Estimation of Radial Basis Function Network Centres via Information Forces

1st E. F. Sousa Júnior
Technological Centre
UFPI
Teresina, Brazil
edilsonferreira.junior@gmail.com

2nd A. A. Carneiro de Freitas
Technological Centre
UFPI
Teresina, Brazil
carneirofreitas@uol.com.br

3rd R. A. Lira Rabelo
Natural Science Centre
UFPI
Teresina, Brazil
ricardoalr@ufpi.edu.br

4th W. R. N. Santos.
Technological Centre
UFPI
Teresina, Brazil
welflen@ufpi.edu.br

***Abstract*—The Radial Basis Function Network centres determination is a open problem. In this work, the cluster centres are determined by a proposed gradient algorithm using the information forces acting on each data point. These centres are applied to a Radial Basis Function Network for data classification. A threshold is established based on Information Potential to classify the outliers. Combined, the threshold and the centres determined by information forces show good results in comparison to a similar Network with a k-means clustering algorithm.**

***Index Terms*—Radial Basis Functions Networks, Classification, Clustering and Outliers.**

I. Introduction

Broomhead and Lowe in 1988 [1] presented the Radial Basis Function Network (RBFN) concept. It is a universal approximator [2] [3]. The training of a RBFN is done in two stages: initially, the centres c_j and the variance σ_j of the basis functions are determined, then, the network weights w_{ij}. This work focuses on determining these RBF centres. Clustering techniques can be used to determine the RBF centres. These techniques find the cluster centers that reflects the distribution of the data points [4]. The most common is the k-means algorithm [5].

The Information Potential (IP) and Information Force (IF) constitute two concepts of Information Theory [6] that describe, respectively, the amount of agglomeration and the direction where this agglomeration increases. These concepts are used in some clustering techniques, such as the one developed by Jenssen et. al. [7].

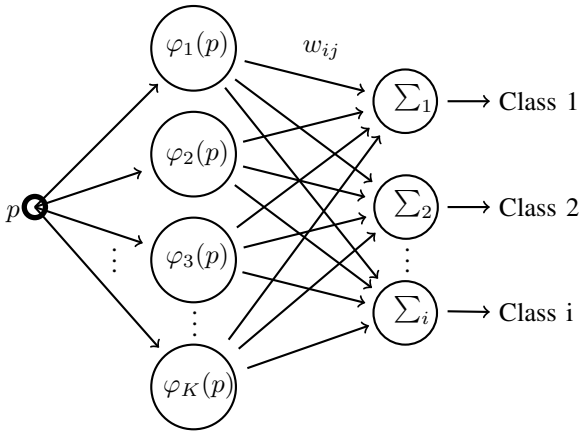
This work presents two algorithms that are developed based on those two concepts above. The principal one finds the cluster centres by a gradient algorithm using Information Forces. These cluster centres are applied to the RBFN. The second one uses the concept of Information Potential to reduce the number of outliers. The algorithms are applied to a data classification problem.

This article was organised as follows: In Section II, the RBFN is illustrated; In Section III, the concepts of Information Potential and Force are described, also the algorithm to estimate the RBFN centres is presented. In Section IV, the algorithm to reduce the outliers is described; In Section V, the Data is presented; In Section VI, results are displayed and the algorithm parameters are analysed. In Section VI, conclusion and future works are discussed.

II. Radial Basis Function Network

The RBFN is shown in figure 1. This network is composed by an input layer p, a hidden layer, and an output layer which provide the classification. When an input datapoint p is fed into a node, distance is calculated from a centre $\boldsymbol{c_j}$, transformed by a Radial Basis Function $\varphi_j(\cdot)$ and multiplied by a weighting value w_{ij} [8]. All the values produced in the K nodes are summed for each class, and the point p is classified where this sum is maximum.

