WSC17 Online version
2012 Online Conference on
Soft Computing in Industrial Applications
Anywhere on Earth, December 10-21, 2012

A New RBFNDDA-KNN Network and Its Application to Medical Pattern Classification

Shing Chiang Tan[1*], Chee Peng Lim[2], Robert F. Harrison[3], R. Lee Kennedy[4]

[1]Faculty of Information Science and Technology, Multimedia University, Malaysia
*Email(corresponding author): sctan@mmu.edu.my
[2]Centre for Intelligent Systems Research, Deakin University, Australia
[3]Department of Automatic Control and Systems Engineering, University of Sheffield, UK
[4]School of Medicine, Deakin University, Australia

Abstract. In this paper, a new variant of the Radial Basis Function Network with Dynamic Decay Adjust (RBFNDDA) is introduced for undertaking pattern classification problems with noisy data. The RBFNDDA network is integrated with the *k*-nearest neighbours algorithm to form the proposed RBFNDDA-KNN model. Given a set of labelled data samples, the RBFNDDA network undergoes a constructive learning algorithm that exhibits a greedy insertion behaviour. As a result, many prototypes (hidden neurons) that represent small (with respect to a threshold) clusters of labelled data are introduced in the hidden layer. This results in a large network size. Such small prototypes can be caused by noisy data, or they can be valid representatives of small clusters of labelled data. The KNN algorithm is used to identify small prototypes that exist in the vicinity (with respect to a distance metric) of the majority of large prototypes from different classes. These small prototypes are treated as noise, and are, therefore, pruned from the network. To evaluate the effectiveness of RBFNDDA-KNN, a series of experiments using pattern classification problems in the medical domain is conducted. Benchmark and real medical data sets are experimented, and the results are compared, analysed, and discussed. The outcomes show that RBFNDDA-KNN is able to learn information with a compact network structure and to produce fast and accurate classification results.

Keywords: Radial Basis Function Neural Network, Nearest Neighbour, Pattern Classification

1 Introduction

During the past few decades, many different soft-computing techniques such as artificial neural networks [1-3], decision trees [4-6], and fuzzy system [7-9] have been successfully applied as intelligent data processing tools to various domains, e.g., biomedicine [1], [4], [7], manufacturing processes [2], [5], [8], and power systems [3], [6], [9]. This line of research continues to attract the attention of the soft-computing community to further develop more efficient models for learning information from databases and for dealing with various complex problems encountered in different

adfa, p. 1, 2011.

domains. Learning from databases is a challenging task. Normally, real-world data samples are complicated to analyse, as they usually contain noise. Indeed, noisy data denote a group of unrepresentative samples in the database [10]. On the other hand, the database can also contain some small (with respect to a threshold) clusters of labelled data that contain important information for supervised learning. As an example, in medical diagnosis, some patients may develop some distinctive reactions or symptoms pertaining to a treatment, e.g. a severe chest pain [11] can be caused by a novel occurrence of drug allergy, and it should be captured as an important case for further study. Hence, it is useful to have a learning model that is able to first identify small prototypes (i.e. representatives of small clusters of labelled data) in the vicinity (with respect to a distance metric) of large ones. Then, the learning model needs to estimate whether these small prototypes are noise, or they are valid representatives of small clusters of labelled data (which carry important information) whereby these small representatives belong to the same class as the majority of the large prototypes in the vicinity (with respect to a distance metric). The former needs to be pruned, while the latter needs to be preserved in the network. This is the main motivation of the current study.

The main purpose of this study is to devise a new variant of the Radial Basis Function Network with Dynamic Decay Adjust (RBFNDDA) [12] model for pattern classification with noisy data. Specifically, the *k*-nearest neighbour algorithm [13-14] is integrated with RBFNDDA to form the proposed RBFNDDA-KNN model to achieve this purpose. The original RBFNDDA network has a fast constructive supervised learning process, and is able to produce good classification rates. One of the salient features of RBFNDDA is that its structure grows incrementally during the training process, whereby new hidden neurons are created to include new training instances in form of prototypes. However, RBFNDDA is sensitive to noise. Its constructive learning algorithm manifests a greedy insertion behaviour, which can lead to an oversized network that contains a lot of small prototypes (hidden neurons). The proposed RBFNDDA-KNN network is designed with the aim to prune small noisy prototypes while preserving other small useful prototypes into its network structure. To demonstrate the effectiveness of the proposed RBFNDDA-KNN network, case studies in the medical pattern classification domain are conducted. Both benchmark and real medical data sets are used in the experiments. The results are analysed, compared, and discussed.

