

Pattern Classification Using Radial Basis Function Neural Networks Enhanced with the Rvachev Function Method

Mark S. Varvak

NAWCTSD, 12350 Research Parkway,
Orlando, Florida, 32826, USA
`mark.varvak@navy.mil`

Abstract. The proposed method for classifying clusters of patterns in complex non-convex, disconnected domains using Radial Basis Function Neural Networks (RBFNNs) enhanced with the Rvachev Function Method (RFM) is presented with numerical examples. R -functions are used to construct complex pattern cluster domain, parameters of which are applied to RBFNNs to establish boundaries for classification. The error functional is a convex quadratic one with respect to weight functions which take weight values on the discrete connectors between neurons. Activation function of neurons of RBFNNs is the $\text{sgn}(\cdot)$ function and, therefore, the error function is non-smooth. The delta learning rule during training phase is applied. The sub-gradient of the discretized error function is used rather than its gradient, because it is not smooth. The application of the RFM allows for the creation, implementation, and resolution of large heterogeneous NNs capable to solving diverse sets of classification problems with greater accuracy.

Keywords: Rvachev Function Method (RFM), clustering, classification, Radial Basis Functions (RBFs), Artificial Neural Networks (ANNs).

1 Introduction

The Vector Quantization clustering method for classifying clusters of vectors uses a covariance matrix and standard deviation or a Euclidian distance between the cluster centroid and a tested vector for classifying the tested vector for inclusion in that cluster [3,4]. This technique is applicable only to compact, simply connected, convex domains. A more powerful and robust clustering method is needed to classify vectors into pattern clusters in the more general types of geometrical domains.

This paper describes a new method for pattern cluster classification applying R -functions, developed by V.L.Rvachev in [8], and used in combination with RBFNNs. These R -functions depict a cluster's shell, i.e., boundaries of convex as well as non-convex, and connected as well as disjoint domains in \mathbb{R}^n in closed functional form.

The proposed method applies the R -functions as factors to the Radial Basis Functions (RBFs) in RBFNNs that ensure improved cluster classification with, consequently, improved accuracy of class representations and simplified NNs cluster classification algorithms.

Support Vector Machines (SVMs) with Bayes decision rules are also used in multi-category classification [5], but the statistics upon which these classification methods are based does not support classification of clusters in a complex domain.

RBFNNs are often used in the classification of scattered data because they can describe cluster property of the data via parameters [2,3]. However, the description of data's cluster properties can be improved using the Rvachev Function Method (RFM) with parameterized R -functions. In addition, implementing the RFM allows creating and applying large heterogeneous neural networks to diverse sets of classification problems with greater accuracy.

Traditional Neural Network techniques are also used in solving data classification problems [4]. But, the acceptable results require large amounts of computational power and the classification accuracy is often insufficient for practical applications. For example, it is impossible to classify clusters by a perceptron whose inputs are taken from regions of limited diameter [7]. The proposed clustering method, based upon the application of R -functions in the construction of general type cluster domains together with the RBFNNs approach copes with this constraint.

2 The Rvachev Function Method and R -Functions

R -functions possess the property of almost differentiability in \mathbb{R}^n and inherit some sort of boolean properties such as being positive or negative in some specific regions according to the boolean functions from which they were constructed. R -functions are composed from real almost differentiable functions by "boolean operators" represented by their counterparts of operators over real functions. This means that for each "boolean operation" over real functions, there is a combination of functional operators over real functions. The resulting function has a single analytical expression in the sense that it is represented by closed form formulas of almost differentiable functions. An R -function is an implicit real function whose sign is determined solely by the signs of its arguments [8]. An R -function and its arguments are real-valued, but by interpreting positive values as *true* and negative values as *false*, an R -function is transformed into equivalent boolean function. Boolean functions and corresponding R -functions are termed "friends". An R -function is defined mathematically as a mapping $f(\mathbf{x}) : \mathbb{R}^n \mapsto \mathbb{B}$, where \mathbb{R}^n is an n -dimensional space of real values and \mathbb{B} is the boolean bipolar space $\{-1, 1\}$. The function $f(\mathbf{x})$ is implicitly defined in \mathbb{R}^n as almost differentiable a finite number of times.

For the purpose of demonstrating the strengths of boolean closed form implicit real functions and the RFM, the examples of the constructed boolean closed form implicit real function in 2-dimensional domains Ω are presented below.

