mu_T L_T \\ &= \mu_S \sum_{i=1}^M \omega_{S_i} L_{S_i} + \mu_T \sum_{i=1}^N \omega_{T_i} L_{T_i} \\ \mu_S + \mu_T &= 1, \quad \mu_S, \mu_T \geq 0. \end{aligned} \quad (7)$$

Since optimizing the combining weights ω_{S_i} and ω_{T_i} can easily lead to the problem of overfitting which simple averaging seems to avoid [15], KBNNE uses simple averaging and without additional information μ_T is set larger than μ_S , for L_T contains more knowledge than L_S in most cases.

$$\begin{aligned} f = \mu_S L_S + \mu_T L_T &= \frac{\mu_S \sum_{i=1}^M L_{S_i} + \mu_T \sum_{i=1}^N L_{T_i}}{M+N} \\ \mu_S + \mu_T &= 1, \quad \mu_T > \mu_S \geq 0. \end{aligned} \quad (8)$$

3 Experiments and Results

This section presents experimental comparisons among different methods, which are performed by the free & open source numerical software package - Scilab [16]. In KBNNE L_S is composed of SVMs classifiers, which can be regarded as a special kind of ANNs and show better generalization than ordinary ANNs classifiers [17], and L_T is composed of ANNs classifiers. RBF kernels are used to form SVMs classifiers, and BP algorithm is used to train ANNs. In order to make comparisons “fair”, we use the same values of classifier parameters (such as the number of hidden neurons, the width of RBF kernel σ) in all of them.

$$f(x, y) = \frac{1}{2\pi\sigma^2} \exp\left\{-\frac{(x - \mu_x)^2 + (y - \mu_y)^2}{2\sigma^2}\right\}. \quad (9)$$

Experiment 1: Similar to the experiments in paper [18], the training and test data have the same prior probability

Table 1. Comparison of error rates for experiment 1

Model	Training Error	Test Error
KBNNE	2.73%	4.18%
SVMs	5.45%	5.91%
NN Ensembles	10.91%	12.45%

0.5 for both classes. The data are generated stochastically as follows:

- Class 1 data (“ \otimes ” in Fig.2) is a mixture of two Gaussian (Equation 9) centered at $(-0.5, 0.3)^T$ and $(0.3, 0.6)^T$, both having the same standard deviation 0.05, stochastic seed for μ_x is -10, while stochastic seed for μ_y is -20. The probability that class 1 data is generated from the first cluster is 10/11 and the probability that class 1 data is generated from the second cluster is 1/11.
- Class 2 data (“ ∇ ” in Fig 2) is a mixture of two Gaussian centered at $(0.5, 0.6)^T$ and $(-0.3, 0.3)^T$, both having the same standard deviation 0.03. Stochastic seed for μ_x is -10, while -20 for μ_y . The probability that class 2 data is generated from the first cluster is 10/11 and the probability that class 2 data is generated from the second cluster is 1/11.

The training data set has 55 samples from Class 1 and 55 samples from Class 2. A test set of 1100 samples is used to estimate the prediction risk (error rate) of the three classification methods under comparison.

Table 1 shows comparison results among KBNNE, SVMs and Neural Network Ensembles. Actually, many other combinations of values of classifier parameters are tried - not shown here due to space limitations. Fig.2 shows actual decision boundaries formed by each method and these results indicate that KBNNE is a very competitive method.

Experiment 2: The training and test data are all coming from the Internet - nonlinedata100 (<http://bach.ece.jhu.edu/pub/gert/svm/incremental>). To testify the robustness of KBNNE, different numbers of training data are chosen, such as 20, 30, 40 and 50, but the numbers of testing data are all 50. Fig.3 show the comparison results among KBNNE, SVMs, neural network ensembles and Shepard interpolation [19], which evidence that on average KBNNE is a stable method for knowledge discovery and integration.