The organization of this paper is as follows. In Section 2, the background of RBFNDDA-based networks is first presented. Then, the proposed RBFNDDA-KNN network is explained in detail. In Section 3, an empirical study using a number of benchmark data sets from the UCI machine-learning repository [15] and a real medical data set from a UK hospital is described. The results are compared and analysed. In Section 4, a summary pertaining to the current study and suggestions for further work is included.

2 Classification Methods

In this section, the dynamics of RBFNDDA and KNN are explained. This is followed by a description of the proposed RBFNDDA-KNN network for selectively pruning small prototypes for undertaking pattern classification problems.

2.1 Radial Basis Function Neural Network with Dynamic Decay Adjustment (RBFNDDA)

RBFNDDA inherits the salient features of the Probabilistic Neural Network (PNN) [16] and the Restricted Coulomb Energy Network (RCEN) [17-18]. It extracts the probabilistic characteristics of a static PNN and the ability of adjusting the prototype (hidden neuron) width of the RCEN, and embeds them into a network structure that can grow incrementally during training. The RBFNDDA network is a three-layered architecture, as shown in Fig. 1.

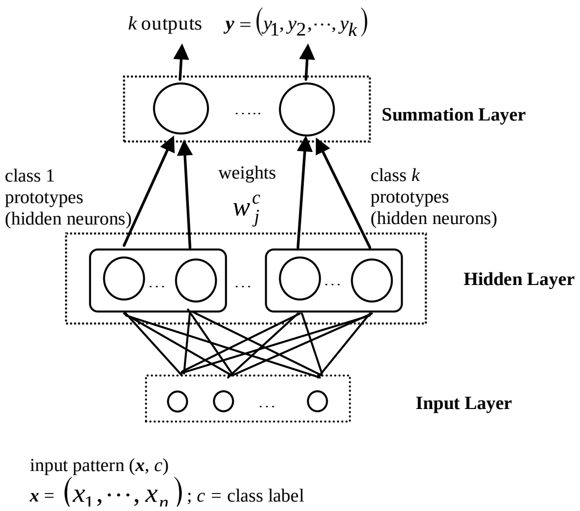

Fig. 1. The RBFNDDA architecture

RBFNDDA requires only two user-defined thresholds (i.e., θ^+ and θ^-) for adjusting the prototype width during the network training phase. These two user-defined thresholds are shown in Fig. 2. The thresholds separate a prototype from its neighbours (prototypes) of different classes. For each training pattern, threshold θ^+ sets the minimum correct-classification probability for the correct class, whereas threshold θ^- sets the highest probability allowable for an incorrect class. This corresponds to an area of conflict whereby neither matching nor conflicting training patterns can reside, as shown in Fig. 2. In other words, thresholds θ^+ and θ^- control the size of the overlapping region of each prototype in RBFNDDA. According to [10], [12], $\theta^{+} = 0.40$ and $\theta^{-} = 0.20$ are suitable to be used as the default settings, and they are good enough to give satisfactory classification results.

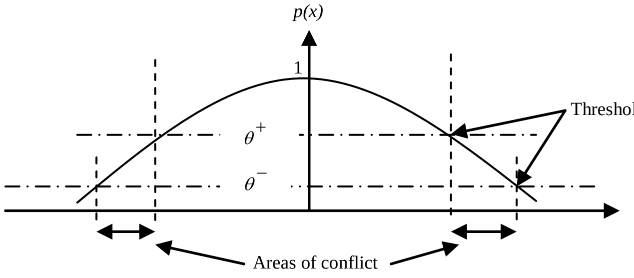

Fig. 2. The two user-defined thresholds of a radial basis hidden node

RBFNDDA employs the DDA algorithm to construct its network structure. During the training phase, additional prototypes are inserted into the network in order to learn new information from incoming data. All committed prototypes are accumulated along the course of its training session. Fig. 3 illustrates the learning procedure of RBFNDDA in a single epoch.

The RBFNDDA training procedure can be summarised as follows. First, all the weights of the prototypes are initialized to zero (in order to prevent accumulation of duplicated information pertaining to the training patterns). On presentation of a new training sample, if the sample is correctly classified by a few existing prototypes, the weight of the largest prototype is increased. However, if the training sample is incorrectly classified, a new prototype is introduced to include the sample in the network. In this case, the new prototype has the sample as its centre, with its weight set to one. Its initial width is then set in such a way to avoid overlapping with the neighbouring prototypes of different classes. The next step is to perform width shrinking for all prototypes of different classes, if their activations caused by the training sample are above θ^{-}. Details of the RBFNDDA can be found in [10], [12].

