2016 Joint 8th International Conference on Soft Computing and Intelligent Systems and 2016 17th International Symposium on Advanced Intelligent Systems

On the Parameter Setting of A Network-Growing Algorithm for Radial Basis Kernel Networks

Tetsuya Hoya
Department of Mathematics
College of Science and Technology, Nihon University
Tokyo, Japan
hoya@math.cst.nihon-u.ac.jp

***Abstract*—This paper reveals that the previously proposed network growing algorithm for radial basis kernel networks (a.k.a. `self-structuring kernel memory networks') essentially depends upon only a single network parameter of the radius unique to all the RBFs. It is also shown that the parameter range is bounded. Both these properties are in practice quite attractive, where an efficient design of a pattern classifier is desired. Simulation studies using four data sets, each extracted from public domain database for pattern classification tasks, confirmed this new findings.**

***Keywords*—radial basis functions; network growing algorithm; kernel memory networks**

I. Introduction

Since the early work by Broomhead and Lowe [1], neural network models based upon radial basis functions (RBFs; or RBF kernels) have been extensively studied and successfully applied to various application areas of interest.

Among many, a variant of the RBF network models known as probabilistic neural network (PNN) [2] was proposed for general pattern classification purposes. The salient feature of PNN compared with other RBF networks is the use of a kernel width (or radius) unique to all the RBFs and the fixed weights between an RBF and output units, each identical to the corresponding target value. Then, training of PNN is simply done; an RBF per training pattern is assigned. In many situations, however, this manner of training involves redundant addition of RBFs, and the resultant network size can even become prohibitively large after the training.

A number of efforts have been made, in order to circumvent the aforementioned problem specific to PNN, by way of constructive approaches [3]-[7] (note that the network exploited in [4] and [5] is essentially the same as PNN). In a similar context to the PNN, various constructive methods for RBF networks have also been proposed to date. Some well-acclaimed methods are i) resource allocation network (RAN) [8] and its enhancements -- RAN trained using extended Kalman filter (RAN-EKF) [9] and minimal RAN (MRAN) [10], ii) the method utilizing orthogonal least squares (OLS) [11][12], iii) the method based upon an evolutionary algorithm, such as genetic algorithm (GA) [13] or particle swarm optimization (PSO) [14][15].

This work is in part supported by the Nihon University Academic Research grant (Total Research 14-002).

A. *Comparison of Some Existing Constructive Methods for RBF Networks*

Most of the aforementioned constructive methods generally yield compact-sized classifiers, while achieving reasonably high generalization performance. However, in practice, they normally require many parameters to tune before their utility for the testing and/or involve some arduous approximation / iterative optimization procedure(s) over entire training data, the latter of which quite often leads to long training and may even suffer from numerical instability. These may give a good reason to detract from their utility in practical situations. Table I shows a comparison of some existing constructive methods in terms of i) the computational complexity and the cause, and ii) the number of parameters required to tune, in the training mode. In the second column in Table I, the numbers within the curly brackets followed after "high" indicate "(1)" possible involvement of the heavy computational load due to the resource required in both memory storage and computation wises, "(2)" that due to an iterative optimization procedure and the possibility of falling into local minima, and "(3)" the necessity of repeatedly presenting a whole training data set (i.e. typically more than twice), respectively.

TABLE I. Comparison of Some Existing Constructive Methods for PNN/RBF Networks

Constructive method	***Computational Complexity***	***Num. free params.***
[3][6]	high(3)	2
IPNN [7]	high(1)	2
RAN [8]	low	6
RAN-EKF [9]	medium	9
MRAN [10]	low	9
OLS [11]	high(1)/(2)	3
GA [13]	high(1)	6
PSO [14]	high(2)	5
SSKM [16]	low	1

In [16], a very simple network-growing algorithm for RBFs, termed the self-structuring kernel memory (SSKM) networks, is proposed. The SSKM can be viewed as a variant of PNN, in that the model inherits the use of a unique radius

