A simple clustering technique to extract subsets of data for function approximation

Dulakshi Santhusitha Kumari Karunasingha and Shie-Yui Liong

ABSTRACT

A simple clustering method is proposed for extracting representative subsets from lengthy data sets. The main purpose of the extracted subset of data is to use it to build prediction models (of the form of approximating functional relationships) instead of using the entire large data set. Such smaller subsets of data are often required in exploratory analysis stages of studies that involve resource consuming investigations. A few recent studies have used a subtractive clustering method (SCM) for such data extraction, in the absence of clustering methods for function approximation. SCM, however, requires several parameters to be specified. This study proposes a clustering method, which requires only a single parameter to be specified, yet it is shown to be as effective as the SCM. A method to find suitable values for the parameter is also proposed. Due to having only a single parameter, using the proposed clustering method is shown to be orders of magnitudes more efficient than using SCM. The effectiveness of the proposed method is demonstrated on phase space prediction of three univariate time series and prediction of two multivariate data sets. Some drawbacks of SCM when applied for data extraction are identified, and the proposed method is shown to be a solution for them.

Key words | clustering, data sub sets, function approximation, fuzzy models, phase space, time series

Dulakshi Santhusitha Kumari Karunasingha
(corresponding author)
Department of Engineering Mathematics,
Faculty of Engineering,
University of Peradeniya,
Peradeniya,
Sri Lanka
E-mail: *dulakshi@yahoo.com*

Shie-Yui Liong
Tropical Marine Science Institute, National
University of Singapore,
Singapore 119223,
Singapore

INTRODUCTION

Large data sets can often cause difficulties in practice as well as in analysis, where it may require lots of computational resources such as memory and time. Although the computational power keeps on increasing, the complexity of the problems analyzed also increases. For example, applications that involve evolutionary methods may contain large numbers of resource intensive evaluations that require considerable computational time (e.g., Karunasingha *et al.* 2011). In such cases the usual practice is to use a randomly chosen smaller subset of data, at least, in the exploratory stage. However, such a random subset is not necessarily representative of the total data or the system under investigation. Therefore, a methodology that can extract a smaller set of most representative data from a large data record is highly desirable.

Clustering, a means of finding patterns in the data (Ghahramani 2004), may be used to extract a smaller set of data from large data records. The choice of the 'best' clustering method for a given application is problem dependent (e.g., Xu & Wunsch 2005; Luxburg *et al.* 2012) and the simplest method that serves the purpose is always preferred. Consequently, the parameters of a clustering technique are also not generally based on theoretical knowledge and the appropriate values for them are, most times, determined by trial and error so that the final goal is satisfactorily achieved.

Most of the applications of clustering have been on classification problems. Applications of clustering techniques in function approximation problems are few (Chiu 1994; Yager & Filev 1994a, 1994b; Kreinovich & Yam 2000), and all those applications are centered on data extraction for local function approximation purposes, specifically for fuzzy model generation. Recently, the need for different

doi: 10.2166/hydro.2015.065

clustering methods for function approximation has been identified and an algorithm has been proposed by Gonzalez *et al.* (2002). However, some weaknesses of this method were identified and several improved versions have been proposed (Guillen *et al.* 2007; Pomares *et al.* 2012). Nevertheless, all those algorithms have been tested only on the problem of selection of initial centers for radial basis function neural networks (RBFN), which is a crucial issue in RBFN (Lin & Chen 2005). A few studies (Karunasingha & Liong 2003; Doan *et al.* 2005) have adopted a subtractive clustering method (SCM) (Chiu 1994) for data extraction from long hydrological data records, including chaotic hydrological time series. They demonstrated the effectiveness of the extracted data using neural networks, fuzzy models, and local phase space prediction methods. To the best of the authors' knowledge, the above applications are the only studies that have used a clustering technique to extract a subset of data for function approximation purposes, and demonstrated its effectiveness on several prediction methods. SCM used in the above studies, however, has four free parameters, and requires a great deal of computational effort to determine the optimal values for these four parameters.

This study proposes a new, simpler clustering technique, for function approximation, with only one single parameter of which the optimal value can be easily found with much less computational effort compared to finding four optimal parameters in SCM. Its effectiveness in extracting system representative smaller subsets from total data sets is also demonstrated. It should be noted that in this paper a subset of data, which provides similar prediction performance as the total data set, is regarded as a 'representative data set' or a 'system representative subset' of the total data records. The method is developed with phase space prediction of chaotic time series in mind; however, applications on a synthetic benchmark data set and Mekong river flow prediction show that the method is effective on other multivariate data sets as well. Three univariate time series, namely, a noisy Lorenz series, Mississippi River flow series, and Wabash River flow series are also used for demonstration. After a brief explanation on chaotic time series analysis in the next section, the new clustering technique is introduced. This is followed by its application, results, and the conclusions of the study.

CHAOTIC TIME SERIES ANALYSIS

Short-term prediction of hydrological/meteorological time series using deterministic chaotic dynamical systems approach has become an alternative to the conventional linear stochastic approach (e.g., Sivakumar 2004; Karunasinghe & Liong 2006; Yu & Liong 2007). In chaotic time series analysis, a prediction model for a dynamical system is directly constructed from the past records of a single variable observed as a time series, say, $x_1,\ x_2\ \dots x_n$. Using time delay coordinate method (e.g., Packard *et al.* 1980; Takens 1981), the dynamics of the time series can be embedded in an m dimensional space called phase space. The m dimensional vectors $\boldsymbol{X}_i$, given by

$$\boldsymbol{X}_i = \{x_i,\ x_{i-\tau},\ \dots x_{i-(m-1)\tau}\} \tag{1}$$

where $i = (m-1)\tau + 1,\ (m-1)\tau + 2,\ \dots,\ n$ and τ is the time delay are called phase space vectors. The paths traced by this vector series $\boldsymbol{X}_i$ are called the trajectories. There are standard methods and inverse methods to determine the phase space parameters embedding dimension (m) and time delay (τ).

In phase space prediction, the basic idea is to set a functional relationship between the current state $\boldsymbol{X}_t$ and future state $\boldsymbol{X}_{t+T}$ in the form

$$\boldsymbol{X}_{t+T} = f_T(\boldsymbol{X}_t) \tag{2}$$

where T is referred to as lead time or prediction horizon. At time t, for an observation value x_t, the current state of the system is $\boldsymbol{X}_t$, where $\boldsymbol{X}_t = (x_t,\ x_{t-\tau},\ \dots x_{t-(m-1)\tau})$ and similarly the future state at time $t+T$ is $\boldsymbol{X}_{t+T}$. In the functional relationship (Equation (2)), we are only interested in forecasting the first component, x_{i+T} of $\boldsymbol{X}_{i+T}$, the search is limited to a map $F_T{:}R^m \Rightarrow R$ (see Equation (3)), which interpolates the pairs $(\boldsymbol{X}_i,\ x_{i+T})$ instead of a function $F_T{:}R^m \Rightarrow R^m$. For a chaotic system, the predictor F_T that approximates f_T is necessarily nonlinear. Multilayer perceptron (MLP) models (Haykin 1999) with m-dimensional phase space vectors $\boldsymbol{X}_i$ as the inputs and the scalars x_{i+T} as the outputs are used in this study to approximate a map $F_T{:}R^m \Rightarrow R$, valid over the entire phase space, i.e., a global fit. Note that Equation (3) is a special case of the more general function approximation problem of approximating function f in the map $y = f(\boldsymbol{x})$, using the available data $(\boldsymbol{x}_i, y_i)$, $i = 1, 2, \ldots, N$ where $\boldsymbol{x}_i \in \boldsymbol{X}$, $\boldsymbol{X} \subseteq R^m$ is the m-dimensional input space (independent variables) and $y_i \in Y$, where Y is the output space (dependent variable)