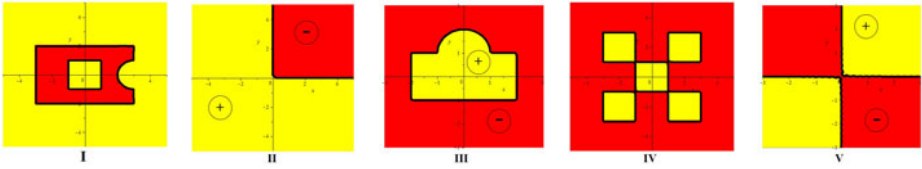


Fig. 1. i) **R**-function "Drawing" given in work [8, page 77]. ii) **R**-function "NAND" given by (1). iii) **R**-function "Rectangle + Half Circle" given in work [8, page 66]. iv) **R**-function "Tic Tac Toe" given in work [8, page 71]. v) **R**-function "XNOR" given by (2).

1. Closed functional expression [8, page 77] for "Drawing" domain shown on the position (i) of Figure 1.
2. Closed functional expression for "NAND" domain shown on the position (ii) of Figure 1.

$$\omega_2(x, y) = \frac{1}{2} \times (|y - x| - x - y) > 0. \tag{1}$$

3. Closed functional expression [8, page 66] for "Rectangle + Half Circle" domain shown on the position (iii) of Figure 1.
4. Closed functional expression [8, page 71] for "Tic Tac Toe" domain shown on the position (iv) of Figure 1.
5. Closed functional expression for "XNOR" domain shown on the position (v) of Figure 1.

$$\omega_5(x, y) = xy > 0. \tag{2}$$

Thus, this demonstrates capabilities of boolean analytical closed form functions in construction of versatile geometrical objects.

3 Discrete-Continuous NNs and *R*-Functions

The application of NNs as classifiers has a sufficiently long history and many established results reflected in articles and summarized in monographs [3,4]. Nonetheless, there are situations where the regular NN method does not work well and may lead to errors in classification of patterns of clusters. To improve and avoid such misclassifications many authors propose different types of computationally complex techniques: additional layers of nodes, new activation functions, changing network topology with connectors, modification of learning rules, new target optimal functions, and then substantiate these techniques with numerical experiments. For rigorous users these methods cause somewhat feelings of dissatisfaction and lack of confidence in the results of these computational experiments even though there are some asymptotical and probabilistic error estimations of misclassification.

The approach presented here with applications of *R*-functions and NNs can be viewed as a generalization or modification of Radial Basis Functions (RBFs)

with NNs when optimizing parameters include not only weights of nodes' connectors but also parameters of outlining shells of clusters, and tends to increase confidence in the numerical results.

At first, we describe regular NNs to be used for classification of clusters of pattern and then we will show the proposed modification aimed to improve the quality of classification even in difficult cases such as: non-convex, and disjoint domains of clusters of patterns. In NNs literature this situation has been termed *meshed and/or general type* domains [6].

For typical linear node of a NNs we have the activation function ϕ which provides linear separation of clusters of patterns on scalar product of vectors of weights w_i and inputs x_i .

$$y = \phi \left(\sum_i^n w_i x_i - \theta \right), \tag{3}$$

where w_i are weights of connectors, x_i are input values, $\phi(\cdot)$ is an activation function, which in this article particularly is $\text{sgn}(\cdot)$, and θ is the threshold.

We consider incoming into neuron values x_i subject to Lebesgue - Stiltjes integration with integrand including an R -function ω_i , and Radial Basis Functions f_i with a differential of Lebesgue measurable weight function $W(s)$ in the field S . We use the following definition of Radial Basis Function (RBF):

A function $f : \mathbb{R}^d \rightarrow \mathbb{R}$ is said to be radial if there exists a function $\psi : [0, \infty] \rightarrow \mathbb{R}$ such that $f(x) = \psi(\|x\|_2), \forall x \in \mathbb{R}^d$ [9].