Fig. 1: Artificial Neural Network



This Network is a function presented in the equation:

$$Class_i = f(p) = w_0 + \sum_{j=1}^{K} w_{ij}\varphi_j(p) \tag{1}$$

The methods to obtain those parameters influence the classification performance. In this work, the centres are estimated via IF and, for comparison, the k-means algorithm [5]. The weights w_{ij} are determined by pseudoinverse matrix and a Gaussian Function are chosen for RBF:

$$\varphi_j(\boldsymbol{p}) = \exp\left(-\frac{||\boldsymbol{p} - \boldsymbol{c_j}||^2}{2 \cdot \sigma_j^2}\right) \tag{2}$$

For the k-means clustering, the variance is estimated for each node as the average distance between the n_j data points p_i in the cluster and its centre c_j:

$$\sigma_j = \frac{1}{n_j} \cdot \sqrt{\sum_{i=1}^{n_j} (p_i - c_j)^2} \tag{3}$$

The IF algorithm proposed in this work does not separate the data into clusters, which makes it impossible to use equation 3. Therefore, for this algorithm, the variance is estimated by the equation proposed by Haykin [5]:

$$\sigma = \frac{d_{max}}{\sqrt{2 \cdot K}} \tag{4}$$

In which d_{max} is the maximum distance between the cluster centres and K is the number of nodes. The variance is equal for all nodes.

III. RBF centres estimation via IF Gradient algorithm

A. Information Forces

Considering $x_i \in \mathbb{R}^m$, $i = 1, 2, ..., n$ a set of samples belonging to a random variable $X \in R^m$, a Parzen Window [9] can be associated with a Gaussian kernel, directly estimating the Probability Density Function (PDF) of the data. This function can be described by:

$$\hat{f}_x(x) = \frac{1}{n} \sum_{i=1}^{n} G(x - x_i, h^2) \tag{5}$$

Where G is the Gaussian Kernel and h is the kernel bandwidth. There are several ways to estimate the ideal bandwidth h [10]. In Probability Density Function (PDF) that are near to the Normal distribution the Rule-of-Thumb [11] is the most practical and simple one:

$$h = 1,06 \cdot \min\{\hat{\sigma}, \frac{IQR}{1,34}\} n^{-\frac{1}{5}} \tag{6}$$

where n is the number of data points, $\hat{\sigma}$ is the estimated standard deviation of the dataset and $IQR = Q_3 - Q_1$ is the interquartile range.

The Renyi entropy equation of order two [12] is given by

$$H_{R_2}(X) = -\log[\int_X f_x^2(x) dx] \tag{7}$$

Applying the Parzen Window:

$$\begin{aligned} \int_X f_x^2(x) dx = \\ \int [\frac{1}{n} \sum_{i=1}^{n} G(x - x_i, h^2)]^2 dx = \\ \frac{1}{n^2} \sum_{i=1}^{n} \sum_{j=1}^{n} \int G(x - x_i, h^2) G(x - x_j, h^2) dx = \\ \frac{1}{n^2} \sum_{i=1}^{n} \sum_{j=1}^{n} G(x_i - x_j, 2 \cdot h^2) \end{aligned} \tag{8}$$

That means:

$$H_{R_2}(X) = -\log[\frac{1}{n^2} \sum_{i=1}^{n} \sum_{j=1}^{n} G(x_i - x_j, 2h^2)] \tag{9}$$

The argument of the natural logarithm above is the Information Potential over all the dataset, in an analogy with potential energy of physical particles [13].

$$P(\{x\}) = \frac{1}{n^2} \sum_{i=1}^{n} \sum_{j=1}^{n} G(x_i - x_j, 2h^2) \tag{10}$$

The Information Potential over a single point x_i in the dataset is the sum of interactions of this point across all the dataset.

$$P(\{x_i\}) = \frac{1}{n^2} \sum_{j=1}^{n} G(x_i - x_j, 2h^2) \tag{11}$$

The IP indicates the amount of agglomeration around the point. Its derivative is the Information Force acting in point x_i [7].

$$\begin{aligned} \boldsymbol{F}_{if} = \frac{\partial}{\partial(\boldsymbol{x}_i - \boldsymbol{x}_j)} P(\{x_i\}) = \\ -\frac{1}{N^2 \cdot h^2} \sum_{j=1}^{n} G(x_i - x_j, 2h^2) \cdot (\boldsymbol{x}_i - \boldsymbol{x}_j) \end{aligned} \tag{12}$$

B. Gradient Algorithm

The IF points in the direction where the amount of agglomeration increases. Then, a centre candidate c_i can approximate to the central cluster by successive interaction of the equation:

$$c_i(s+1) = c_i(s) + \eta \boldsymbol{F}_{if} \tag{13}$$

This candidate c_i could erroneously converge to a local maximum, similar to other algorithms based on Gradient Descent/Ascent. Two approaches via Information Theory could minimise this error. The first one is reducing the number of local maximum by smoothing the IP distribution over. [7]. The ideal h (eq. 6) is multiplied by a parameter $\kappa > 1$ which smooths the PDF's distribution over.