$$x_{t+T} = f_T(\boldsymbol{X}_t) \tag{3}$$

NEW CLUSTERING METHOD

In deriving relationships using lengthy data records redundancy of data can occur due to at least two reasons: (1) it is possible that not all the points are necessary to represent a certain relationship (e.g., two points suffice to represent a linear relationship of a single input/single output system); (2) also, there can be points that are identical and/or that are closer than noise level, which do not contain any distinct information. The algorithm of the present clustering method is based on the following observation. For example, consider a few trajectories of a Lorenz dynamical system (Lorenz 1963), the most widely used benchmark chaotic system, lying close to each other. If the time series is noisy, the points may take positions deviated from their true states. What this means is for a noisy time series one cannot distinguish the trajectories closer than the noise level. Since the points closer than the noise level do not provide any distinct information, one may choose one point to represent a neighborhood roughly of the order of the noise level. An overview of the proposed clustering algorithm can be given as follows.

The algorithm uses both a density measure and a distance measure to select the cluster centers. The density measure, similar to the one in SCM, is such that the points that are closely surrounded by other points have a higher density and are more likely to be chosen as cluster centers. The distance measure defines the neighborhood size or the minimum distance between two cluster centers. The present algorithm ensures that every point in the original data set is either a cluster center or close to a center by a distance $\leq d$, where d is the distance measure of the method.

Practically all clustering techniques, based on classification point of view, treat isolated points (points that are far from other points) as outliers. However, in analysis of real world systems (e.g., river flow), such points often represent extreme and perhaps infrequent events (e.g., flood flows), which are important. The points lying in the less dense areas of data space are not considered as outliers in the proposed clustering method. This is how the present method is radically different from other clustering methods. The selection of points lying far from other points is achieved in the algorithm in a way that an additional parameter determining a stopping criterion is eliminated.

Algorithm

The simple clustering algorithm can be given as follows. Consider N points, $\boldsymbol{X}_i$, $i = 1, 2, \ldots N$, of dimension m. Assume that these points have been normalized so that they lie in a unit hypercube. This makes it possible for the only parameter of this method, d, defining the neighborhood, to be specified without using the domain specific knowledge.

Step 1: Calculate a density measure for each point $\boldsymbol{X}_i$. A Gaussian 'influence function', which indicates the influence of each data point on a certain data point, is used as the density measure,

$$P_i = \sum_{j=1}^{N} \mathrm{e}^{-(\|\boldsymbol{X}_i - \boldsymbol{X}_j\|^2/d^2)} \tag{4}$$

where P_i = density measure of point i and d = radius of neighborhood. The density P is higher for points closely surrounded by other points and lower for the points with less neighboring points. Theoretically, d may vary from 0 to $\sqrt{m}$. Practically, d may take small positive values generally less than 1.

Step 2: Select the point with the highest density as the first cluster center.

Step 3: Set the density measure of the selected cluster center and the density of points closer than d from the selected cluster center to zero.

Step 4: Select the point with the next highest density measure. If its density measure is greater than zero select the point as a cluster center and go to step 3. Or stop.

Note that no additional parameter is required as a stopping criterion.

In Equation (4), the radius within which the points contribute significantly to the potential P_i for the calculation of P_i is approximately $2d$ while the neighborhood size (or the minimum distance between two cluster centers) is d.

APPLICATION OF THE PROPOSED CLUSTERING METHOD

Data used

The effectiveness of the proposed clustering algorithm to extract a compact set of system representative patterns from a fairly large set of patterns is demonstrated on a noisy chaotic Lorenz time series, two univariate daily mean river flow time series, one synthetic multivariate data set, and one multivariate river flow time series. The river flow series are: (1) the Mississippi river at Vicksburg (1975–1993) with mean flow around 18,500 m^3/s; (2) Wabash River at Mt Carmel (1960–1978) with mean flow around 750 m^3/s; and (3) Mekong River flow data, respectively. The first two univariate river flow series are available in the US Geological Survey website (http://water.usgs.gov/).

Lorenz model is given by the following three ordinary differential equations:

$$\dot{x} = \sigma(y - x); \quad \dot{y} = -xz + \gamma x - y; \quad \dot{z} = xy - bz \tag{5}$$

When standard parameter values $\sigma = 16$, $b = 4$, and $\gamma = 45.92$ are used, the orbits of the Lorenz system reside on a geometric object of dimension 2.06 (approximately) and exhibit non-periodic, chaotic motion (Abarbanel 1996). The *x(t)* component is solved from the above equations by fourth order Runge–Kutta method with a time step of $\Delta t = 0.001$. A zero mean Gaussian noise with standard deviation 5% of the standard deviation of the noise-free *x(t)* series was added to the *x(t)* series in order to obtain the noisy Lorenz series. The standard deviation of noise-free Lorenz series is 12.68 and the standard deviation of noise is 0.63. The Mississippi River is one of the world's largest river systems of about 3,705 km in length and a river basin of around 3.2 million km^2. The average amount of water discharged to the Gulf of Mexico is about 18,500 m^3/s and a standard deviation of 9,728 m^3/s. The minimum and maximum flow values are about 3,900 and 52,100 m^3/s, respectively. The daily flow time series of the Mississippi River measured at Vicksburg, station number 07289000 (hydrologic region 8 of USGS) for the period from 1 January 1975 to 31 December 1993 is used in this study. The Wabash River is 765 km long, flowing southwest from northwest Ohio. The basin area is approximately 85,750 km^2. The Wabash River has a mean flow rate of about 750 m^3/s and a standard deviation of around 792 m^3/s. The minimum and maximum flow values are about 48 and 7023 m^3/s, respectively. The daily river flow measured at Mt Carmel Station, station number 03377500 (hydrologic region 5 of USGS), for the period from 1 January 1960 to 31 December 1978 is used in this study.

