A modified fuzzy min–max neural network for data clustering and its application to power quality monitoring

Manjeevan Seera\textsuperscript{a,*}, Chee Peng Lim\textsuperscript{b}, Chu Kiong Loo\textsuperscript{a}, Harapajan Singh\textsuperscript{c}

\textsuperscript{a} Faculty of Computer Science and Information Technology, University of Malaya, Kuala Lumpur, Malaysia
\textsuperscript{b} Centre for Intelligent Systems Research, Deakin University, Geelong, Victoria, Australia
\textsuperscript{c} Faculty of Electrical Engineering, Universiti Teknologi MARA, Penang, Malaysia

A R T I C L E   I N F O

Article history:
Received 28 December 2013
Received in revised form 25 July 2014
Accepted 20 September 2014
Available online 8 December 2014

Keywords:
Clustering
Fuzzy min–max neural network
Benchmark study
Power quality monitoring

A B S T R A C T

When no prior knowledge is available, clustering is a useful technique for categorizing data into meaningful groups or clusters. In this paper, a modified fuzzy min–max (MFMM) clustering neural network is proposed. Its efficacy for tackling power quality monitoring tasks is demonstrated. A literature review on various clustering techniques is first presented. To evaluate the proposed MFMM model, a performance comparison study using benchmark data sets pertaining to clustering problems is conducted. The results obtained are comparable with those reported in the literature. Then, a real-world case study on power quality monitoring tasks is performed. The results are compared with those from the fuzzy c-means and k-means clustering methods. The experimental outcome positively indicates the potential of MFMM in undertaking data clustering tasks and its applicability to the power systems domain.

© 2014 Elsevier B.V. All rights reserved.

1. Introduction

Data analysis procedures can be broadly categorized as either exploratory or confirmatory, based on the models used for processing the data source [1]. Regardless of the methods used in both categories, one key component is data grouping using either goodness-of-fit to a postulated model or clustering through analysis [1]. Indeed, clustering is one of the main methods in data mining [2]. It is an unsupervised method that categorizes data into groups, such that objects in a cluster are more similar to one another as compared with those in another cluster [3]. In supervised methods, data are labeled in accordance with a specific number of target classes. In clustering methods, data samples are unlabeled, and the challenge is how to categorize them into meaningful clusters. Being a fundamental data analysis method, clustering is commonly used in many applications, which include pattern recognition, image segmentation, and function approximation [4]. Unlike standard statistical methods, many clustering methods do not depend on assumptions; therefore they are useful in situations where little or no prior knowledge is available [3].

In terms of clustering methods, they can be broadly divided into two groups: hierarchical and partitional [5]. Hierarchical clustering methods recursively locate nested clusters in either (i) an agglomerative mode, where each data sample in its own cluster is merged into the most similar pair; or (ii) in a divisive mode (also known as the top-down mode), where all data samples in a single cluster are divided into smaller clusters recursively [5]. As an example, a dendrogram represents a nested grouping of patterns, where the similarity level is produced from a hierarchical algorithm [1]. On the other hand, partitional clustering methods locate all clusters at one go, as the data partition does not impose a hierarchical structure [5]. In applications with large data sets, partitional clustering methods are advantageous as construction of a hierarchical structure (e.g. a dendrogram) can be computationally prohibitive [1].