Another solution is to variate the learning rate η over the data space. The magnitude of the IF is bigger in the border of the cluster and decreases as the candidate c approximates to the central cluster and the force vectors are balanced out. Then,

$$\eta = \alpha \cdot e^{||\boldsymbol{F_{if}}||} \tag{14}$$

Outliers also hinder the IF gradient algorithm. They are removed from the data in the training phase for a better estimation of the cluster centres. Candidates with small Information Potential behave like outliers. Then, they are also removed for the centre estimation. The detailed description of the IF Gradient Algorithm is illustrated in Algorithm 1.

Initially, a set of centre candidates c is raffled between the dataset. This set is sufficiently big to ensure that at least one point is raffled on each cluster. Some candidates could be too close. In this case, one of the points is eliminated. Many candidates tend to converge to a single central cluster. On each

Algorithm 1: Estimation of RBFN Centres via IF

input :
- X_{train} % Data to train the RBFN;
- α % Learning rate constant;
- n_center % Number of initial candidates;
- δ % Threshold of Information Potential;
- β % Minimum distance between candidates;
- max_epochs % Maximum epochs;
- γ % Constant of convergence;

output:
- $C_{cluster}$ % Cluster centres.

begin
  % $P_{if}(\cdot)$ calculates the information potential.
  % $F_{if}(\cdot)$ calculates the information force.
  for $i \in 1 : n\_center$ **do**
    % Raffle the candidates
    $C_{cand}(i) = random(X_{train})$;
    if $P_{if}(C_{cand}(i)) < \delta$ **then**
      Eliminate $C_{cand}(i)$;
    end
  end
  $s = 0$;
  while *($s < max\_epochs$) and (not all $C_{candidate}$ are eliminated or converged)* **do**
    % Eliminate points to close each other
    for i *and* $j \in 1 : n\_center$ **do**
      if $||(C_{cand}(i) - C_{cand}(j))|| < \beta$ **then**
        Eliminate $C_{cand}(i)$
      end
    end
    % Update the center candidate
    for $i \in 1 : n\_center$ **do**
      $C_{cand}(i, s+1) = C_{cand}(i, s) + \alpha * e^{||F_{if}(i,s)||} * F_{if}(i, s)$
      if $|e^{||F_{if}(i,s+1)||} - e^{||F_{if}(i,s)||}| < \gamma$ **then**
        $C_{cand}(i)$ converge
      end
    end
    $s = s + 1$
  end
  $C_{cluster} = \cup C_{cand}$ that converge.
  return $C_{cluster}$
end

interaction, if two candidates are too close to each other, one of them is eliminated.

The points raffled with small IP constitute another problem. Far from the central cluster, the greatest information force is exerted by the point initially picked. In this way, the centre candidate c_i is stuck to the starting point. To avoid that, the IP is calculated in the initial epoch over the candidates. If it is below a threshold, the centre candidate is eliminated.

The interactions over a specific centre candidate stop when the difference is reached:

$$|\exp(|\boldsymbol{F_{if}}|)(s+1) - \exp(|\boldsymbol{F_{if}}|)(s)| < \gamma \qquad (15)$$

Where γ is a small value described in result section. When a centre candidate nears a cluster centre, the forces tend to equilibrium and the left-hand side of inequality 15 tends to zero.

The algorithm completely stops when all the candidates' centres converge or are eliminated. If they do not converge, it stops when it reaches the maximum number of epochs.

IV. Outlier Reduction

The RBFN has difficulties in identifying the outliers. A mechanism of outlier detection improves the RBFN results. This can be done by observing the IP on each point, because outliers have small information potential.

A threshold δ can be established with the training data. Then, this threshold can be applied to the test data and most outliers can be identified. The detailed description of this mechanism is in algorithm 2 below.