The impedance of a certain type of an alternating current series circuit can be expressed by the form given in Equation (6) (Friedman 1991),

$$f(x) = \left(x_1^2 + \left(x_2x_3 - \frac{1}{x_2x_4}\right)^2\right)^{1/2} \tag{6}$$

where x_1, x_2, x_3 and x_4 are resistance, angular frequency, inductance, and capacitance, respectively. This function has been widely used for benchmarking regression problems in different applications (e.g., Vapnik 1999; Roth 2004; Martinez-Estudillo *et al.* 2006). In this study this function is used to generate a multivariate data set. It should be noted that this is not a time series data set. With the domain $(x_1 \in [0,\ 100],\ \ x_2 \in [40\pi,\ 560\pi],\ \ x_3 \in [0,\ 1],\ \ x_4 \in [1,\ 11])$ that has been used in the said studies, a data set of 7,000 records was randomly generated using uniform distributions. Then, to simulate real life situations, for training data, both the inputs (i.e., x_1, x_2, x_3 and x_4) and the output $f(x)$ were contaminated with zero mean Gaussian noises with standard deviations of 5% of the standard deviation of each variable. These noise added data were used in the study.

The Mekong River is perhaps the most important and most controversial river in Asia. It originates in China and runs through Myanmar, Lao, Thailand, Cambodia, and finally, Vietnam before it drains into the South China Sea. The river has a length of about 4,620 km and drains a combined area of 795,500 km^2. The middle reach of the river has 10 gauging stations (Karunasingha *et al.* 2011) named Chiang Saen (most upstream), Luang Prabang, Chiang Khan, Pa Mong Dam Site, Vientiane, Nong Khai, Nakhon Phanom, Mukdahan, Khong Chiam, and Pakse (most downstream), respectively. In this study, daily river flows measured at the above referenced ten stations of the Mekong River from April 1972 to December 1989 and January 1989 to December 1994 are used (IDI, 1960–1994: Water Series No. 10). The problem of predicting 1 day ahead river flow at Pakse as a function of the flow values of nine upstream stations and Pakse itself was formulated as given in Equation (7) where $ST_i(t)$ is the flow measured at ith upstream station on tth day.

$$ST_{\text{Pakse}}(t+1) = f(ST_1(t),\ ST_2(t),\ \ldots ST_9(t),\ ST_{\text{Pakse}}(t)) \qquad (7)$$

Methodology

Six thousand data points from noisy Lorenz series and a fairly long set of river flow data, approximately 6,900 data points from the Mississippi and Wabash River flow time series, and approximately 6,470 data points from the Mekong River flow series are used in the analysis. Each of these time series is divided into three separate parts: a training set, test set, and a validation set. Based on previous experience (e.g., Yu *et al.* 2004; Liong *et al.* 2005), in each river flow series, the first 5,480 data points (approximately) are used for the training set, the next 730 data points as the test set, and the last 730 data points as the validation set. For the Lorenz series, the first 4,800 values are used for training, the next 600 values are used for testing, and the last 600 values are used for validation. In the synthetic multivariate data set, 5,600 records for training, 400 for testing, and 1,000 records for validation were used. For the multivariate Mekong River flow time series, 5,740, 365, and 365 records were used, respectively, for training, test, and validation sets.

The effectiveness of the data extraction is evaluated by comparing the one step ahead prediction performance of models trained with extracted smaller subsets of training data with that of a model trained with entire training data set. First, the phase space vectors are constructed with phase space parameters, (m, τ) pairs, (10, 3), (3, 1), (5, 1), respectively, for Lorenz, Mississippi and Wabash series based on previous experience (Karunasinghe & Liong 2006). Second, for each series, a MLP prediction model that approximates the function F_T in Equation (3), where $T = 1$, is derived using the entire training data patterns and the test data patterns. Here, an input/output pair ($\boldsymbol{X}_i$, x_{i+T}) is referred to as a pattern. The prediction errors of this model on the validation set are calculated. Third, the new clustering method is applied on the normalized training data patterns and a smaller subset of patterns is extracted (see Figure 1). The smaller set of input/output patterns and the test set are then used to train an MLP prediction model such as F_T in Equation (3). This model is then used for prediction of the validation set and its performance on the validation set is compared with that of the model trained with the entire data set. The whole procedure is schematically expressed in Figure 2. The study used MLPs with a single hidden layer. Logistic sigmoid transfer function was used for all hidden neurons. Linear transfer function was used in the output neurons. The numbers of hidden neurons were 25, 25, 25, 10, and 3 for Lorenz, Mississippi, Wabash, Friedman, and Mekong data sets, respectively. Since the phase space parameter selection and the way of model training does not affect the results of the current study such details are not presented here and interested readers are referred to Karunasinghe & Liong (2006) for more details.

For illustrative purposes, when applying the proposed clustering method, subsets of different sizes were obtained

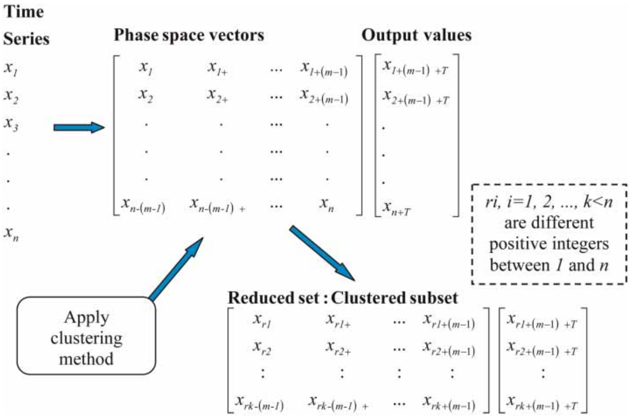
Time Series
x_1
x_2
x_3
.
.
.
x_n
Phase space vectors
x_1 x_{1+} ... $x_{1+(m-1)}$
x_2 x_{2+} ... $x_{2+(m-1)}$
$x_{n-(m-1)}$ $x_{n-(m-1)+}$ ... x_n
Output values
$x_{1+(m-1)+T}$
$x_{2+(m-1)+T}$
x_{n+T}
ri, i=1, 2, ..., k<n are different positive integers between 1 and n
Apply clustering method
Reduced set : Clustered subset
x_{r1} x_{r1+} ... $x_{r1+(m-1)}$
x_{r2} x_{r2+} ... $x_{r2+(m-1)}$
$x_{rk-(m-1)}$ $x_{rk-(m-1)+}$ ... $x_{rk+(m-1)}$
$x_{r1+(m-1)+T}$
$x_{r2+(m-1)+T}$
$x_{rk+(m-1)+T}$

Figure 1 | Overview of the data extraction procedure.