978-1-5090-2678-4/16 $31.00 © 2016 IEEE
DOI 10.1109/SCIS&ISIS.2016.168

for all the RBFs and binary weight values between the RBFs and output units. For an M-class pattern classification task, M distinct SSKM networks (i.e. for a single-domain pattern classification task; for the detail, see [16]), each responsible for one of the M classes and consisting of multiple RBFs, are generated by applying a network-growing algorithm; automatic RBF assignment within the respective distinct SSKM networks is performed based upon a non-iterative one-pass training procedure and free from all the aforementioned three concerns relevant to the computational complexity. Unlike typical RBF-type models, lateral connections (coined as "link weights" or LWs, for short) between some of the RBFs within a single SSKM network may also be established during the construction phase. In the testing phase, a maximally activated RBF among all the RBFs within all the M distinct SSKM networks is picked out, and the network ID of the RBF so chosen corresponds to the final classification result.

Although the construction algorithm [16], to be reviewed in the next section, originally came with two free parameters, this paper shows that there is essentially only a single parameter to tune, i.e. a unique radius of RBF, during the construction phase. Hence, the number of the degree of freedom equals that of the original PNN model [2]. It is also shown that the range of the radius value is theoretically bounded. These properties are quite attractive in practice, for a constructive design of RBF-based pattern classifiers.

II. Review of the Self-Structuring Algorithm for Kernel Memory Networks [16]

The construction algorithm [16] for the m-th (m=1,2,..., M) distinct SSKM network, responsible for Class m, is summarized as follows:

1) Initially, there is only a single RBF in the m-th SSKM network, with the centroid vector $\boldsymbol{c}_1=\boldsymbol{x}(1)$ and the radius σ. Set $N(m)=1$.

2) Repeat the following, for all the training patterns $\boldsymbol{x}(n)$ ($n = 2,3,\ldots,P(m)$) that fall into Class m:

   - If there is no excited RBF in the network by $\boldsymbol{x}(n)$ i.e.

   $$h_i(\boldsymbol{x}(n)) = \exp\left(\frac{-\|\boldsymbol{x}(n)-\boldsymbol{c}_i\|_2^2}{\sigma^2}\right) < \theta \quad (i = 1,2,\ldots,N(m)), \tag{1}$$

   where $\|\bullet\|$ denotes $L2$-norm and θ ($0<\theta<1$) is a given threshold, add a new RBF in the m-th SSKM network i.e.

   $$N(m) = N(m)+1$$
   $$\boldsymbol{c}_{N(m)} = \boldsymbol{x}(n) \tag{2}$$

   with the radius σ.

   - Otherwise, if there are no LWs between the excited RBFs, established new LW connections among all such RBFs within the m-th network.

In Step 2) above, notice that some RBF units can also be excited by the activation transfer via the LW connection from other units, even if they are not initially excited by the direct input $\boldsymbol{x}$; if an LW connection is established between the p-th and q-th RBF units in an SSKM network and if the q-th unit is excited by $\boldsymbol{x}$, whereas the p-th unit is not, the p-th unit can still be excited if

$$h_p(\boldsymbol{x}) = w_{pq} h_q(\boldsymbol{x}) \geq \theta \ \ (p \neq q) . \tag{3}$$

As in [16], we hereafter consider only the case where LWs have all unity weight values, i.e. w_{pq} = 1, without loss of generality. In the construction phase, the network-growing algorithm shown in the above is applied to form all the M distinct SSKM networks.

As described above, the algorithm has two free parameters σ and θ, but these parameters are shown to be interrelated in the next section.