Many clustering algorithms are available in the literature. Fundamentally, clustering is accomplished with some assumptions on a distance metric, data structure, and/or the number of clusters [6]. Among different clustering methods, k-means clustering is one of the popular algorithms [7]. The k-means clustering algorithm iteratively assigns each data sample to the closest cluster center using a distance metric. Different hybrid models involving the k-means clustering algorithm are also available, e.g. a hybrid differential evolution and one-step k-means clustering model [8]. One drawback is the estimated distance metric can be inaccurate [7]. Another popular clustering method is the fuzzy c-means algorithm [9,10]. Many of the fuzzy clustering methods can only process spatial data samples and not non-spatial ones [9]. In addition, other clustering methods are available, which include fuzzy spectral clustering algorithm [11] and subspace clustering algorithms [12].
In terms of data-based methods, incremental learning neural network models offer a number of benefits owing to their robustness in handling large scale data sets and their distributed learning capabilities [13]. Incremental learning constitutes an efficient technique in knowledge discovery, as it allows acquisition of additional knowledge/information on the fly without forgetting previously learned knowledge/information [14]. Another advantage of incremental learning is that all training data can be immediately used for learning, rather than waiting for a representative training set to be collected for learning [15]. In addition, the memory requirements tend to be smaller because a training data sample can be discarded once it has been used for learning [15]. In this domain, Simpson proposed two fuzzy min–max (FMM) networks equipped with incremental learning capabilities: one for data classification with a supervised learning model [16] and another for data clustering with an unsupervised learning model [17].

Based on both original FMM models [16,17], a number of FMM variants have been developed in the literature. In our previous work, a hybrid model consisting of supervised FMM and the classification and regression tree was proposed for fault detection and diagnosis (FDD) of induction motors [18]. The model was further enhanced with the online learning capability to tackle FDD problems [19]. A modified FMM network for tackling the phenomenon of small numbers of large hyperboxes was devised [20]. The model was then improved with the capability of rule extraction using the genetic algorithm [21]. A general FMM network with the principle of expansion and contraction combining both supervised and unsupervised learning in one model was introduced [22]. Besides that, a general reflex FMM network integrating both FMM classification and clustering algorithms, together with the concept of human reflect mechanism was proposed [23]. Based on the supervised model, the FMM network with compensatory neurons [24] that allowed online learning and, at the same time, eliminated the hyperbox contraction process was developed. In order to learn and classify data samples with multiple granularities, the granular reflex FMM network comprising hyperbox fuzzy sets to represent multi-granular data was proposed [25]. The data-core-based FMM network proposed in [26] deployed new membership functions with two types of neurons (i.e., classifying and overlapping neurons), while eliminating the contraction process. A stochastic FMM network with reinforcement learning was introduced [27]. Instead of a class label, the probability vector in a stochastic automation procedure was utilized to determine which action to take based on random selection. Motivated by the success of the aforementioned FMM-based models, we improve the clustering FMM network [17] (hereafter known as FMM) so that it is efficient for handling data clustering tasks in this study. We further demonstrate the usefulness of the modified FMM model to undertake a real-world power quality monitoring application.

In general, FMM is able to establish connection between clusters and fuzzy sets [17]. In addition, FMM possesses a number of salient features for undertaking data clustering problems, i.e., it does not require a pre-specified number of clusters and does not limit the number of clusters (i.e., it grows incrementally); it entails a simple and efficient procedure: it has only one key parameter (i.e., the hyperbox size) that needs to be fine-tuned by users [17]. Popular clustering algorithms with batch learning procedures such as fuzzy c-means and k-means clustering requires a pre-defined number of clusters to begin with, which can be a difficult task for a large data set, or when the underlying data structure keeps changing, e.g. in non-stationary environments. FMM is able to circumvent this difficulty by forming a dynamic network that is able to create the number of clusters incrementally based on the characteristics of the incoming data samples.

The main contributions of this study are two-fold: a modified FMM (MFMM) model for undertaking data clustering problems and a real-world application of MFMM to power quality monitoring task. The key innovations of this research include equipping MFMM with a centroid formation procedure in online clustering as well as allowing cluster validity analysis and performance assessment using the cophenetic correlation coefficient (CCC) [28]. The original FMM network proposed by Simpson [17] forms data clusters using a hyperbox structure. The minimum and maximum vertices of each hyperbox are encoded as the network weights. However, no centroid information with respect to the data samples clustered in each hyperbox is available. As a result, a cluster centroid estimation procedure is incorporated into original FMM in this study. The centroids are constantly monitored in every hyperbox update cycle to ensure that they remain within a hyperbox during the learning stage. It should be noted that having the cluster centroid information in MFMM is useful as this allows cluster validity analysis to be conducted. In this study, the CCC metric is adopted as a quantitative measure pertaining to the generated clusters, whereby the centroids are used for performance assessment. To demonstrate the effectiveness of MFMM, a real-world power quality monitoring problem is undertaken. Power quality is an important aspect of an electrical network, as poor power quality could lead to financial loss, especially to the industrial sector [29].