$$y = \text{sgn} \left(\int_S \omega(\bar{\mathbf{x}}) (f(\|\bar{\mathbf{x}} - \bar{\mathbf{r}}\|_2) - \theta) dW(s) \right) \tag{4}$$

$$= \begin{cases} \text{sgn} \left(\int_S \omega(\bar{\mathbf{x}}) (f(\|\bar{\mathbf{x}} - \bar{\mathbf{r}}\|_2) - \theta) w(s) ds \right) & \text{if } W(s) \text{ is Riemann integrable} \\ \text{sgn} \left(\sum_i \omega(\bar{\mathbf{x}}) (f_i(\|\bar{\mathbf{x}} - \bar{\mathbf{r}}_i\|_2) - \theta) w_i \right) & \text{if } W(s) \text{ is discrete.} \end{cases}$$

Thus, we consider a neuron immersed into a continuous field with discrete connectors between neurons. The continuous part of so called neuron field may resemble any known physical field like electrical, gravitational, etc. Properties of such field has to be defined phenomenologically. This functional may contain aging and stochastic parameters. In the given form (4), it expresses energy of stimulus factors $\bar{\mathbf{x}}$ on the density of a neuron field $w(s)$. Based on the similarities in operations between R -functions and traditional neurons: see section 2, (3),(4), a heterogeneous NNs can be constructed whilst it is not exploring here.

4 Error Functions for Classification in RBFNNs Training

We consider Sum-of-Squares Error SSE function, which is given by a sum over all patterns $\bar{\mathbf{x}}$ in the training set, over all k outputs of vector $\bar{\mathbf{y}} = (y_1, y_2, \dots, y_k)$, and a corresponding vector field of weights $\bar{\mathbf{W}}$, with a regularization factor λ in the form

$$SSE(\bar{\mathbf{W}}) = \|\bar{\mathbf{y}}(\bar{\mathbf{x}}) - \text{sgn} \left(\int_S \omega(\bar{\mathbf{x}}) (f(\|\bar{\mathbf{x}} - \bar{\mathbf{r}}\|_2) - \theta) d\bar{\mathbf{W}}(s) \right)\|_2^2 + \lambda \|\bar{\mathbf{w}}\|_2^2. \tag{5}$$

For only discrete connections between neurons, Lebesgue-Stiltjes integration is converted from (5) into summation, shown in (6), over weighted w_i connection values for each output $y_j, j = \overline{1, k}$. Thus, in the absence of a field of interactions, and only for the discrete case we have for SSE

$$SSE(\bar{\mathbf{w}}) = \|\bar{\mathbf{y}}(\bar{\mathbf{x}}) - \mathbf{sgn}\left(\sum_{i=1}^k \omega(\bar{\mathbf{x}})(f(\|\bar{\mathbf{x}} - \bar{\mathbf{r}}\|_2) - \theta)w_i\right)\|_2^2 + \lambda\|\bar{\mathbf{w}}\|_2^2. \quad (6)$$

4.1 Problem Formulation

To solve classification problem of finding discriminant expression separating clusters of different classes in a NNs form, we need to find

$$\mathbf{w}^* \in W^* = \mathit{argmin}\{SSE(\mathbf{w}) : w \in W, |\omega(\bar{x})| \neq 0\}. \quad (7)$$

It is assumed that the optimal set of weights W^* is not empty and is bounded. To find a minimum of (7) the gradient method is used. Since the objective function (7) is not everywhere differentiable and functionally complex, we use finite differences. The calculation of finite differences can be done numerically by perturbing each weight sequentially. Our activation function is the $\mathbf{sgn}(\cdot)$ which is differentiable only in a generalized sense and, therefore, can be estimated as some value $g(\bar{\mathbf{x}}_k; w_i)$ for error on the pattern $\bar{\mathbf{x}}_k$. Thus, we have for the batch learning delta rule

$$\Delta w_i = -\eta \sum_k w_{kj}(\mathbf{y}(\bar{\mathbf{x}}_k) - \mathbf{o}(\bar{\mathbf{x}}_k))g(\bar{\mathbf{x}}_k; w_i). \quad (8)$$

We denote $\mathbf{o}(\bar{\mathbf{x}}_k)$ as vector node output corresponding to input vector pattern $\bar{\mathbf{x}}_k$, $\mathbf{y}(\bar{\mathbf{x}}_k)$ as the target vector corresponding to the same input vector pattern, $g(\bar{\mathbf{x}}_k; w_i)$ as the sub-gradient of the optimization problem, and η as learning rate.