2.2 The *k*-Nearest Neighbour (KNN) Algorithm

The KNN algorithm [13-14] is a simple instance-based learning method. It classifies a new instance (training sample) based on the *k* closest instances in the data space. Normally, a distance metric, e.g., the Euclidean distance, is used to measure the closeness between the new sample and the existing ones. Using a single neighbour classification scheme, the new sample is classified as belonging to the class of the closest sample. If more than one (i.e., *k*-) nearest neighbours from the existing samples is considered for classification, then the new sample is assigned to the majority class of its closest *k* neighbours.

// reset weights
FORALL prototypes p_j^c **DO**
$\quad w_j^c = 0$
ENDFOR
// train one complete epoch
FORALL training patterns $(\boldsymbol{x}, c)$ **DO**:
$\quad$ **IF** $\exists p_j^k : p_j^k(\boldsymbol{x}) \geq \theta^+$ **THEN**
$\quad\quad \boldsymbol{w}_j^c = \boldsymbol{w}_j^c + 1$
$\quad$ **ELSE**
$\quad$ // introduce a new prototype (hidden neuron)
$\quad\quad m_c = m_c + 1$
$\quad\quad w_{m_c}^c = 1$
$\quad\quad z_{m_c}^c = x$
$\quad$ // adapt radii
$$r_{m_c}^c = \min_{\substack{s \neq c \\ 1 \leq j \leq m_s}} \left\{ \sqrt{-\frac{\left\| \mathbf{z}_j^s - \mathbf{z}_{m_c}^c \right\|^2}{\ln \theta^-}} \right\}$$
$\quad$ **END**
$\quad$ // shrink radii of conflicting prototypes (hidden neurons)
$\quad$ **FORALL** $s \neq c,\ 1 \leq i \leq m_s$ **DO**
$$r_i^s = \min \left\{ r_i^s, \sqrt{-\frac{\left\| \mathbf{z}_j^s - \mathbf{z}_{m_c}^c \right\|^2}{\ln \theta^-}} \right\}$$
$\quad$ **END**
END

Fig. 3. The learning algorithm of RBFNDDA in one epoch [10]

2.3 The RBFNDDA-KNN Network

RBFNDDA-KNN is an extension of RBFNDDA for which its learning process encompasses an instance-based KNN algorithm to deal with noisy information. A collection of prototypes is formed in the network after a training epoch of RBFNDDA-KNN. This collection of prototypes captures information from all training samples. Some of the prototypes are committed to cover only a small number of training samples. These small prototypes are caused either by noisy data or valid representatives of small clusters of labelled data. The KNN algorithm helps avoid establishing a large number of small prototypes in the network, especially in the presence of noisy samples. In each training epoch, such small prototypes, p_j^c, $c \in \{1, 2, \cdots, C\}$, $j \in \{1, 2, \cdots, m_c\}$, are identified if their weights are smaller than a threshold value, v.

On the other hand, those prototypes with weights larger than *v*, i.e., p_l^c $c \in \{1,2,\cdots,C\}$, $l \in \{1,2,\cdots,m_c\}$, are deemed useful. When applying the KNN algorithm, let $N_k\left(p_l^c\right)$ be a set of *k* centres of p_l^c prototypes that are the closest to each centre of p_j^c measured using the standard Euclidean distance metric. The majority class of $N_k\left(p_l^c\right)$ is determined according to

$$s = \arg \max_{c \in \{1,2,\cdots,C\}} N_k\left(p_l^c\right) \tag{1}$$

If $c \neq s$, prototype p_j^c is removed from the network because it is considered as a noisy prototype locating close to a group of large prototypes from other classes in the data space. However, if $c = s$, it is retained in the network because both prototypes belong to the same class, except one is a large prototype while another is a small prototype (which may contain some specific and important information) that needs to be preserved. Fig. 4 shows a schematic diagram (a two-dimensional example) that uses the KNN algorithm to identify and remove a small noisy prototype from the RBFNDDA-KNN network.