The rest of the paper is structured as follows. A literature review on various clustering methods, which include commonly used fuzzy c-means and k-means clustering algorithms, is detailed in Section 2. The overall FMM clustering neural network and its modifications are explained in Section 3. In Sections 4 and 5, a series of experiments is presented. To evaluate the usefulness of MFMM, a number of benchmark data sets are first used in Section 4, with the results compared with those reported in the literature. Then, a real-world data set from actual measurements of a power quality monitoring task is utilized in Section 5. The results from both benchmark and real-world problems are analyzed and discussed. Conclusions and suggestions for further work are presented in Section 6.

2. Literature review

A literature review on data clustering is presented in this section. The review comprises four main clustering categories, i.e., hierarchical clustering, centroid-based clustering, distribution-based clustering, and density-based clustering.

2.1. Hierarchical clustering

Hierarchical clustering (HC), also known as connectivity-based clustering, is built upon the rationale that objects that are closer together have a tighter relationship, as compared with those far away [30]. In essence, HC connects related objects based on distance to form clusters. Numerous HC applications have been reported in the literature, which include a hierarchical agglomerative clustering algorithm for planning of distributed generator units [31]. The algorithm was validated with the weighted sum method, and evaluated using two distribution systems [31]. The simulation results demonstrated the capability of the proposed HC algorithm [31]. In [32], three Trollius species of trollflowers from various regions in China were distinguished using HC analysis. The proposed method was useful for routine analysis and quality control of trollflowers [32].

A study on the asymptotic behavior of HC in conditions where both sample size and dimension grow to infinity was conducted [33]. Explicit signal versus noise boundaries among different types of clustering behaviors were derived. The analysis showed that clustering behavior inside the boundaries was similar within a wide spectrum of asymptotic settings [33]. In analyzing microarray
gene expression data, a robust complementary HC algorithm was proposed in [34]. Sequential extraction of a gene-set with proper groups of individuals was conducted. Analyses with simulated and real gene expression data sets indicated the robust properties of the HC algorithm for gene expression clustering with respect to data contamination [34].

To accurately partition a web social network, an innovative HC algorithm based on block modeling was introduced [35]. The proposed algorithm, i.e., a bottom-up agglomerative HC algorithm, used inter-connectivity and closeness of clusters to group similar data samples together in an effective manner [35]. The experimental results with real data showed that the proposed algorithm outperformed the k-means clustering algorithm [35]. An automated traffic sign detection and recognition system using HC with interest points and joint transform correlation was presented [36]. The detected image points were clustered based on a number of stable local features related to brightness and color information extracted using the Gabor filters [36]. The results indicated efficiency of the proposed method with a low false hit rate [36].

2.2. Centroid-based clustering

Clusters are represented by a centroid vector, which may not be a member of the data set in centroid-based clustering [30]. In k-means clustering, the number of clusters is fixed to a predefined setting, i.e., k. The algorithm aims to find the k cluster centroids and assign objects nearest to the cluster centroid such that the squared distances among clusters are minimized. Examples of centroid-based clustering applications include a novel binary search algorithm [2]. The proposed algorithm selected an initial set of centroids from different regions of the test data distribution, and the optimal locations were found by thoroughly exploring the regions surrounding the initial centroids [2]. The simulation results with benchmark data sets showed that the proposed algorithm was efficient for data clustering [2].

A clustering method that exploited knowledge provided by background ontologies was proposed [37]. The method was able to formulate the centroids from multivariate data sets [37]. It was evaluated using a real data set consisting of short textual answers provided by visitors to a natural park, and good results were reported [37]. In improving accuracy of cluster analysis, an algorithm that employed the James–Stein shrinkage effect in k-means clustering was studied [38]. The cluster centroids were shrunk toward the overall mean of the data samples using the James–Stein type of adjustment, and the shrinkage estimators acted as the new centroids in the next clustering iteration [38]. Using simulated and real data sets, the results obtained were better than those from k-means clustering [38].