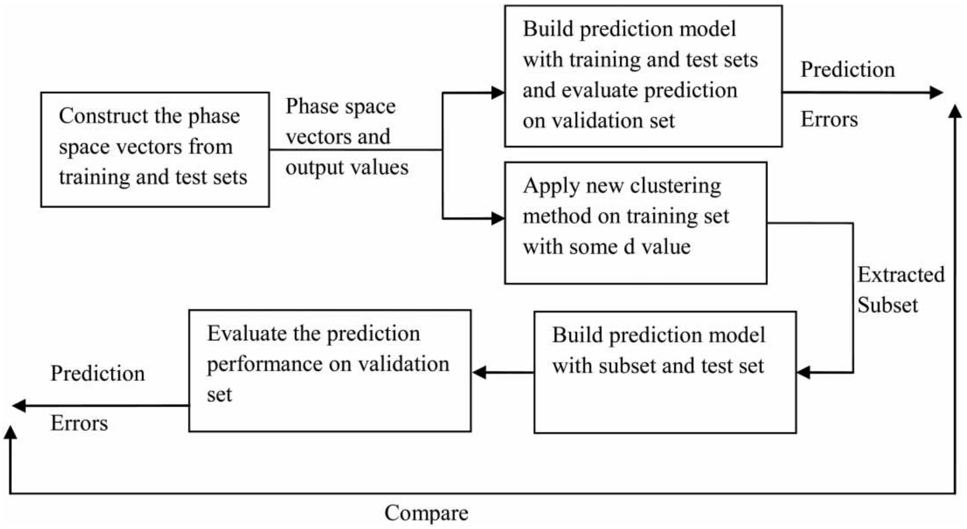
Construct the phase space vectors from training and test sets
Phase space vectors and output values
Build prediction model with training and test sets and evaluate prediction on validation set
Prediction
Errors
Apply new clustering method on training set with some d value
Extracted Subset
Build prediction model with subset and test set
Evaluate the prediction performance on validation set
Prediction
Errors
Compare

Figure 2 | Schematic diagram of the whole procedure.

by considering a series of different d values as follows. A higher value of d extracts smaller number of patterns and vice versa. When the number of patterns is too small, the resulting prediction performance is anticipated to be poor. An optimal d value which gives a balance between the number of patterns and prediction performance is preferred. Noting that d may be related to the noise level, and for moderate noise levels the effective values of d may take values close to zero, this study started off with three trial values for d, 0.001, 0.1, and 0.5, and used interval bisection strategy (similar to bisection method for root finding, which repeatedly bisects an interval and then selects a subinterval in which a desired solution lies for further processing) to identify a suitable range for d (values which give low numbers of patterns and satisfactory predictions) to be explored. Once a suitable range for d is determined, the range is evenly subdivided into approximately 50 points. The subsets resulting from these d values are used for building MLP models and prediction.

Prediction performance is evaluated in terms of four error measures: (1) two relative error measures, the normalized root mean square error (NRMSE) given in Equation (8a) and modified coefficient of efficiency (MCE) recommended in Legates & McCabe (1999) given in Equation (8b); (2) two absolute error measures, mean absolute error (MAE) given in Equation (9a) and root mean square error (RMSE) given in Equation (9b).

$$\mathrm{NRMSE} = \sqrt{\frac{\sum_{i=1}^{n}(x_i - \hat{x}_i)^2}{\sum_{i=1}^{n}(x_i - \bar{x})^2}} \tag{8a}$$

$$\mathrm{MCE} = 1 - \frac{\sum_{i=1}^{n}|x_i - \hat{x}_i|}{\sum_{i=1}^{n}|x_i - \bar{x}|} \tag{8b}$$

In Equations (8a), (8b), (9a), and (9b), x_i is the observed value, $\hat{x}_i$ is the predicted value, n is the number of points predicted, and $\bar{x}$ is the average value of the time series. A zero/one value in NRMSE/MCE indicates a perfect prediction, while a value greater than one/less than zero for NRMSE/MCE indicates that the predictions are no better than using the average value of the time series. A zero value in MAE and RMSE represents a perfect prediction.

$$\mathrm{MAE} = \frac{\sum_{i=1}^{n}|x_i - \hat{x}_i|}{n} \tag{9a}$$

$$\mathrm{RMSE} = \sqrt{\frac{\sum_{i=1}^{n}(x_i - \hat{x}_i)}{n}} \tag{9b}$$

For the comparative study, subsets were obtained with SCM and their prediction performances were compared with those of subsets obtained with the proposed clustering method. SCM was used for comparison purposes for several reasons. The study uses SCM as the only method, to the best of the authors' knowledge, that has been tested for function approximation purposes using several prediction methods. Cluster centers of SCM are a subset of the original data. Other widely used techniques, such as k-means clustering, produces centers which are not necessarily data points. Such 'artificial' points may not be acceptable in function approximation problems. As such, investigating whether and how the existing clustering methods, that have been developed for the purpose of classification, could be incorporated for function approximation require further studies that are beyond the scope of this study. SCM has four parameters, namely: (1) radius of influence (R) of a cluster center; (2) the accept ratio (AR), that determines whether data point will be accepted as a cluster center; (3) the reject ratio (RR), that determines whether a data point will be rejected as a cluster center; (4) squash factor (SF), that determines the neighborhood of a cluster center within which the existence of other cluster centers are discouraged. It is tedious to use a trial and error approach for finding optimal values for them. Therefore, a micro-genetic algorithm (mGA) was used to find suitable values for these parameters, following Doan *et al.* (2005). The procedure can be explained as follows. mGA was used to generate SCM parameter sets. Corresponding to these parameter sets, clustered subsets were obtained from the training data set. Then, these subsets were used to make the prediction on the test set using local averaging nonlinear prediction (NLP) method (Karunasinghe & Liong 2006) to approximate functions, *f*, in Equations (3), (6), and (7). The prediction accuracy on the test set was used as the fitness indicator in mGA. The procedure was repeated until the stopping criterion was met. The maximum number of evaluations (i.e., number of SCM parameter sets tested) was limited to 1,000, based on previous experience, and was used as the stopping criterion. The NLP, instead of MLP, was used for fitness evaluation in mGA to reduce the time taken to find optimal SCM parameters. For more details of this procedure readers are referred to Doan *et al.* (2005).

Information content in clustered data

The subsets obtained by the proposed method were tested for information content compared to those of random subsets of similar sizes. The quadratic Renyi entropy (Renyi 1961) is used for this purpose. Entropy is calculated for clustered subsets and randomly chosen subsets of the same size. Entropy is a measure of the system randomization. The larger the entropy the more information involved in the sample and better the randomization of the sample is (Jiang *et al.* 2008). For a continuous random variable X with probability density function $p(x)$ the expression for quadratic Renyi entropy is given by Equation (10)

$$H_{R2} = -\log\int p^2(x)\mathrm{d}x \tag{10}$$

Employing nonparametric density estimation using Gaussian kernel, $\int p^2(x)\mathrm{d}x$ can be estimated using the sample points $x_1,\ _2,\ \ldots,\ x_N$ as in Equation (11) (Girolami 2002)

$$\int p^2(x)\mathrm{d}x \approx \int \hat{p}^2(x)\mathrm{d}x = \frac{1}{N^2}\sum_{i=1}^{N}\sum_{j=1}^{N}K(x_i, x_j) \tag{11}$$

where

$$K(x, y) = \exp\left(-\frac{\|\mathbf{x}-\mathbf{y}\|^2}{2\sigma^2}\right), \tag{12}$$

and σ is the width of the Gaussian kernel. Then, the quadratic Renyi entropy is given by Equation (13):

$$H_{R2} = -\log\left(\frac{1}{N^2}\sum_{i=1}^{N}\sum_{j=1}^{N}K(\mathbf{x}_i, \mathbf{x}_j)\right) \tag{13}$$