Algorithm 2: Outlier Detection

input :
- X_{train} % Data to train the RBFN;
- X_{test} % Data to test the RBFN;
- $n\_outlier$ % nº of outliers in training data;
- θ % constant of outlier reduction;

output:
- $C_{Outlier}$ % Set of outliers.

begin
  for $i \in 1 : size(X_{train})$ **do**
    $pot_{trei}(i) = P_{if}(X_{train}(i))$ % The IP
  end
  % Sort in ascend order.
  $pot_{trei} = sort(pot_{trei}, 'ascend')$
  $\delta = pot_{trei}(\theta \cdot n\_outlier)$ % The threshold.
  for $i \in 1 : size(X_{test})$ **do**
    if $P_{if}(X_{test}(i)) \leq \delta$ **then**
      $X_{test}(i) \in C_{Outlier}$
    end
  end
  return $C_{Outlier}$
end

The threshold δ is estimated using the IP values of the outliers. Some points in the clusters also have small IP and could be erroneously classified as outliers. The constant θ (< 1) is established to avoid this problem.

V. Data

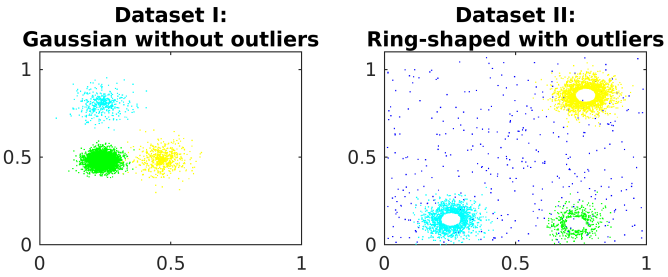
The data are generated artificially via the Multidimensional Dataset Generator for Clustering (MDCGen) [14]. Two datasets are generated with an ascendant level of complexity. All datasets are two-dimensional spaces with three clusters and points located in the interval $(0, 1)$ on all directions. Each set is

divided into two subsets: train and test, in a ratio respectively of 80%/20%. The data characteristics are presented in Table I and the data distribution in Figure 2.

TABLE I: Dataset characteristics.

	Distribution	Number of points	Number of outliers
Dataset I	Gaussian	4000	0
Dataset II	Ring-shaped	4400	400

Fig. 2: Dataset distributions.

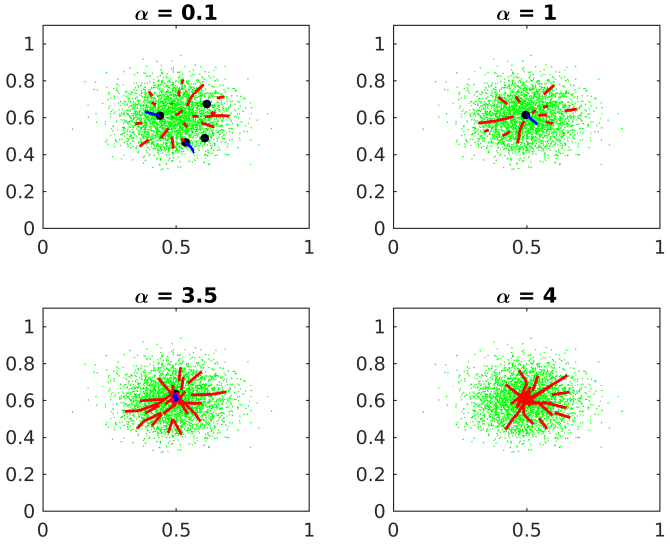


VI. Parameters and Results

A. Parameters

1) Learning rate constant: Figure 3 shows the effect of the Learning rate constant α on algorithm 1.

Fig. 3: Learning rate constant

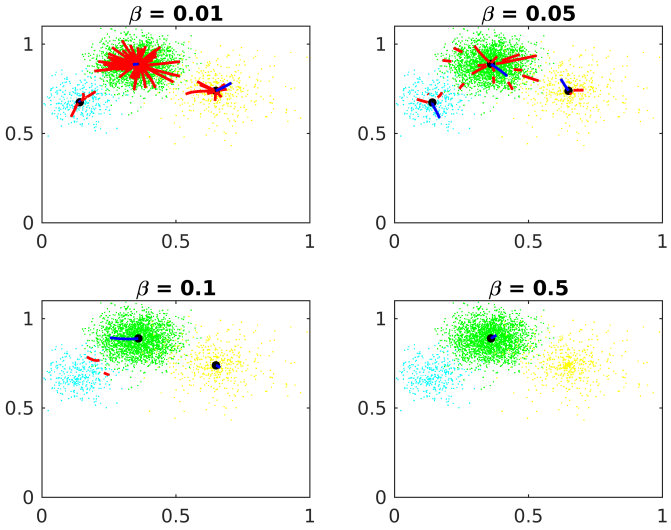


The black dots represent the estimated cluster centres. The red lines represent the trajectory of the eliminated candidates. The blue lines represent the converged ones.