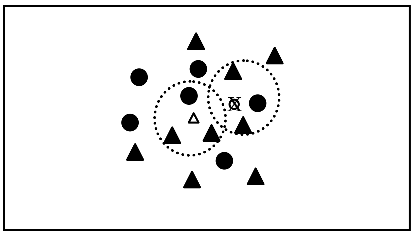

Fig. 4. The use of the KNN algorithm to identify and remove noisy prototypes (hidden neurons). Note that filled elements represent "large" prototypes (larger than *v*) and hollow elements represent "small" prototypes (smaller than *v*) from two classes (C1=circle and C2=triangles). With *k*=3, the small C2 prototype (hollow triangle) is preserved while the small C1 prototype (hollow circle) is removed (marked by X) from the RBFNDDA-KNN network.

3 Experimental Study

The proposed RBFNDDA-KNN network was evaluated with three benchmark and one real medical data sets in this study. The benchmark data sets used were taken from the UCI Machine Learning Repository [15]. The real medical data set comprised Myocardial Infarction (MI) patient records collected from a hospital in the UK. The purpose of the experiment was two-fold: (i) to evaluate the effectiveness of the proposed RBFNDDA-KNN network by comparing its classification performance with RBFNDDA (as reported in [10]) using the benchmark data sets; (ii) to demonstrate the applicability of the proposed RBFNDDA-KNN network using real MI patient records. A number of experimental runs were conducted, and the average results and their 95% confidence intervals were estimated using the bootstrap method [19]. The bootstrapped results were reported mainly because of the ability of the bootstrap method to provide accurate estimates of statistical parameters (e.g. the average results and their confidence intervals) based on a small sample size. These results were then analysed, compared, and discussed.

3.1 Benchmark Medical Problems

Three medical data sets from the UCI machine learning repository were used in this experimental study. They were the *diabetes*, the *cancer* and the *heart* data sets. Their respective number of data samples, number of attributes, number of classes were 768, 8, 2, 699, 9, 2, and 270, 13, 2. The study aimed to compare the classification performances of RBFNDDA-KNN (from this study), RBFNDDA-T (from [10]), and original RBFNDDA (from [10]) based on the same experimental setup as in [10]. Each data set was divided equally into a training set and a test set. For clarity, RBFNDDA-T is a modified version of RBFNDDA that has been enhanced with an online pruning algorithm for removing noise from the network after every training epoch. If a prototype does not contain a sufficient number of training samples (by comparing the number of training samples against a threshold), then it is treated as a noise and is kept in a blacklist. In this regard, the prototype is removed for subsequent training epochs. In [10], both RBFNDADDA and RBFNDDA-T were trained in multiple epochs with $\theta^{+} = 0.40$, $\theta^{-} = 0.20$. In addition, RBFNDDA-T used a threshold setting of *v*=2 [10]. In this study, the RBFNDDA-KNN network was trained repeatedly in multiple epochs with the same setting of $\theta^{+} = 0.40$, $\theta^{-} = 0.20$, and with *k*=10. The experiment was conducted using a personal computer with a configuration of Genuine Intel (R) CPU 2160 at 1.80GHz 1.80GHz, 4GB of memory. The trained network was then evaluated using unseen samples from the test set. The experiment was conducted for eight runs, and the average results in terms of the accuracy rates and number of prototypes were computed. Table 1 shows the classification performances from the RBFNDDA-KNN, RBFNDDA and RBFNDDA-T networks. Note that the classification results in terms of the accuracy rates and number of prototypes of RBNDDDA and RBFNDDA-T are those reported in [10].

Table 1. The classification results of RBFNDDA-KNN, RBFNDDA-T and RBFNDDA from the Diabetes, Cancer and Heart case studies (Acc.-average test accuracy rate; #Prototypes–average number of prototypes (hidden neurons); standard deviations are typed in round brackets).

Data Set	RBFNDDA (from [10])		RBFNDDA-T (from [10])		RBFNDDA-KNN (from this study)	
	Acc.(%)	#Prototypes	Acc.(%)	#Prototypes	Acc.(%)	#Prototypes
Diabetes	74.35(0.97)	288.5(6.1)	73.50(2.38)	65.6(4.2)	71.14(1.92)	161.9(8.5)
Cancer	96.86(0.99)	70.6(13.4)	96.90(0.39)	38.3(4.6)	96.85(0.47)	38.7(5.3)
Heart	79.26(2.83)	83.6(3.3)	79.82(3.54)	32.6(2.1)	79.90(2.45)	39.5(9.5)