The width of the Gaussian kernel (also called the kernel size or bandwidth) in Equation (12) is a free parameter that needs to be selected by the user. Therefore, the resulting values of entropy depend on the kernel size selected and they have little absolute meaning, but they gauge performance in a relative sense when comparing data generated with the same set of parameters (Principe 2010). There is no definite criterion to choose a value for σ. Methods available to find an optimal value for σ include cross-validation methods, bootstrapping methods, reference rules, etc. (Brabanter *et al.* 2010). This study used the Silverman's rule given in Equation (14), which was found to be sufficient for many applications by Principe (2010). In Equation (14), N is the number of samples, d is the data dimensionality, and σ_X is the data standard deviation.

$$\sigma_{\mathrm{opt}} = \sigma_X (4N^{-1}(2d+1)^{-1})^{1/(d+4)} \tag{14}$$

RESULTS

Effectiveness of the extracted sets is evaluated by comparing their prediction performance with that of using the total data set. The prediction errors on the validation sets using the models trained with entire training data are given in Table 1. The prediction performance of reduced data sets shows similar trends in terms of NRMSE, MCE, MAE, and RMSE. The prediction performance of the models trained with the extracted data subsets with the number of patterns extracted are, therefore, shown in Figure 3(a) and 3(b) in terms of MAE only for 5% noisy Lorenz series and Wabash river flow time series, respectively. Figure 4(a) shows the prediction performance of extracted subsets of the Mississippi series from 50 different d values of the proposed clustering method and the selected best solutions, out of 1,000 evaluations with subtractive clustering method (Chiu 1994) (following the procedure of Doan *et al.* (2005) where mGA was used to optimize the parameters), superimposed. In the figures, the prediction error in using the total training data set is shown as a solid line. 105% of the error, i.e., 5% worse than in using the entire training data set, is marked with a dashed line.

According to Figure 3(a), for the noisy Lorenz series, the reduction of patterns up to about 1,500 from a total of about 4,775 (approximately 30%) does not seem to affect prediction accuracy considerably. Similarly, in the Wabash series (Figure 3(b)) and Mississippi series (Figure 4(a)), too subsets of about 30–40% of the entire training data set produce equally good predictions as that using the entire training data set. This shows that the proposed clustering algorithm

Table 1 | The prediction errors on validation sets of all the data sets used (with MLPs trained with entire training data sets)

		Prediction error on validation			
Time series	**Total number of patterns (approx.)**	**NRMSE**	**MCE**	**MAE**	**RMSE**
Lorenz	4,775	0.0634	0.9415	0.6395	0.8022
Mississippi	5,480	0.0388	0.9689	207.31 $\mathrm{m^3/s}$	304.68 $\mathrm{m^3/s}$
Wabash	5,480	0.0606	0.9519	25.66 $\mathrm{m^3/s}$	46.61 $\mathrm{m^3/s}$
Friedman	5,600	0.0089	0.9920	2.5540	3.4332
Mekong	5,740	0.0662	0.9546	434.50 $\mathrm{m^3/s}$	707.90 $\mathrm{m^3/s}$

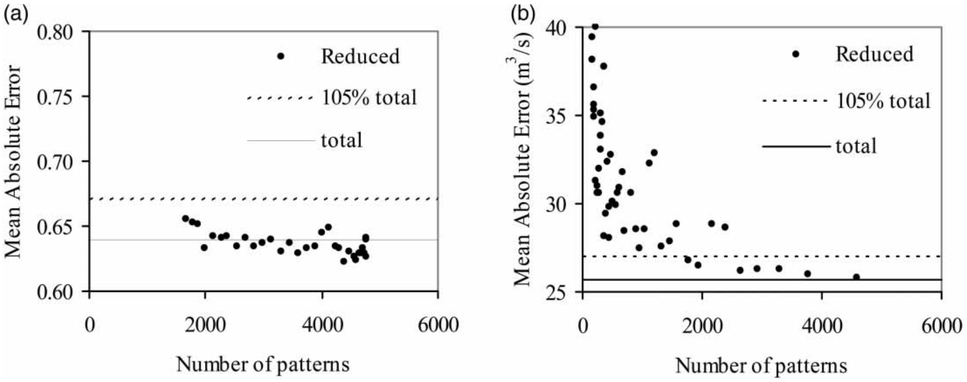

Figure 3 | Performance of the proposed clustering method on (a) Lorenz series with 5% noise level, and (b) Wabash River flow series.

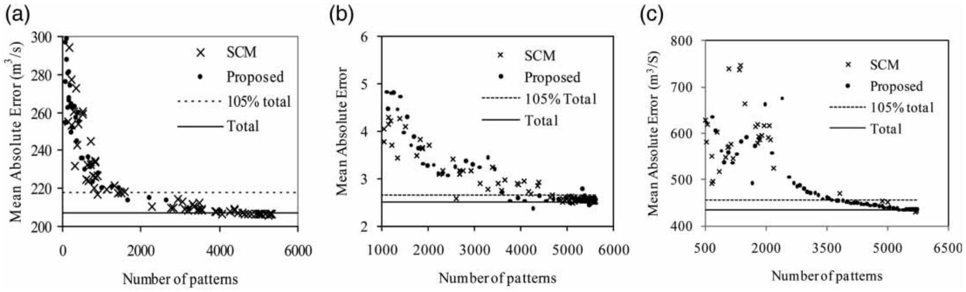

Figure 4 | Performance of the proposed clustering method and the SCM superimposed on (a) Mississippi River flow series, (b) synthetic multivariate data set, and (c) Mekong River flow series.

is capable of extracting a smaller number of effective data patterns, setting aside the redundant patterns, that give the same accuracy as using the total patterns. Comparison of the SCM and the proposed method shows that the new clustering method is equally effective as the SCM (Chiu 1994) for extracting smaller sets of system representative data. This comparison for the Mississippi data is shown in Figure 4(a). Other univariate data sets also showed similar results.

The comparisons for synthetic multivariate data set and Mekong River flow data are shown in Figure 4(b) and 4(c), respectively. They show that subsets of approximately 50–60% of the total training data set produce equally good predictions as that of using the entire training data set for the said data sets. This result shows the effectiveness of the proposed method on multivariate data sets. It is noteworthy that in the Mekong data, even with a large number of evaluations, the SCM coupled with mGA has achieved only a few solutions with a relatively large number of patterns. The amount of reduction possible in the two multivariate data sets is relatively smaller compared to the previous three data sets. Investigations showed that in Mekong and Friedman (synthetic) data sets, a relatively large number of patterns had 'distant' nearest neighbors whereas in Mississippi, Wabash, and Lorenz relatively large number of patterns had 'close' nearest neighbors. In the Mekong data this could be due to the relatively large number of variables with different values. In Friedman data which have been generated randomly using uniform distributions, the data points are, by creation, 'distantly' spaced within the domain. When points are 'closely' packed, one point can represent several such points, as explained in the section 'New Clustering Method', whereas when the points are 'distant' each of them can be carrying distinct information about the system and cannot be ignored. This can be the reason why the amount of reduction possible in Mekong and Friedman data is relatively lower than the other series considered.