III. Interrelation Between the Unique Radius and Threshold Value for the Self-Structuring Algorithm

In the network-growing algorithm for SSKM as shown in the previous section, whether addition of a new RBF unit takes place or not depends only upon the condition (1). Then, we consider the case where a particular training pattern $\boldsymbol{x}(n)$ is given to an RBF h_i such that

$$h_i(\boldsymbol{x}(n)) = \theta . \tag{4}$$

Let the distance d between $\boldsymbol{x}(n)$ and $\boldsymbol{c}_i$ $d = \|\boldsymbol{x}(n)-\boldsymbol{c}_i\|_2 \ (\geq 0)$, and rewriting (4) yields the relation

$$d^2 = \sigma^2 \ln\left(\frac{1}{\theta}\right) \ \text{or} \ \sigma^2 = \frac{d^2}{\ln\left(\frac{1}{\theta}\right)}$$

Since $0<\theta<1$, $\ln\left(\frac{1}{\theta}\right) > 0$, and since $\sigma > 0$, the relation above can be rewritten:

$$d = \sigma\sqrt{\ln\left(\frac{1}{\theta}\right)} \ \text{or} \ \sigma = \frac{d}{\sqrt{\ln\left(\frac{1}{\theta}\right)}} \tag{5}$$

The former indicates that there is essentially only a single combination of the two parameters σ and θ that yields a unique value d, whereas the latter that the radius σ is given as a function of θ.

Hence, the reduction in terms of the numbers in the degrees of freedom (i.e. from two to one) can be achieved for the SSKM construction.

A. Lower and Upper Bounds for the Value of σ

As described in II, when a new RBF is allocated within an SSKM network, its centroid vector is set to the training pattern. Since the contents of the centroid vectors within SSKM networks remain intact during the construction phase, they do not alter from those of the training pattern vectors originally given. Therefore, the following condition is met:

$$d_{\min} \le d \le d_{\max}, \quad (6)$$

where $d_{\min}$ and $d_{\max}$ are a minimum and maximum distance found among all the pairs of training pattern vectors, respectively, i.e.

$$d_{\min} = \min(\|\boldsymbol{x}(n_1) - \boldsymbol{x}(n_2)\|_2),$$
$$d_{\max} = \max(\|\boldsymbol{x}(n_1) - \boldsymbol{x}(n_2)\|_2) \ (n_1 \ne n_2).$$

Substituting the latter equation in (5) into (6) yields

$$\frac{d_{\min}}{\sqrt{\ln(\frac{1}{\theta})}} \le \sigma \le \frac{d_{\max}}{\sqrt{\ln(\frac{1}{\theta})}}. \quad (7)$$

Hence, the relation above provides both the lower and upper bounds for the single parameter σ (and replaces the range of the value originally suggested in [16]). In practice, the relation (7) generally holds, unless the distribution of the testing data is dramatically changed from that of the training data.

In summary, exploiting both the relations (5) and (7) leads to the following new parameter setting scheme, which depends upon only a unique radius σ for the construction of the SSKM:

1) Set the value of θ, chosen arbitrarily within the range $0 < \theta < 1$.

2) Compute both the lower and upper bounds $\sigma_{\min}$ and $\sigma_{\max}$ by

$$\sigma_{\min} = \frac{d_{\min}}{\sqrt{\ln(\frac{1}{\theta})}} \text{ and } \sigma_{\max} = \frac{d_{\max}}{\sqrt{\ln(\frac{1}{\theta})}}. \quad (8)$$

3) Choose a unique radius σ within the range

$$\sigma_{\min} \le \sigma \le \sigma_{\max}. \quad (9)$$

4) Construct M distinct SSKM networks by applying the network-growing algorithm [16] as described in II.

IV. Simulation Study

Simulations were conducted, in order to i) validate the hypothesis that the single parameter σ is given by (5) as a function of θ and ii) observe the impact upon the performance by varying σ.