A summation-based incremental learning algorithm for k-means clustering was introduced [39]. The algorithm avoided the zero-feature dilemma by replacing the computation of divergence between instances and centroids with just the computation of centroid entropy alone [39]. The experimental results with multiple benchmark data sets demonstrated effectiveness of the algorithm [39]. A hybrid clustering algorithm using gravitational search and k-means was described [40]. The gravitational search algorithm assisted the k-means algorithm in escaping from local optima, and increased the convergence speed of the resulting hybrid algorithm [40]. The experimental results showed better quality of the solution with increased convergence speed [40].

In [41], the prototype of a cluster was represented by a combination of the mean and fuzzy centroid information. A new measure based on co-occurrence of values to evaluate dissimilarity between data objects and cluster prototypes was employed. The performance was evaluated using four real-world data set, and improved results as compared with those from traditional clustering algorithms were reported [41]. In document retrieval indexing and gene expression analysis problems, a hybrid model known as the cooperative bisecting k-means algorithm was introduced [42]. Both k-means and bisecting k-means results were combined at each level of the binary hierarchical tree using the cooperative and merging matrices [42]. The experimental results with artificial text documents and gene expression data sets indicated better clustering quality than those from k-means alone, and with comparable computational time [42].

2.3. Distribution-based clustering

In distribution-based clustering, data clusters comprise objects that contain a similar distribution [30]. This method closely resembles the principle used to create artificial data sets, i.e., sampling random objects from a distribution [30]. In this aspect, a data elimination method based on hybridization of the k nearest neighbors and expectation maximization (EM) algorithms was proposed [43]. The k nearest neighbor algorithm was used as a preprocessor for the EM algorithm to reduce the number of training data [43]. Good performances with a number of UCI data sets were reported [43]. In [44], a total variation minimization technique was integrated into the EM algorithm for image segmentation. The EM algorithm was then used for classifying pixels into different classes based on similarity in measurements. Image segmentation was effectively produced by applying a total variation regularization technique directly to the clustering results [44].

A hybrid algorithm comprising EM and the neighborhood EM (NEM) algorithm was suggested [45]. Training was first performed using a selective hard EM method before employing NEM for fine-tuning purposes [45]. The empirical results indicated that the final clustering quality was similar to, or slightly better than, those from standard NEM [45]. An enhanced clustering model that tackled the operating cost in planning horizon was introduced [46]. Instead of using the original data samples, the algorithm created adaptive core clusters for use in the clustering process. This reduced the complexity of the original problems, and resulted in efficient solutions for tackling production inventory and distribution problems [46].

2.4. Density-based clustering

Clusters are defined as areas of higher density, as compared with the rest of the data samples, in density-based clustering (DC) [30]. Some common DC algorithms include DBSCAN (density-based spatial clustering of applications with noise) and OPTICS (ordering points to identify the clustering structure) [47]. In making DBSCAN independent of domain knowledge and satisfying the unsupervised notion of clustering, the key input parameter, i.e., ε-distance, was adaptively determined [48]. Deriving ε-distance based on data distribution of each dimension was conducted so that the algorithm could be used for subspace clustering [48]. The experimental results showed that the method was effective in finding clusters of varying sizes and shapes [48]. A novel density-based clustering algorithm with streaming data was proposed [49]. The aim was to improve both accuracy and effectiveness, and the experimental outcome demonstrated promising results with reduced computational time [49].

A new DC algorithm, DBCURE-MR, i.e., a parallelized version of DBCURE (Density-Based Clustering algorithm for large data) with MapReduce, was proposed [47]. Compared with the standard DC algorithms that finds clusters one at a time, DBCURE-MT obtains several clusters together in parallel [47]. The experimental results with various data sets indicated efficacy of DBCURE-MR in identifying cluster structures, without being too sensitive to clusters with varying densities [47]. A method to discover the community structure in a complex network using DC was proposed [50]. To
3. The fuzzy min–max clustering network

In this section, the FMM clustering network is described. The FMM learning procedure is explained in the following sections. The proposed modifications pertaining to the FMM learning procedure are also presented.