Figure 5 shows the quadratic Renyi entropy for subsets of data obtained using proposed clustering method and subsets of data of the same size chosen randomly for the Mississippi River flow data. The kernel width was chosen using Equation (14). Clearly, the entropy of a clustered subset is larger than that of a randomly chosen subset of data with the same number of patterns. In order to make sure that this finding is independent of the particular value chosen for σ, H_{R2} was calculated using $0.5\sigma_{\text{opt}}$ and $1.5\sigma_{\text{opt}}$ as well. Both produced similar results as in Figure 5. Similar trends were observed for all the other data sets as well. This means that a subset obtained with the proposed clustering method is rich in information content compared to random selection of the same size.

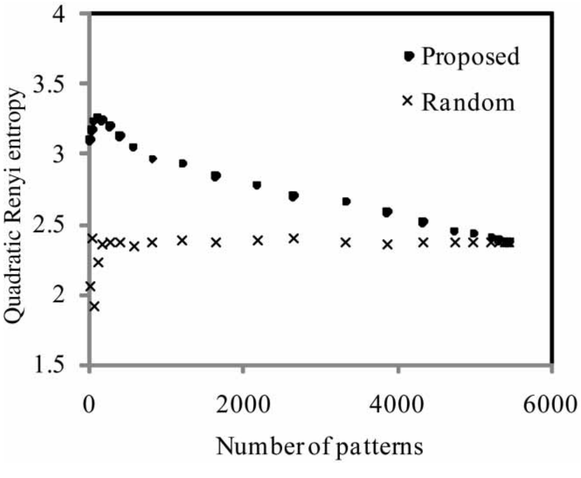

Figure 5 | Quadratic Renyi entropy of subsets of data obtained by proposed clustering method and by random selection on Mississippi River flow series.

All the tests (including the choice of d presented in the next section) were performed on a few other multivariate data sets as well. These included a Bangladesh water level prediction problem and a stream flow prediction problem with four rain gauge measurement series. They too showed similar results (not shown) to those demonstrated with the synthetic data set and the Mekong River flow data set.

DISCUSSION

Choice of *d*

Higher values of d produce a smaller number of cluster centers and vice versa. In the extreme cases, very high values will select only one cluster center and very low values will select all the points as centers. The demonstrations made so far used a trial and error approach to select suitable ranges of d to highlight the simplicity in using the proposed method. Such a trial and error approach is impractical with SCM with four parameters as will be explained in the next section. Our experience of the algorithm perhaps made it easier for us to use a trial and error approach, especially to notice that d could be as small as 0.001, in the possible range of approximately (0–1). Since suitable values for d can vary by orders of magnitudes (i.e., 10^{-1}, 10^{-3}, etc.) depending on the data sets, the following method is proposed as a foolproof way to choose subsets with the number of patterns ranging from very small numbers to close to the maximum possible, and thereby to identify suitable values for d or a range of d to explore. Here, a series of d values are chosen such that:

$$d = 0.5^{i/m}, \quad i = 1,\ 2,\ \ldots,\ 13m. \tag{15}$$

In Equation (15), m is the number of input variables which is an integer, therefore $13m$ is an integer. Figure 6(a) and 6(b) show the variation of the number of patterns with d values chosen as above for Wabash and Mekong River flow data. They show that there is a monotonic variation of the subset size with parameter d. For Wabash data the size of subsets varies from 3 to 5,397 over 65 different d values and those for Mekong data vary from 2 to 5,740 over 130 different d values. Similar trends followed for the other data sets too. Since the subset size monotonically varies with d when a desired subset size falls between the sizes obtained by the method, an interval bisection between the adjacent d values or a simple curve fitting can be used to choose a subset close to the desired size.

Problems in choosing SCM parameters

The reason for using mGA to find optimal parameter sets for SCM should better be discussed in more detail. Many studies (e.g., Lughofer 2008; Zio & Bazzo 2010; Yetilmezsoy *et al.* 2011) have used SCM with some fixed (default) values for three SCM parameters, AR (=0.5), RR (=0.15), and SF (=1.25), and varied R to obtain subsets (cluster centers) of different sizes. However, in all those applications the number of cluster centers expected had been small and these default values with varying R have been able to produce the desired results. Doan *et al.* (2005) used mGA to

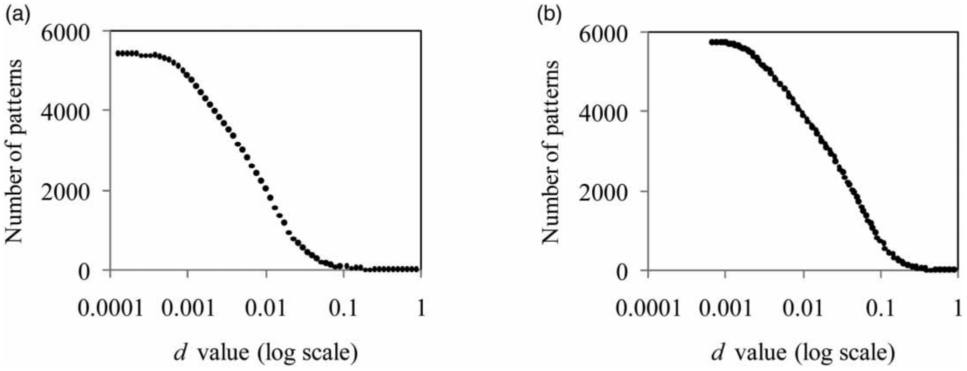

Figure 6 | Number of patterns extracted with neighborhood size, *d*, varied as proposed in Equation (15): (a) Wabash River, (b) Mekong River.

find optimal parameter sets for SCM because SCM with default parameter values did not produce optimal results for extracting effective subsets from data. This study gives the following further evidence where these default values partially/completely fail to extract system representative subsets. The variation of subset size with default values for AR, RR, and SF and with varying R is shown in Figure 7 for the Mississippi flow data. There is an abrupt change in subset size within a very small change in R when its value is around 0.0026. After increasing the resolution of R many times around this value, $R = 0.00264887$ gave a subset of size 3,122 while $R = 0.00264888$, with only a difference of 10^{-8}, produced a subset of 1,131 patterns. It was very hard to close this gap. This may be due to some nonlinear dependence and interaction between SCM parameters. In another example, in Wabash data no R value was able to produce subsets of size more than approximately 25, with the other parameters set at their defaults, thereby totally failing to extract a system representative subset. Therefore, for good subset extraction, choice of suitable values for all four SCM parameters is necessary.