If the constant α is too small, some centres candidates erroneously converge to local maxima. If α is too big, the candidates oscillate around the central cluster and the algorithm loses accuracy. If α is very big, the algorithm does not converge before reaching the maximum epochs. Experimentally, a good value for α is ten times the standard deviation of the clusters.

2) Minimum distance between centres candidates: Figure 4 shows the constant β effects on algorithm 1.

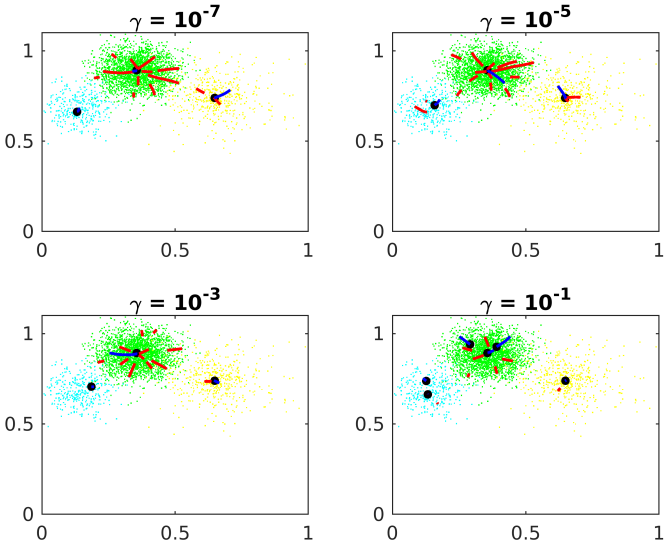
Fig. 4: Minimum distance between centres candidates



If this distance is too small, just a few points are eliminated on each epoch and the algorithm demands more computational effort. If β is too big, a centre candidate in one cluster could eliminate good centre candidates in other clusters. Experimentally, a good value for β is 10% of the standard deviation of the clusters.

3) Constant of convergence: Figure 5 shows the constant γ effects on algorithm 1.

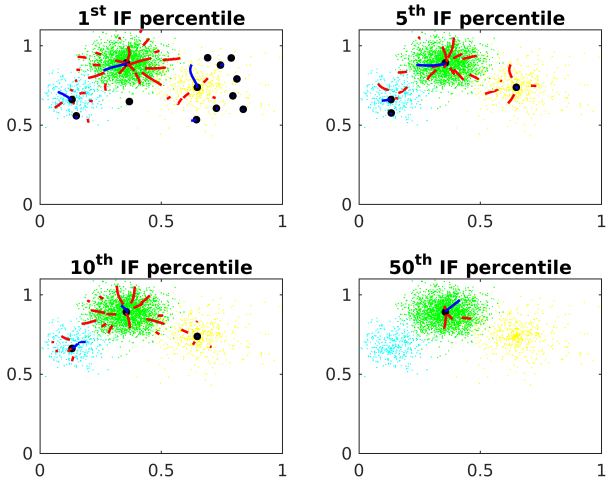
Fig. 5: Constant of convergence



The precision increases when the constant γ diminishes, although the algorithm takes more epochs to converge, requiring more computational effort. If γ is too big, the centre candidates stop before the forces balance out, far from the actual central cluster. Experimentally, $\gamma = 10^{-5}$ is an appropriate value.

4) IP threshold: Figure 6 shows the threshold δ effects on the algorithm 1. The potential is calculated at each point. The threshold is tested as the 1st, 5th, 10th and 50th percentile from the IP distribution of the points.