TABLE II. Properties of the Data Sets Used for the Simulation Study

Data set	*Num. training patterns*	*Num. testing patterns*	*Length of each vector*	*Num. Classes M*	*d_{min} by (6)*	*d_{max} by (6)*
OptDigits	1200	400	64	10	1.0	9.3
PenDigits	1200	400	16	10	0.1	5.7
SFS	540	360	256	10	2.4	11.4
ISOLET	1560	390	617	26	4.2	24.9

In the simulation study, four data sets, each extracted from OptDigits, PenDigits, SFS, and ISOLET database, respectively, were used. The OptDigits and PenDigits databases are public handwritten digit databases, whereas the ISOLET is a handwritten letter database, all the three obtained from the University of California at Irvine (UCI) Machine Learning Repository [19]. The SFS is a database for spoken digit recognition available from University College London [20]. For the two UCI (i.e. OptDigits and PenDigits) and SFS, the data sets of the same sizes as those in [16] were used. The properties of the four data sets used in the simulation study are tabulated in Table II.

TABLE III. The Two Parameters that Yield the Same Network Structure and Classification Performance

θ	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
σ (Opt-Digits)	1.5	1.8	2.1	2.4	2.7	3.2	3.8	4.8	7.0
σ (Pen-Digits)	0.3	0.4	0.4	0.5	0.6	0.7	0.8	1.0	1.5
σ (SFS)	2.1	2.5	2.9	3.4	3.9	4.5	5.4	6.8	9.9

A. Validating the Interrelation between σ and θ

For giving an empirical justification of the interrelation between the radius σ and threshold θ parameters described in III, the three data sets OptDigits, PenDigits, and SFS were used. The value of the distance d was first calculated by the former equation in (5), using the same parameter combinations as reported in [16], i.e. (σ=3.8, θ=0.7; d=2.26944) for the OptDigits, (σ=1.0, θ=0.8; d=0.48811) for the PenDigits, and (σ=5.4, θ=0.7; d=3.225) for the SFS. Second, a series of simulations were performed using nine equally-spaced values of θ within the range [0.1, 0.9]. The values of the other parameter σ were calculated for each of the three data sets as shown in Table III, using (5) with varying the threshold θ within this range.

For all the combinations of θ and σ in Table III, an identical SSKM network structure (i.e. indicated by both the numbers of RBFs N_{RBF} and LWs N_{LW} generated; OptDigits: N_{RBF}=771, N_{LW}=2764, PenDigits: N_{RBF}=735, N_{LW}=2125, and SFS: N_{RBF}=519, N_{LW}=12, respectively), as well as exactly the same classification performance, was obtained for all the three data sets (OptDigits: 96.2%, PenDigits: 96.8%, and SFS: 97.5%, respectively) by applying the network-growing algorithm reviewed in II.

From these observations, it is concluded that the parameters σ and θ are interrelated and thus that only tuning of either σ or θ is required, while the remaining one fixed arbitrarily, during the construction phase of the SSKM.

TABLE IV. Simulation Results of the RAN and SSKM

Data set	***RAN***		***SSKM (Sigma: 8)***	
	C. Rate (%)	*Num. RBFs*	*C. Rate (%)*	*Num. RBFs*
OptDigits	95.3	844	96.3	870
PenDigits	96.0	454	95.5	236
SFS	98.1	516	97.8	486
ISOLET	96.4	1392	91.3	1116

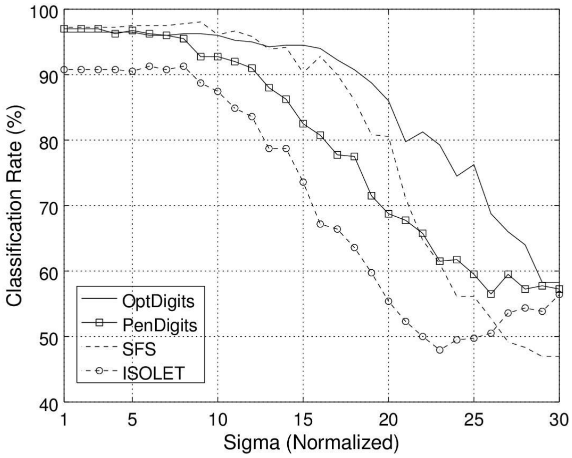

Fig. 1. Classification rates with varying σ obtained using SSKM.