On the other hand, in SCM, the variation of the size of subsets with its parameters is not monotonic. As an example, some SCM parameter sets and the sizes of the subsets they produced on Mississippi data are shown in the second and third columns of Table 2. The parameter set (3) is the mid point of the values in parameter sets (1) and (2). Ideally, we expect this set to produce a subset of which the size falls between 978 and 2,297. In this case the parameter set (3) produced a subset of size 1,027, which is between 978 and 2,297, however, this value is very much closer to 978. In a worse example, the parameter set (4) which has values between parameter set (1) and (2) produced a subset of size 442, which is outside the range of 978 to 2,297. What this means is one cannot manipulate parameters between known values to obtain a subset of desired size. A similar test for the proposed clustering method is shown in columns 4 and 5. The mid parameter value produced a subset of size 1,410, which is between 987 and 2,288, and is not as biased as in the case of SCM. Since there is a monotonic variation of subset size with the parameter d (shown in the previous section), it is easy to manipulate between $d_1 = 0.0143$ and $d_2 = 0.0075$ to obtain a subset of required size between 987 and 2,288. In this case, by a second order polynomial fit between the subset size and d, the value of d to obtain a subset of approximately 1,500 patterns was determined as 0.0103. This d value produced a subset of size 1,517, which is the desired result. Therefore, having a single parameter and its monotonic variation with the subsets obtained greatly help the user to identify subsets of required sizes and thereby acceptable prediction performance. With SCM, however, such simple manipulations are impossible and adopting a method like mGA to find optimal values for all four parameters is necessary. Nevertheless, it is noticed that in some cases (see Figure 4(c)) even with a large number of evaluations with mGAs, SCM does not produce solutions that sufficiently cover the whole range of number of patterns (i.e., from 1 to N where N is the total number of patterns).

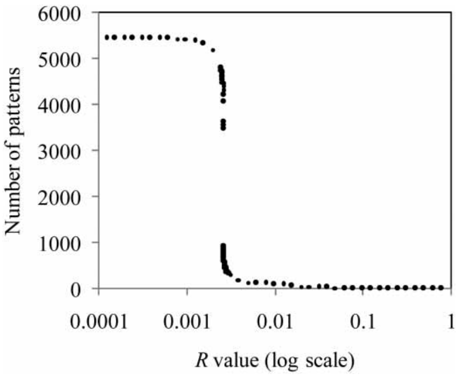

Figure 7 | Number of patterns extracted with SCM with radius of influence (R) varied with other three parameters kept at default values: Mississippi River.

Table 2 | SCM and proposed clustering parameter and the sizes of subsets obtained

SCM			**Proposed clustering method**	
Parameter set	**SCM parameters (R, AR, RR, SF)**	**Size of subset**	**Parameter *d***	**Size of subset**
(1)	(0.009, 0.80, 0.032, 1.95)	978	0.0143	987
(2)	(0.005, 0.25, 0.060, 1.80)	2,297	0.0075	2,288
(3)	(0.007, 0.525, 0.046, 1.875)	1,027	0.0109	1,410
(4)	(0.007, 0.75, 0.051, 1.85)	442	0.0103	1,517

Computational times taken by SCM and the simple clustering method

A comparison of times taken by both procedures, SCM (with mGA and NLP) and proposed clustering method with d varied, as suggested in Equation (15), to produce subsets of varying sizes, was also conducted. The times taken by the two methods for each data set are shown in Table 3. The second row shows the times taken by SCM alone in SCM-mGA-NLP coupled procedure. These times show that the use of NLP in the procedure has not affected the computational time taken by the procedure considerably. Clearly, the subsets extraction with the proposed clustering method is more efficient than using SCM. For Mekong data, the proposed method is approximately eight times more efficient than using SCM while it is as high as approximately 230 times more efficient for Friedman data. Having only a single parameter and its monotonic variation with subset size has contributed to the efficiency of the proposed method.

Remark on using subsets of time series data

It should be noted that when extracting subsets of data from time series data, the complete temporal sequence of the data set will not be preserved. In this study, the data have been extracted for the purpose of prediction model building where the modeling methods do not require the whole temporal sequence of the underlying process. For example, in building models of the form given in Equation (3), the pairs $(x_{i+T},\ \boldsymbol{X}_i)$; $i = 1,\ 2,\ ....,\ n$ are treated as independent patterns, which do not require any particular ordering. Whatever the required time, correlations are included within the pairs $(x_{i+T},\ \boldsymbol{X}_i)$ themselves, by using time delayed values (see Equations (1) and (7)).

Table 3 | Times taken (in seconds) by SCM and proposed clustering method

Method	Lorenz	Mississippi	Wabash	Friedman	Mekong
SCM with mGA and NLP	3,595	2,423	1,305	9,793	1,967
SCM alone	3,486	2,372	1,253	9,705	1,917
Proposed clustering method	143	40	105	42	258

Some similarities and differences of the proposed clustering method and SCM

The SCM and the proposed clustering method share several similarities. Both the methods select a subset of original data as cluster centers. They use a similar density measure to evaluate the potential of a data point as a cluster center. Thus, both methods give priority to points closely surrounded by other points as cluster centers. The SCM discourages closely spaced cluster centers whereas the new clustering method eliminates the selection of cluster centers which are closer than a certain distance. The SCM discourages points, lying far away from other points, being selected as cluster centers, whereas the new clustering method ensures the selection of such points. Thus the proposed clustering method tries to strike a balance between two seemingly contradicting objectives; choosing sparsely situated points while still encouraging more points to be selected from densely populated areas in the data space. Unlike almost all other clustering methods, the proposed method ensures the selection of points lying in less dense areas of data space. This is expected to be useful in function approximation problems where such points represent important infrequent events of dynamical systems. However, it can be disadvantageous in situations where such points are outlying due to noise rather than due to system dynamics. Although the proposed method was developed considering the phase space prediction of chaotic time series, it was shown to be equally effective as SCM on other multivariate data sets as well.

CONCLUSIONS

A new, simple clustering algorithm is proposed for data extraction for function approximation. The purpose of the extracted data is to serve as a representative data set of the total data set. The proposed 'simple clustering method' has only a single parameter to be specified, and suitable values for this parameter can be found more efficiently than with, for example, SCM, which has four parameters, without the necessity to adopt computationally costly methods like mGA. A foolproof method to find suitable values for this single parameter is also proposed. A comparison of times taken by SCM coupled with mGA and the proposed simple clustering method with the said method for parameter selection shows that the simple clustering method is orders of magnitude more efficient than SCM. Yet the simple clustering method is shown to be equally effective as SCM in extracting system representative subsets. Having a single parameter and its monotonic variation with the subset sizes contribute to efficient derivation of subsets of required sizes using the simple clustering method. Further, the extracted subsets are shown to contain more information content than randomly selected subsets.

The effectiveness of the subsets extracted was tested only for the prediction model formulation aspect. On the tested data sets, subsets of approximately 30–50% of the total data sets provided the same level of prediction accuracy as using the total data sets. Four different error measures supported the finding. The method was shown to be effective on phase space prediction of univariate time series data and on multivariate data sets. Demonstrations carried out on one univariate synthetic chaotic time series data set (Lorenz), one benchmark multivariate synthetic data set (Friedman), two univariate river flow series data sets, and one multivariate river flow series data set confirm the findings. Further investigation of the new technique on different multivariate data sets and on different function approximation applications is being pursued and will serve to show its robustness.

ACKNOWLEDGEMENTS

The first author wishes to thank the National University of Singapore for granting a research scholarship. US Geological Survey (USGS) data were downloaded from http://water.usgs.gov/.