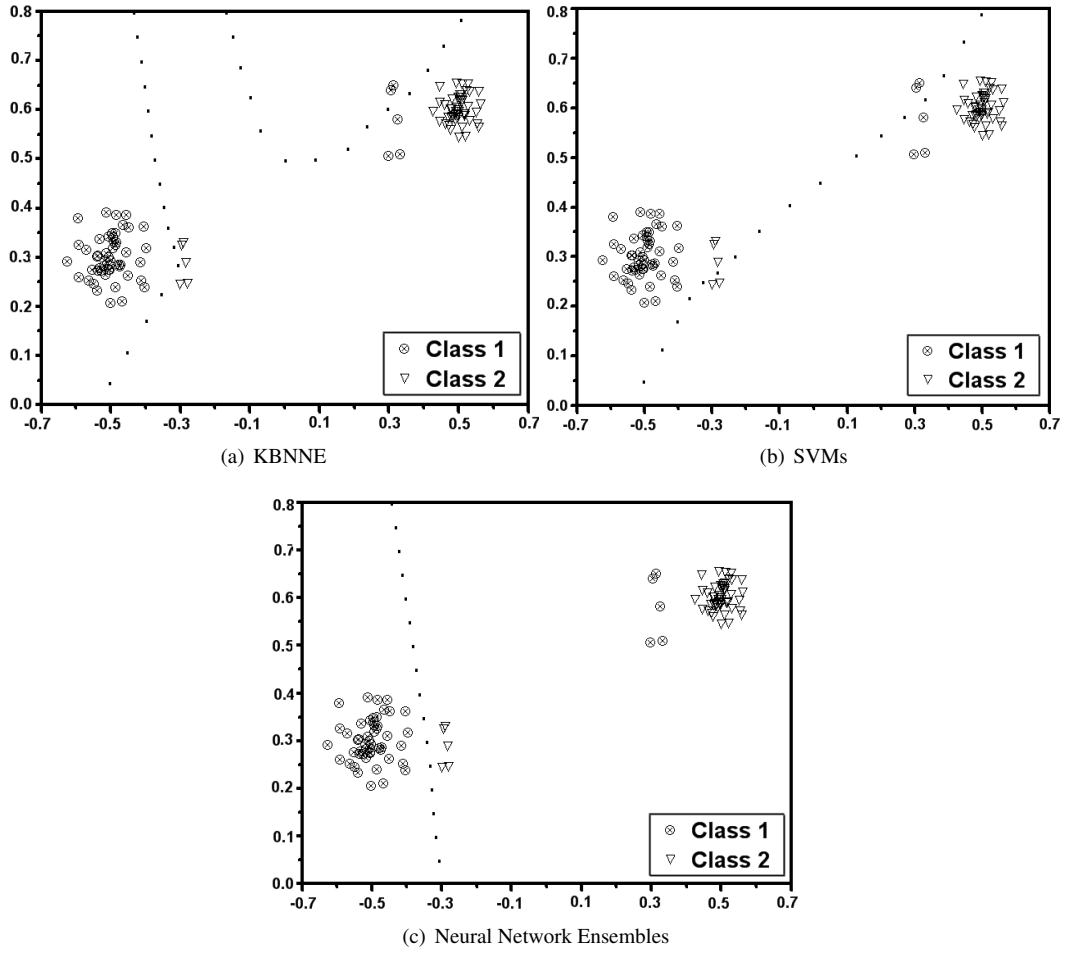


Figure 2. Comparisons of decision boundaries for experiment 1.

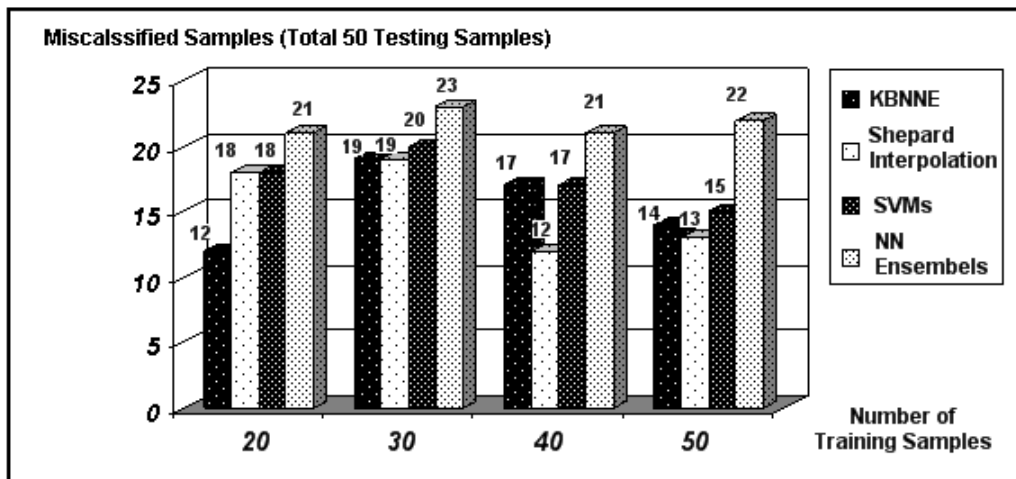


Figure 3. Comparison of misclassified testing data for experiment 2 (total 50 testing samples). Both the experiments can be downloaded from <http://www.ia.ac.cn/personal/yong.wang/>.