TABLE V. SIMULATION RESULTS OF THE SVMS

Data set	*c*=0.1		*c*=10		*c*=100	
	C. Rate (%)	***Num. SVs***	***C. Rate (%)***	***Num. SVs***	***C. Rate (%)***	***Num. SVs***
OptDigits	90.3	2609	97.0	774	96.3	728
PenDigits	89.0	2479	96.3	434	96.3	298
SFS	94.2	1354	98.3	875	98.3	865
ISOLET	90.8	4086	97.2	2611	96.9	2584

B. Impact Upon the Performance by Varying σ

Another set of simulations was run, in order to observe the impact upon the performance of the SSKM networks by varying σ, while fixing θ(=0.5). For this observation, both the lower and upper bounds of the single parameter σ were computed for all the four data sets by applying (9), given the respective values of d_{min} and d_{max} as in Table II. In practice, it is, however, considered that the radius chosen close to the upper bound yields only sparse coverage of the data space. Therefore, σ was varied between σ_{min} and $0.6\sigma_{max}$, and the performance was obtained for each of the 30 equally-spaced values of σ within this range.

Figures 1-3 show the overall classification rates, the total number of RBFs generated, and the value μ (Myu), showing the number of LWs established per RBF, i.e.

$$\mu = \frac{\{\text{total num. LWs}\}}{\{\text{total num. RBFs}\}}, \quad (10)$$

respectively. In these figures, each value in the x-axis corresponds to a normalized value of σ (i.e. Sigma: 1 corresponds to σ_{min}, whereas Sigma: 30 to $0.6\sigma_{max}$). We then find similar tendencies for all the four data sets in the plots: i) the overall classification rates tend to degrade and start to drop quickly at around Sigma: 10, ii) the number of RBFs accommodated within the SSKM networks falls rapidly at around Sigmas: 2-4, and iii) there is always a peak of the value μ somewhere within the range [σ_{min}, $0.6\sigma_{max}$]. From these observations, an appropriate choice of the unique radius which

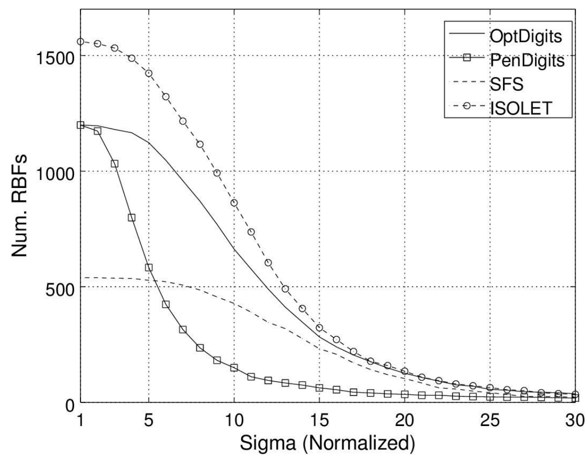

Fig. 3. Number of the RBFs generated with varying σ obtained using SSKM.

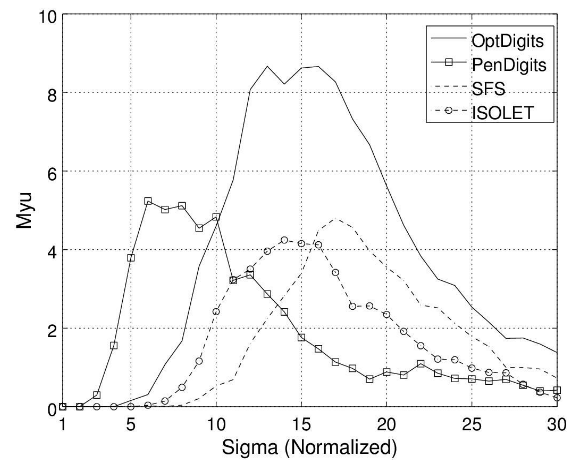

Fig. 2. The value μ with varying σ obtained using SSKM.

yields a good trade-off between the classification rate and network size can be, as a rule of thumb, given by $\sigma = \gamma\sigma_{max}$, where γ is a scalar constant set within the range [0.1, 0.3]; for instance, the results obtained at Sigma: 8 are shown in the right part of Table IV.