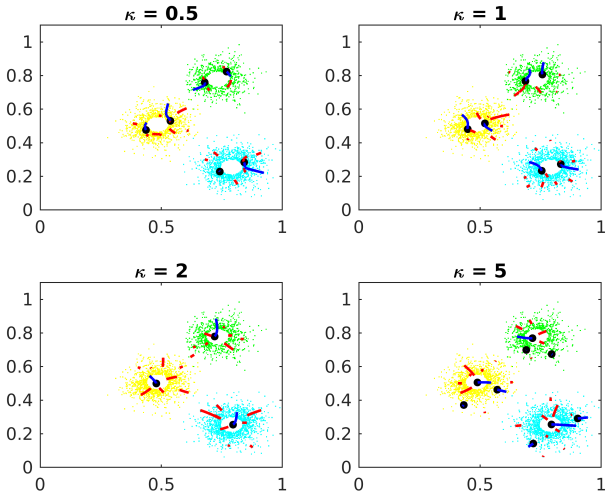
Fig. 6: IP threshold



Some centre candidates are raffled on points with small IP. These candidates are stuck close to the origin, not converging to the actual cluster centres. If the threshold δ is too small, these centre candidates are not eliminated by the algorithm. In the other side, if δ is too big, it eliminates good centre candidates in clusters with small IP.

5) Smoothie parameter: Figure 7 shows the parameter κ effects on Algorithm 1.

Fig. 7: Smoothie parameter.



If κ is small, the candidates converge to local maxima inside the clusters but far from the actual centre. If κ is big, points too far from the central cluster exert too much influence in the IF vectors, confusing the gradient algorithm. Experimentally, κ values between 1 an 3 enable good results.

6) Constant of outlier reduction: Table II shows, in Dataset II, the performance of the RBFN associated with Algorithm 2 using different values of the parameter θ.

TABLE II: Accuracy of the RBFN associated with the outlier reduction for different θ values.

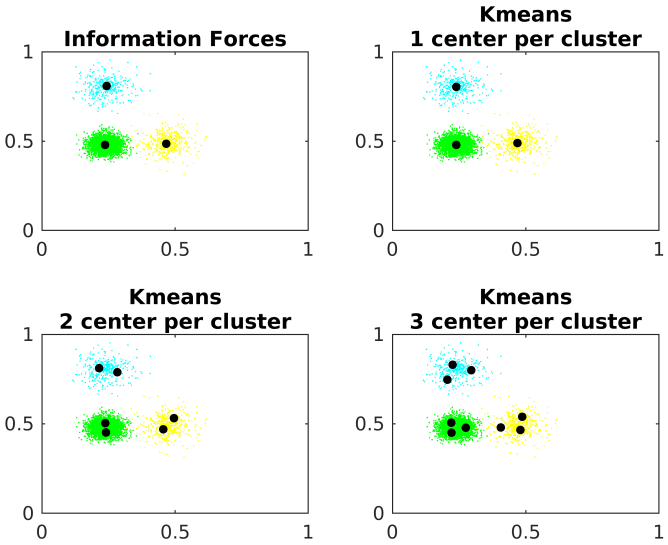
θ	Correctly Classified
0,5	96,82%
0,75	97,61%
1	96,14%

As described in section III, some points inside the cluster but far from the actual centre have small IP. The parameter θ partially avoids that the outlier reduction algorithm misclassifies these points. The value of θ depends on the concentration level of points in the clusters.

B. Results

The RBFN centres are estimated by information forces (Algorithm 1) and by k-means algorithm for comparison. The k-means are tested with one, two and three centres per cluster. The figures 8 and 9 show the centres location on each dataset.

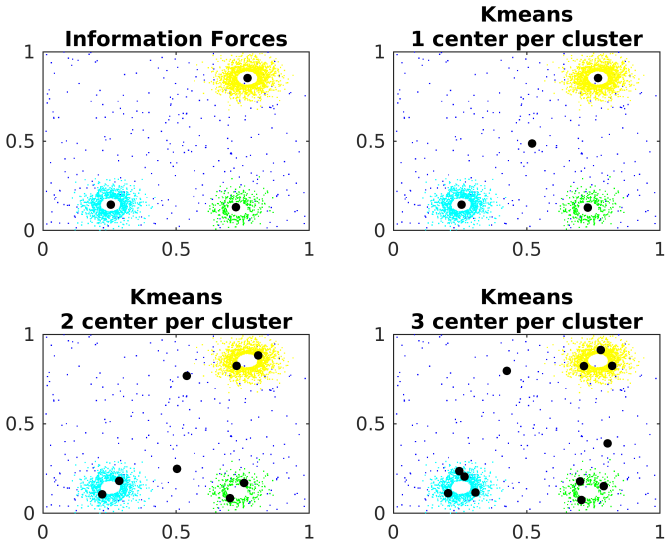
Fig. 8: Estimated cluster centres in dataset I



The black dots represent the estimated cluster centres.