The accuracy rates of RBFNDDA-KNN, RBFNDDA and RBFNDDA-T are almost similar, as shown in Table 1. However, RBFNDDA-KNN used approximately 50% fewer number of prototypes than RBFNDDA to encode information from the data sets. By analysing the RBFNDDA-KNN structure, it was noticed that those “small” prototypes, which were located among other prototypes representing the general concept of other class(es) in the data space, could be identified as noise and, therefore, removed from the network. The network size of RBFNDDA-KNN was larger than that of RBFNDDA-T. This was because the former did not simply remove all small prototypes during the training process, while the latter would remove all prototypes smaller than the specified threshold and put them in a blacklist. However, RBFNDDA-KNN had an advantage from the computational viewpoint. While RBFNDDA-T could identify and accumulate a list of unrepresentative samples to prevent the network from absorbing noise, RBFNDDA-KNN, on the other hand, did not need such a blacklist to prune noisy prototypes from its network. This computational advantage of RBFNDDA-KNN was clearly exhibited in the next case study.

3.2 A Real Medical Problem

In this experiment, a data set comprising 500 patient records collected from the Northern General Hospital, Sheffield, UK, was used to evaluate the applicability of the RBFNDDA-KNN network to real-world medical classification problems. The task was to categorise the data samples into to two categories: suspected Myocardial Infarction (MI) and non-MI cases. There were about 31% and 69% MI and non-MI cases, respectively, in the data set. Each data sample comprised a total of 29 features that were found to be significant to MI diagnosis, which included clinical measurements (e.g. ECG readings), physical symptoms (e.g. sweating, vomiting, description of chest pain), and other relevant information (e.g. age, smoker/ex- smoker). The features were coded as binary numbers, with a 1/0 indicating presence/absence of symptoms. In addition, there were two real-valued features for age and duration of pain (normalised between 0 and 1), respectively.

The experimental setup and the network parameter settings were the same as those described in Section 3.1. To better evaluate the stability of the performances, the experiment was repeated for 100 times with different sequences of the training samples. In addition, the Data Retention Rate (DRR) [20], i.e. a useful measure of network compactness, was also computed, as follows.

$$\text{data retention rate} = \frac{\text{number of hidden neurons (prototypes)}}{\text{number of training samples}} \tag{2}$$

Table 2 summarises the bootstrapped results of RBFNDDA, RBFNDDA-T, and RBFNDDA-KNN. The average accuracy rate of RBFNDDA-KNN is slightly higher than those of RBFNDDA-T and RBFNDDA. The number of prototypes of RBFNDDA-KNN and RFNDDA-T are smaller than that of RBFNDDA. The empirical results indicated that RBFNDDA-KNN was able to mitigate the influence of unrepresentative data samples and to produce high classification accuracy. Between

RBFNDDA-KNN and RBFNDDA-T, the former created more number of prototypes than the latter, as shown by the DRR results. However, the computational time of RBFNDDA-KNN was faster than that of RBFNDDA-T, although the former had more than twice the number of prototypes than the latter. As explained in Section 3.1, RBFNDDA-T stored a list of unrepresentative samples to check against noise from the training samples in each training epoch. Hence, additional computational time was required. This problem could become severe when large and noisy training data sets were used. In short, RBFNDDA-KNN could form a compact network structure (as compared with RBFNDDA) and produce fast and accurate classification results (as compared with both RBFNDDA and RBFNDDA-T).

Table 2. The average boostrapped results of RBFNDDA-KNN, RBFNDDA-T and RBFNDDA from the MI case study (the ranges in the square brackets indicate the 95% confidence intervals of the estimated average boostrapped results).

Classifier	Acc. (%)	#Prototypes	DDR	Time(s)
RBFNDDA	69.69 [69.28 70.12]	119.0 [119.3 125.4]	0.48	2.5 [2.4 2.5]
RBFNDDA-T	69.54 [69.15 69.98]	30.9 [29.9 31.8]	0.12	3.4 [3.3 3.5]
RBFNDDA-KNN	70.27 [69.73 70.84]	69.8 [67.3 72.3]	0.28	2.7 [2.6 2.7]

4 Summary

In this paper, a new variant of the RBFNDDA network has been proposed for undertaking pattern classification problems with noisy data. The proposed network, known as RBFNDDA-KNN, is based on integration between RBFNDDA and the KNN algorithm. The proposed RBFNDDA-KNN network is able to identify small noisy prototypes and preserve small useful prototypes that represent small clusters of important labelled data. A series of experiments using both benchmark and real medical data sets has been conducted to evaluate the effectiveness of RBFNDDA-KNN. The results have shown that RBFNDDA-KNN is able to learn and encode information from the database using a smaller number of prototypes than RBFNDDA, and with improved classification results. As compared with RBFNDDA-T, RBFNDDA-KNN has exhibited a larger network structure. However, RBFNDDA-T does not need to keep a list of unrepresentative samples (as what RBFNDDA-T does to identify noisy data) during the training process. As such, RBFNDDA-KNN is able to produce fast and accurate results in the presence of noisy samples in the database.