A performance comparison was made, using the two benchmark methods of RAN [8] and support vector machines (SVMs) [17][18]. RAN was chosen as a baseline method, since the algorithm was found to be relatively straightforward to implement with low computational complexity, compared with other constructive approaches in Table I. SVMs with Gaussian kernels were used as another baseline, due to their popularity in the machine learning related areas of study for its robust classification capability and a relatively smaller number of parameters to tune (i.e. two).

For the RAN, the six parameters α=0.05, ε=0.2, δ_{min}=0.07, δ_{max}=0.7, κ=0.87, and τ=2 were chosen the same as those used in [16] (the values of which are also almost the same as those reported in the Platt's work [8]).

For the SVM, since each of the four dataset sets contains more than two classes, a commonly-used `one-against-all' approach (see e.g. [18]) was employed during the training phase (i.e. for an M-class classification task, a total of M SVMs, each responsible for a single class, were trained and then used for the testing). For training the SVMs, the same setting as those in [16] for the kernel width (or radius) σ (=7.9 : OptDigits, 4.8 : PenDigits, and 9.9 : SFS) was used. For the ISOLET, σ=20.0 was used. The other parameter, i.e. the bound for the weight coefficients, was varied: c=0.1, 10, and 100 for all the four data sets. Tables IV (in the left part) and V summarize the simulation results obtained by RAN and SVMs, respectively.

As shown in these tables, the classification rates obtained using the SSKM (i.e. with the setting Sigma: 8) were almost comparable to those using the RAN and SVMs for the three data sets of OptDigits, PenDigits and SFS, while the SSKM with fewer numbers of RBFs than those obtained using the RAN and SVMs with all the three different settings of c, for the PenDigits, SFS, and ISOLET data sets. For the ISOLET, both the SVMs and RAN yielded a higher classification rate than that obtained using the SSKM. The higher rates achieved by the former two methods can be ascribed to a further optimization/introduction of other network parameters during the construction phase; i.e. the use of individual radii values for the RBFs and their adaptation, as well as that of the variable weight coefficients between the RBFs and output units, in the case of RAN, while generating a larger number of Gaussian kernels in the case of the SVMs. It is considered that the parameter optimization in these two methods greatly contributed to compensate for the weak coverage of the data space caused due to an insufficient number of the pattern data given (i.e. a relatively small number of training patterns against higher-dimensional data, compared to the other three data sets), albeit at the expense of an extra (and, in practice, large) amount of computational complexity during the training phase.

V. Conclusion

A new parameter setting scheme for the previously proposed SSKM network construction algorithm [16], which essentially requires the tuning of only a single parameter i.e. the value of the radius unique to all the RBFs, has been proposed in this paper. Both the lower and upper bounds of the radius have also been derived. Then, it has been shown that incorporating these two properties into the SSKM construction enables an efficient design of the RBF based pattern classifiers, which is quite attractive in practical situations. In the simulation study, it was empirically shown that an appropriate choice of the radius can be made, based upon the value of the upper bound for the radius multiplied by a scaling factor. In practice, an optimal parameter choice can be made via performing an n-fold cross-validation, which, however, involves multiple presentation of the whole training data as in the constructive methods [3][6] or SVMs (i.e. those trained by the one-against-all approach) but with low computational complexity, unlike these approaches.

Future work should be directed to the proposal of a better constructive scheme in order to cope with adverse conditions such as the ISOLET case in the simulation study in this paper, while requiring as small the number of the network configuration parameters to tune as possible.

References

[1] D. S. Broomhead and D. Lowe, "Multivariate functional interpolation and adaptive networks," *Complex Systems* vol. 2, pp. 321-355, 1988.