The information forces point to one centre on each cluster. When both algorithms return to the same number of centres, their location are very similar. In dataset I, the centres determined via IF are, on average, only 0.0043 units away from the correspondent centres determined via k-means with one centre per cluster.

Fig. 9: Estimated cluster centres in dataset II



The estimated centres are applied to the RBFN. The percentage of correctly classified points are presented in table III.

TABLE III: Percentage of Accuracy in out-of-sample data.

	Accuracy out-of-sample	
	Dataset I	Dataset II
IF	99,50%	90,91%
IF with outlier reduction	-	97,61%
k-means 1	94,38%	96,36%
k-means 2	93,25%	93,30%
k-means 3	94,38%	93,30%

The estimation via IF without outlier reduction outperforms the k-means algorithm in datasets I. The RBFN with centres estimated by IF have good performance in datasets with a few or no outliers. In datasets II, the k-means outperforms. However, the IF gradient algorithm without outlier reduction still has a reasonable performance on this dataset. There is an improvement when the outlier reduction is used alongside the RBFN with centres estimated by IF. This improvement leads the IF gradient algorithm to outperform the k-means.

VII. Conclusion and Future Works

This proposed method to assign the RBFN centres presents satisfactory preliminary results in comparison to the traditional k-means algorithm. Also, the outlier reduction based on information potential improves the results. It is noteworthy that the proposed method accuracy depends on the correct adjustment of some parameters, but this also happens in other methods.

Future works may analyse the performance of the proposed methods on more complex and non artificially generated datasets. Further on, it may analyse how the RBFN behaves with the IF gradient algorithm alongside other methods to determine the basis function variance and the network weights.

References

[1] D. S. Broomhead and D. Lowe, "Radial basis functions, multi-variable functional interpolation and adaptive networks," tech. rep., **Royal Signals and Radar Establishment Malvern**, 1988.

[2] J. Park and I. W. Sandberg, "Universal approximation using radial-basis-function networks," ***Neural computation***, vol. 3, no. 2, pp. 246–257, 1991.

[3] T. Poggio and F. Girosi, "Networks for approximation and learning," ***Proceedings of the IEEE***, vol. 78, no. 9, pp. 1481–1497, 1990.

[4] C. M. Bishop *et al.*, ***Neural networks for pattern recognition***. Oxford university press, 1995.

[5] S. Haykin, ***Neural Networks and Learning Machines***. Prentice Hall, 2009.

[6] J. C. Principe, ***Information Theoretic Learning:*** *Renyi's Entropy and Kernel Perspectives*. New York: Springer, 2010.

[7] R. Jenssen, D. Erdogmus, K. E. Hild, J. C. Principe, and T. Eltoft, "Information force clustering using directed trees," in ***International Workshop on Energy Minimization Methods in Computer Vision and Pattern Recognition***, pp. 68–82, Springer, 2003.

[8] D. K. Wedding II and K. J. Cios, "Time series forecasting by combining rbf networks, certainty factors, and the box-jenkins model," ***Neurocomputing***, vol. 10, no. 2, pp. 149–168, 1996.

[9] E. Parzen, "On estimation of a probability density function and mode," ***The annals of mathematical statistics***, vol. 33, no. 3, pp. 1065–1076, 1962.

[10] A. Gramacki, ***Nonparametric Kernel Density Estimation and Its Computational Aspects***. Cham: Springer, 2017.

[11] B. Silverman, ***Density Estimation for Statistics and Data Analysis***. Chapman & Hall/CRC Monographs on Statistics & Applied Probability, Taylor & Francis, 1986.

[12] A. Rényi, ***Selected Papers of Alfréd Rényi:*** *1948-1956*, vol. 1. Akadémiai Kiadó, 1976.

[13] D. Xu, J. C. Principe, J. Fisher, and H.-C. Wu, "A novel measure for independent component analysis (ica)," in ***Proceedings of the 1998 IEEE International Conference on Acoustics, Speech and Signal Processing***, vol. 2, pp. 1161–1164, IEEE, 1998.

[14] F. Iglesias, T. Zseby, D. Ferreira, and A. Zimek, "Mdcgen: Multidimensional dataset generator for clustering," ***Journal of Classification***, vol. 36, no. 3, pp. 599–618, 2019.