For future work, additional experiments to ascertain the effectiveness of the proposed RBFNDDA-KNN network in various real-world classification domains will be conducted. In addition, rule extraction from RBFNDDA-KNN can be carried out so that it can provide explanation for its predicted outcomes.

References

1. Lee, J., Steele, C.M., Chau, T.: Classification of Healthy and Abnormal Swallows Based on Accelerometry and Nasal Airflow Signals. Artificial Intelligence in Medicine 52, 17-25 (2011)
2. Yu, J.-b., Xi, L.-f.: A Neural Network Ensemble-Based Model for On-Line Monitoring and Diagnosis of Out-of-Control Signals in Multivariate Manufacturing Processes. Expert Systems with Applications 36, 909–921 (2009)
3. Barakat, M., Druaux, F., Lefebvre, D., Khalil, M., Mustapha, O.: Self Adaptive Growing Neural Network Classifier for Faults Detection and Diagnosis. Neurocomputing 74, 3865-3876 (2011)
4. Son, C.-S., Kim, Y.-N., Kim, H.-S., Park, H.-S., Kim, M.-S.: Decision-Making Model for Early Diagnosis of Congestive Heart Failure Using Rough Set and Decision Tree Approaches. Journal of Biomedical Informatics 45, 999-1008 (2012)
5. Ciflikli, C., Kahya-Özyirmidokuz, E.: Implementing a Data Mining Solution for Enhancing Carpet Manufacturing Productivity. Knowledge-Based Systems 23, 783–788 (2010)
6. Upendar, J., Gupta, C.P., Singh, G.K.: Statistical Decision-Tree Based Fault Classification Scheme for Protection of Power Transmission Lines. Electrical Power and Energy Systems 36, 1-12 (2012)
7. Alayón, S., Robertson, R., Warfield, S.K., Ruiz-Alzola, J.: A Fuzzy System for Helping Medical Diagnosis of Malformations of Cortical Development. Journal of Biomedical Informatics 40, 221-235 (2007)
8. Piltan, M., Mehmanchi, E., Ghaderi, S.F.: Proposing a Decision-Making Model Using Analytical Hierarchy Process and Fuzzy Expert System for Prioritizing Industries in Installation of Combined Heat and Power Systems. Expert Systems with Applications 39, 1124-1133 (2012)
9. Kazemi, M.V., Moradi, M., Kazemi, R.V.: Minimization of Powers Ripple of Direct Power Controlled DFIG by Fuzzy Controller and Improved Discrete Space Vector Modulation. Electric Power Systems Research 89, 23-30 (2012)
10. Paetz, J.: Reducing the Number of Neurons in Radial Basis Function Networks with Dynamic Decay Adjustment. Neurocomputing 62, 79-91 (2004)
11. Davey, P., Lalloo, D.G.: Drug Induced Chest Pain – Rare but Important. Postgrad Med J. 76, 420-422 (2000)
12. Berthold, M.R.; Diamond, J.: Constructive Training of Probabilistic Neural Networks. Neurocomputing 19, 167-183 (1998)
13. Cover, T.M., Hart, P.E.: Nearest Neighbor Pattern Classification. IEEE Transactions on Information Theory 13, 21–7 (1967)
14. Dasarathy, B.V.: Nearest Neighbor (NN) Norms: NN Pattern Classification Techniques. IEEE Computer Society Press, Los Alamitos, CA (1991)
15. Asuncion, A., Newman, D.J.: UCI Machine Learning Repository [http://www.ics.uci.edu/~mlearn/MLRepository.html]. Irvine, CA: University of California, School of Information and Computer Science (2007)
16. Specht, D.F.: Probabilistic Neural Networks. Neural Networks 3, 109-118 (1990)
17. Hudak, M.H.: RCE Classifiers: Theory and Practice. Cybernetics and Systems 23, 483-515 (1992)
18. Reilly, D.L., Cooper, L.N., Elbaum, C.: A Neural Model for Category Learning. Biological Cybernetics 45, 35-41 (1982)