[2] D. F. Specht, "Probabilistic neural networks," *Neural Networks*, vol. 3, no. 1, pp. 109–118, 1990.

[3] M. R. Berthold and J. Diamond, "Constructive training of probabilistic neural networks," *Neurocomputing*, pp. 167–183, 1998.

[4] T. Hoya and A. G. Constantinides, "An heuristic pattern correction scheme for GRNNs and its application to speech recognition," *Proc. IEEE Workshop - Neural Networks for Signal Processing VIII*, pp. 351-359, 1998.

[5] T. Hoya, "Modeling the notions of intuition and consciousness by hierarchically arranged generalized regression neural networks," *Proc. NOLTA-2001*, pp. 403-406, 2001.

[6] X. Jin, D. Srinivasan, and R. L. Cleu, "Classification of freeway traffic patterns for incident detection using constructive probabilistic neural networks," *IEEE Trans. Neural Networks*, vol. 12, no. 5, pp. 1173-1187, 2001.

[7] M. R. Heinen and P. M. Engel, "IPNN: An incremental probabilistic neural network for function approximation and regression tasks," *Proc. 11th Brazilian Symposium on Neural Networks*, pp. 25-30, 2010.

[8] J. Platt, "A resource allocating network for function interpolation," *Neural Computation*, vol. 3, no. 2, pp. 213–225, 1991.

[9] V. Kadirkamanathan and M. Niranjan, "A function estimation approach to sequential learning with neural networks," *Neural Computation*, vol. 5, no. 6, pp. 954–975, 1993.

[10] Y. Lu, N. Sundararajan, and P. Saratchandran, "A sequential learning scheme for function approximation using minimal radial basis function neural networks," *Neural Computation*, vol. 9, no. 2, pp. 461–478, 1997.

[11] S. Chen, C. F. N. Cowan, and P. M. Grant, "Orthogonal least squares learning algorithm for radial basis function networks," *IEEE Trans. Neural Networks*, vol. 2, no. 2, pp. 302-309, 1991.

[12] L. Zhang, K. Li, H. He, and G. W. Irwin, "A new discrete-continuous algorithm for radial basis function networks construction," *IEEE Trans. Neural Networks and Learning Systems*, vol. 24, no. 11, pp. 1785-1798, 2013.

[13] S. A. Billings and G. L. Zheng, "Radial basis function network configuration using genetic algorithms," *Neural Networks*, vol. 8, no. 6, pp. 877-890, 1995.

[14] S. Chen, X. Hong, and C. J. Harris, "Particle swarm optimization aided orthogonal forward regression for united data modeling," *IEEE Trans. Evolutionary Computation*, vol. 14, no. 4, pp. 477-499, 2010.

[15] J. Deng, K. Li, G. W. Irwin, and M. Fei, "Two-stage RBF network construction based on particle swarm optimization," *Trans. of Inst. Meas. Control*, vol. 35, no. 1, pp. 25-33, 2013.

[16] T. Hoya and Y. Washizawa, "Simultaneous pattern classification and multi-domain association using self-structuring kernel memory networks," *IEEE Trans. Neural Networks*, vol. 18, no. 3, pp. 732-744, 2007.

[17] C. Cortes and V. Vapnik, "Support-vector network," *Machine Learning*, vol. 20, pp. 273–297, 1995.

[18] L. Bottou, C. Cortes, J. S. Denker, H. Drucker, I. Guyon, L. Jackel, Y. LeCun, U. A. Muller, E. Sackinger, P. Simard, and V. Vapnik, "Comparison of classifier methods: a case study in handwriting digit recognition," *Proc. Int. Conf. Pattern Recognition*, pp. 77–87, 1994.

[19] E. Alpaydin and C. Kaynak, "UCI machine learning," Univ. California Irvine, Irvine, CA., 1998 [Online]. Available: www.ics.uci.edu/~mlearn

[20] M. Huckvale, "Speech filing system vs3.0 -- computer tools for speech research," London, U.K.: Univ. College, 1996.