4.2 RBFs and R-Functions for Representation of Cluster of Patterns

Radial Basis Functions (RBFs) are used widely in neural networks and it is known that interpolating properties of RBFs are insensitive to their precise form, but the quality of interpolation depends on parameters of interpolating RBFs. RBFs are smoothing and averaging outputs of noisy and scattered data and, therefore, decently represent clusters of data (patterns). Mid-layer of RBFNNs serves for unsupervised training to determine parameters for cluster representation [1]. Adding to these functions a multiplier in the form of R-functions we enhance the capability to describe clusters of patterns for classification procedure.

Let $\omega(x, y)$ be an R-function which describes some domain of patterns. This geometrical description might not precisely outline boundaries of clusters. Performing some transformations, we can adjust the description of boundaries. We

restrict ourselves only to linear transformation in order to substantiate concept on numerical experiments, namely : translation, rotation, scaling, and shearing.

An R -function $\omega(x, y; a, b, \theta, \gamma_1, \gamma_2, s_1, s_2)$ with transformational parameters $a, b, \theta, \gamma_1, \gamma_2, s_1, s_2$ can be adjusted to given clusters in such a way that the clusters will fit better into the boundaries of domains. A general form of transformation including scaling, shearing, rotation, and translation has the expression

$$\begin{aligned} x' &= s_1x \cos \theta - \gamma_1y \sin \theta + a \\ y' &= \gamma_2x \sin \theta + s_2y \cos \theta + b, \end{aligned} \tag{9}$$

where a, b are components of a translational vector, s_1, s_2 are scaling parameters in the direction of x, y respectively, γ_1, γ_2 are shearing parameters, and θ is the rotational parameter.

5 Numerical Explorations: Substantiation of Concepts

To substantiate concepts of this new method for cluster classification, we select clusters for numerical experiments shown in Figures 2 and 3.

For each form of clusters we present equation of boundary with transformational parameters. For case "STRIP" with transformational parameters $a, b, \gamma_1, \gamma_2, \theta$

$$\omega_{strip}(x, y) = 0.25 - (s_1x \cos(\theta) - \gamma_1y \sin(\theta) - a)^2. \tag{10}$$

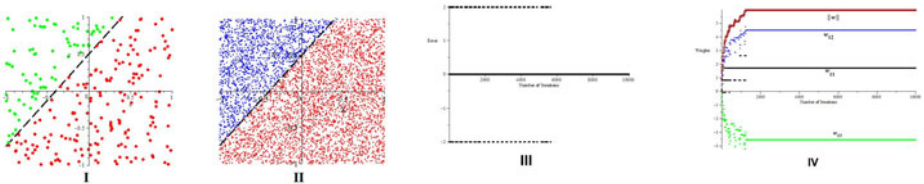


Fig. 2. i) RBFNNs training on 250 patterns in 2 clusters separated by (12). ii) Error of testing 0.4% on 5000 patterns in 2 clusters separated by RBFNNs corresponding to (12). iii) Error of training on 250 patterns in 2 clusters separated by (12). iv) Weights after training on 250 patterns in 2 clusters separated by (12).

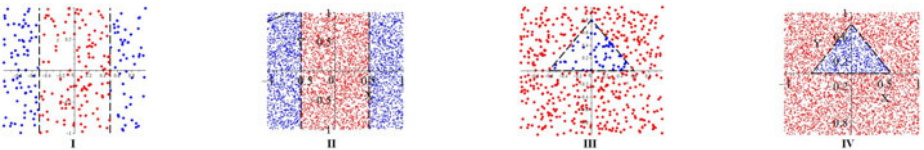


Fig. 3. i) RBFNNs training on 250 patterns in 2 clusters separated by (10). ii) Error of testing 0.1% on 5000 patterns in 2 clusters separated by RBFNNs corresponding to (10). iii) RBFNNs training on 250 patterns in 2 clusters separated by (11). iv) Error of testing 0.32% on 5000 patterns in 2 clusters separated by RBFNNs corresponding to (11).