REFERENCES

Abarbanel, H. D. I. 1996 *Analysis of Observed Chaotic Data*. Springer-Verlag, New York.

Brabanter, K. D., Brabanter, J. D., Suykens, J. A. K. & Moor, B. D. 2010 Optimized fixed-size kernel models for large data sets. *Computational Statistics & Data Analysis* **54** (6), 1484–1504.

Chiu, S. L. 1994 Fuzzy model identification based on cluster estimation. *Journal of Intelligent and Fuzzy Systems* **2**, 267–278.

Doan, C. D., Liong, S. Y. & Karunasinghe, D. S. K. 2005 Derivation of effective and efficient data set with subtractive clustering method and genetic algorithm. *Journal of Hydroinformatics* **7** (4), 219–233.

Friedman, J. H. 1991 Multivariate adaptive regression splines. *Annals of Statistics* **19** (1), 1–7.

Ghahramani, Z. 2004 Unsupervised learning (O. Bousquet, G. Raetsch & U. von Luxburg, eds). *Advanced Lectures on Machine Learning*. Springer, Berlin, pp. 72–112.

Girolami, M. 2002 Orthogonal series density estimation and the kernel eigenvalue problem. *Neural Computation* **14**, 669–688.

Gonzalez, J., Rojas, I., Pomares, H., Ortega, J. & Prieto, A. 2002 A new clustering technique for function approximation. *IEEE Transactions on Neural Networks* **13** (1), 132–142.

Guillen, A., Gonzalez, J., Rojas, I., Pomares, H., Herrera, L. J., Valenzuela, O. & Prieto, A. 2007 Using fuzzy logic to improve a clustering technique for function approximation. *Neurocomputing* **70**, 2853–2860.

Haykin, S. 1999 *Neural Networks: A Comprehensive Foundation*. Prentice Hall, Upper Saddle River, New Jersey.

Infrastructure Development Institute (IDI), Japan 1960–1994 The Mekong River Hydrological Database (1960–1994) in CD-ROM.

Jiang, J., Song, C., Zhao, H., Wu, C. & Liang, Y. 2008 Adaptive and iterative least squares support vector regression based on quadratic Renyi entropy. In: *Granular Computing, GrC 2008*, IEEE Press, New York, pp. 340–345.

Karunasingha, D. S. K. & Liong, S. Y. 2003 Extracting effective phase space vectors for prediction in dynamical systems approach. In: *Proceedings of the First International Conference on Hydrology and Water Resources in Asia Pacific Region, APHW*, Japan, pp. 576–581.

Karunasingha, D. S. K., Jayawardena, A. W. & Li, W. K. 2011 Evolutionary product unit based neural networks for hydrological time series analysis. *Journal of Hydroinformatics* **13** (4), 825–841.

Karunasinghe, D. S. K. & Liong, S. Y. 2006 Chaotic time series prediction with a global model: Artificial Neural Network. *Journal of Hydrology* **323**, 92–105.

Kreinovich, V. & Yam, Y. 2000 Why clustering in function approximation? Theoretical explanation. *International Journal of Intelligent Systems* **15**, 959–966.

Legates, D. R. & McCabe, Jr., G. J. 1999 Evaluating the use of 'goodness-of-fit' measures in hydrologic and hydroclimatic model validation. *Water Resources Research* **35**, 233–241.

Lin, G. F. & Chen, L. H. 2005 Time series forecasting by combining the radial basis function network and the self-organizing map. *Hydrological Processes* **19** (10), 1925–1937.

Liong, S. Y., Phoon, K. K., Pasha, M. F. K. & Doan, C. D. 2005 Efficient implementation of inverse approach for forecasting hydrological time series using micro GA. *Journal of Hydroinformatics* **7**, 151–163.

Lorenz, E. N. 1963 Deterministic non-periodic flow. *Journal of Atmospheric Science* **20**, 130–141.

Lughofer, E. 2008 Extensions of vector quantization for incremental clustering. *Pattern Recognition* **41** (3), 995–1011.

Luxburg, U., Williamson, R. C. & Guyon, I. 2012 Clustering: science or art? *JMLR: Workshop and Conference Proceedings* **27**, 65–79.

Martinez-Estudillo, A., Martinez-Estudillo, F., Hervas-Martinez, C. & Garcia-Pedrajas, N. 2006 Evolutionary product unit based neural networks for regression. *Neural Networks* **19**, 477–486.

Packard, N. H., Crutchfield, J. P., Farmer, J. D. & Shaw, R. S. 1980 Geometry from a time series. *Physical Review Letters* **45** (9), 712–716.

Pomares, H., Rojas, I., Awad, M. & Valenzuela, O. 2012 An enhanced clustering function approximation technique for a radial basis function neural network. *Mathematical and Computer Modelling* **55**, 286–302.

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

Renyi, A. 1961 On measures of entropy and information. *Proceedings of the Fourth Berkeley Symposium on Mathematics, Statistics and Probability*, University of California Press, Berkeley, CA, pp. 547–556.

Roth, V. 2004 The generalized lasso. *IEEE Transactions on Neural Networks* **15** (1), 16–28.

Sivakumar, B. 2004 Chaos theory in geophysics: Past, present and future. *Chaos, Solitons and Fractals* **19** (2), 441–462.

Takens, F. 1981 Detecting strange attractors in turbulence. In: *Dynamical Systems and Turbulence, Lecture Notes in Mathematics* (D. A. Rand & L. S. Young, eds). Springer-Verlag, Berlin, pp. 366–381.

Vapnik, V. 1999 *The Nature of Statistical Learning Theory*. Springer, Berlin.

Xu, R. & Wunsch, D. 2005 Survey of clustering algorithms. *IEEE Transactions on Neural Networks* **16** (3), 645–678.

Yager, R. R. & Filev, D. P. 1994a Approximate clustering via the mountain method. *IEEE Transactions on Systems Man and Cybernetics* **24** (8), 1279–1284.

Yager, R. R. & Filev, D. P. 1994b Generation of fuzzy rules by mountain clustering. *Journal of Intelligent Fuzzy Systems* **2**, 209–219.

Yetilmezsoy, K., Fingas, M. & Fieldhouse, B. 2011 An adaptive neuro-fuzzy approach for modeling of water-in-oil emulsion formation. *Colloids and Surfaces A: Physicochemical and Engineering Aspects* **389**, 50–62.

Yu, X. Y. & Liong, S. Y. 2007 Forecasting of hydrologic time series with ridge regression in feature space. *Journal of Hydrology* **332**, 290–302.

Yu, X. Y., Liong, S. Y. & Babovic, V. 2004 EC-SVM approach for real time hydrologic forecasting. *Journal of Hydroinformatics* **6** (3), 209–223.

Zio, E. & Bazzo, R. 2010 Optimization of the test intervals of a nuclear safety system by genetic algorithms, solution clustering and fuzzy preference assignment. *Nuclear Engineering and Technology* **42** (4), 414–425.

First received 6 June 2014; accepted in revised form 17 March 2015. Available online 28 April 2015