4 Conclusions and Perspectives

Knowledge discovery and integration are important but still open problems in the research of KDD and Knowledge Grid. To deal with it, the paper proposes a novel neural network ensemble model called KBNNE. It integrates SVM classifiers and neural network classifiers to form a neural network ensemble by "parallel operations". Through balancing the relative importance of knowledge learned with induction and deduction, the model overcomes the weakness of knowledge loss and enhances the "transparency" of neural network models. Two sets of classification experiments were carried out and verified its efficiency.

A promising property KBNNE is that it can be used to train complex models, which would be difficult to handle in a generative setting. Future theoretical research will be dedicated to incorporate prior knowledge to adjust the scale of neural network ensembles that are eventually fast, better, more general and easier to understand and interpret.

References

- [1] F. Berman, "From TeraGrid to Knowledge Grid", *Communications of the ACM*, 44(11), 2001, pp. 27–28.
- [2] H. Zhuge, "China's E-Science Knowledge Grid Environment", *IEEE Intelligent Systems*, 19(1), 2004, pp. 13–17.
- [3] H. Zhuge, *The Knowledge Grid*, World Scientific Publishing Co., Singapore, 2004.
- [4] A. Michael (Ed.). *The Handbook of Brain Theory and Neural Networks*(2nd edition), MIT Press, Cambridge, Mass, 2003.
- [5] S. S. Anand, D. A. Bell, and J. G. Hughes, "The Role of Domain Knowledge in Data Mining", *Proceedings of the 4th International ACM Conference on Information and Knowledge Management*, 1995, pp. 37–43.
- [6] R. E. Schapire, M. Rochery, M. Rahim, and N. Gupta, "Boosting With Prior Knowledge for Call Classification", *IEEE Trans. on Speech and Audio Processing*, 13(2), 2005, pp. 174–181.
- [7] P. Niyogi, F. Girosi, and T. Poggio, "Incorporating Prior Information in Machine Learning by Creating Virtual Examples", *Proceedings of the IEEE*, 86(11), 1998, pp. 2196–2209.
- [8] Fayyad U. M., G. P. Shapiro, P. Smyth, and R. Uthurusamy R, *Advances in Knowledge Discovery and Data Mining*. American Association for Artificial Intelligence, Menlo Park, CA, USA, 1996.
- [9] S. Mitra, S. K. Pal, and P. Mitra, "Data Mining in Soft Computing Framework: A Survey", *IEEE Trans. on Neural Networks*, 13(1), 2002, pp. 3–14.
- [10] S. Thrun, *Explanation - Based Neural Network Learning: A Lifelong Learning Approach*. Kluwer Academic, Boston, 1996.
- [11] Z. H. Zhou, J. X. Wu, and W. Tang, "Ensemble Neural Networks: Many Could Be Better Than All", *Artificial Intelligence*, 137(1-2), 2002, pp. 239–263.
- [12] B. Bakker, T. Heskes, "Clustering Ensembles of Neural Network Models", *Neural Networks*, 16, 2003, pp. 261–269.
- [13] S. Gutta, J. R. J. Huang, P. Jonathon, and H. Wechsler, "Mixture of Experts for Classification of Gender, Ethnic Origin, and Pose of Human Faces", *IEEE Trans. on Neural Networks*, 11(4), 2000, pp. 948–960.
- [14] A. Krogh, J. Vedelsby, "Neural network ensembles cross validation, and active learning", In: *Tesauro G., Touretzky D., Leen T. (Ed.) Advances in Neural Information Processing Systems*, MA: MIT Press, Cambridge, 7(NIPS-7), 1995, pp. 231–238.
- [15] D. Opitz, J. Shavlik, "Actively searching for an effective neural network ensemble", *Connection Science*, 8(3-4), 1996, pp. 337–353.
- [16] <http://scilabsoft.inria.fr/>
- [17] V. N. Vapnik, *Statistical Learning Theory*, John Wiley & Sons, New York, 1998.
- [18] Y. Q. Ma, V. Cherkassky, "Multiple Model Classification Using SVM-based Approach", *Proceedings of the International Joint Conference on Neural Networks*, 2, 2003, pp. 20–24.
- [19] T. Wu, H. G. He, and M. K. He, "Interpolation Based Kernel Function's Construction", *Chinese Journal of Computers*, 26(8), 2003, pp. 990–996 (in Chinese).