For case "TRIANGLE"

$$\begin{aligned} \omega_{triangle}(x, y) = & 8. - \gamma_2 x \sin(\theta) - s_2 y \cos(\theta) + b - \\ & \frac{8}{3} \times | - s_1 x \cos(\theta) + \gamma_1 y \sin(\theta) + a | - | 8. - 3\gamma_2 x \sin(\theta) - \\ & 3s_2 y \cos(\theta) + 3b - \frac{8}{3} \times | - s_1 x \cos(\theta) + \gamma_1 y \sin(\theta) + a ||, \end{aligned} \tag{11}$$

and case "LINE"

$$\begin{aligned} \omega_{line}(x, y) = & 1.25s_1x + 1.25\gamma_1y - 1.25a - \\ & \gamma_2x - s_2y + b + .5. \end{aligned} \tag{12}$$

As Radial Basis Functions for hidden layer in the RBFNNs we take system of functions

$$\psi_j(x, y; r_{j1}, r_{j2}) = \exp[-(x - r_{j1})^2 - (y - r_{j2})^2], j = 1, 2, 3. \tag{13}$$

For the proposed numerical experiments, we construct perceptron with two input nodes for patterns x, y , three nodes of hidden layer, and one output node in output layer. Thus, our output function for classification has the form

$$z(x, y) = \mathbf{sgn} \left[\sum_{j=1}^3 w_j \omega_{cluster}(x, y; a_j, b_j, \gamma_{1j}, \gamma_{2j}, \theta_j, s_{1j}, s_{2j}) \psi_j(x, y; r_{j1}, r_{j2}) \right].$$

For example, for clusters separated by a line $\omega_{line}(x, y)$, we reached error of classification on 250 patterns equal to zero after 6000 iterations. RBFNNs weight parameters stabilized, shown on Figure 2, and testing the trained RBFNN for pattern classifications on 5000 patterns showed error of 0.4%. Analogous results for clusters in cases: "STRIP", "TRIANGLE" were received with the difference found only with the values where error of classification tend to zero and parameters of RBFNNs become stabilized on 250 patterns. Results of checking of classification of significantly bigger number of patterns (5000 patterns) shown on the corresponding figures and further substantiate the concept of the new method numerically.

6 Summary

The proposed method for classification of clusters of patterns enables classification of pattern clusters in complex domains, such as: non-convex, non-connected domains on presented numerical examples.

R -functions used to construct complex domain containing pattern clusters perform clustering role together with RBFs in the RBFNNs method. Parameters of RBFs define centers of pattern clusters and parameters of R -functions tune clusters within their boundaries.

The error functional is a convex quadratic one with respect to weight functions which takes weighted values at the discrete connections between neurons. As a generalization of traditional neural networks, a neuron continuous field can be considered; and scalar product between the stimulus patterns and the weights of connections represents energy delivered to a neuron.

The activation function of neurons is the $\text{sgn}(\cdot)$ function, and therefore, the error functional is non-smooth. For numerical examples the continuous part of neural fields is neglected and only the discrete part is used.

For neural networks learning, the delta rule is applied where instead of the gradient factor, the sub-gradient of the discretized error function is used because the error function is not smooth.

In case of multiple classes of pattern cluster classification, the RBFNNs together with R-functions must have k outputs such that this neural network can distinguish up to 2^k classes.

The application of the RFM allows for the creation, implementation, and resolution of large heterogeneous NNs capable to solving diverse sets of classification problems with greater accuracy in principle. In this case, the proposed clustering method copes with the classification of clusters by a perceptron whose inputs are taken from regions of limited diameter [7].

References

1. Bishop, C.M.: Neural Network for Pattern Recognition. Oxford University Press (1995)
2. Buhmann, M.: Radial Basis Functions. Theory and Implementation. Cambridge University Press (2003)
3. Duda, R.E., Hart, P.E., Stork, D.G.: Pattern Classification. Wiley Inerscience (2000)
4. Hassoun, M.H.: Fundamentals of Artificial Neural Networks. The MIT Press (1995)
5. Lee, Y., Lin, Y., Wahba, G.: Multicategory Support Vector Machines: Theory and Application to the Classification of Microarray Data and Satellite Radiance Data. *J. of the American Statistical Association* 99(465), 67–81 (2004)
6. Lippman, R.P.: An Introduction to Computing with Neural Nets. In: Lau, C. (ed.) *Neural Networks. Theoretical Foundation and Analysis*. IEEE Press (1992)
7. Minsky, M.L., Papert, S.A.: *Perceptrons*. MIT Press, Cambridge (1990)
8. Rvachev, V.L.: *Geometric Applications of Logic Algebra*. Naukova Dumka, Kiev (1967) (in Russian)
9. Wendland, H.: *Scattered data approximation*. Cambridge University Press (2010)