3.1. Fuzzy min–max clustering

The FMM model [17] learns by forming hyperbox fuzzy sets in its structure. The hyperbox size is controlled by a parameter known as the hyperbox size, i.e., \( \theta \in [0, 1] \). When \( \theta \) is large, the hyperbox sizes are large, resulting in a small number of hyperboxes in FMM. The opposite scenario occurs for a small setting of \( \theta \). The membership function is computed with respect to the minimum and maximum points of a hyperbox, and to the extent in which a data sample fits within the hyperbox. The \( j \)th hyperbox fuzzy set, \( B_j \), is defined by the ordered set of

\[
B_j = \{A_h, V_j, W_j, bj(A_h, V_j, W_j)\}
\]

(1)

where the number of data samples is \( h = 1, 2, \ldots, m \); \( A_h = (a_{h1}, a_{h2}, \ldots, a_{hn}) \in \mathbb{P} \) is the \( h \)th \( n \)-dimensional data sample, \( V_j = (v_{j1}, v_{j2}, \ldots, v_{jn}) \) and \( W_j = (w_{j1}, w_{j2}, \ldots, w_{jn}) \) are the minimum and maximum points of the \( j \)th hyperbox, respectively and the membership function of the \( j \)th hyperbox is \( 0 \leq bj(A_h, V_j, W_j) \leq 1 \). Fig. 1 illustrates the minimum and maximum points of a three-dimensional box.

The membership function measures the degree to which data sample \( A_h \) falls within the hyperbox formed by \( V_j \) and \( W_j \). This can be viewed as a measure of the extent to which each component is larger (or smaller) than the maximum (or minimum) point along each dimension that falls outside the min–max bound of the hyperbox. As data sample \( A_h \) approaches the hyperbox, \( bj(A_h, V_j, W_j) \) approaches 1. When the data sample is contained within the hyperbox, \( bj(A_h, V_j, W_j) = 1 \). The membership function that meets all these criteria is the sum of two complements, i.e., the average amount of the minimum point violation [17]. The resulting membership function is:

\[
bj(A_h, V_j, W_j) = \frac{1}{n} \sum_{i=1}^{n} \left[1 - f(a_{hi} - w_{ji}, c) - f(v_{ji} - a_{hi}, c)\right]
\]

(2)

where \( f() \) is a two-parameter ramp threshold function:

\[
f(x, c) = \begin{cases} 
1 & \text{if } \frac{x}{c} \geq 1 \\
\frac{x}{c} & \text{if } 0 \leq \frac{x}{c} < 1 \\
0 & \text{if } \frac{x}{c} < 0
\end{cases}
\]

(3)

The sensitivity parameter, \( c \), controls how fast the membership function decreases when a data sample is away from the hyperbox. The fuzzy set becomes crisp when \( c \) is large, and is less crisp when \( c \) is small. The membership degree, \( bj(A_h, V_j, W_j) \), is measured using (1). As explained earlier, the maximum size of a hyperbox is bounded by \( 0 < \theta < 1 \). For hyperbox \( B_j \) to expand and include \( A_h \), the expansion criterion has to satisfy the following constraint:

\[
\sum_{i=1}^{n} (\max(w_{ji}, a_{hi}) - \min(v_{ji}, a_{hi})) \leq n\theta
\]

(4)

When a hyperbox starts to expand, there is a possibility that it overlaps with other existing hyperboxes. An overlap test is introduced where a potential overlap between \( B_j \) and \( B_k \) is checked on a dimension-by-dimension basis. An overlap between \( B_j \) and \( B_k \) is indicated if one of the following four cases is satisfied for each of the \( n \) dimensions:

Case 1 : \( v_{ji} < v_{ki} < w_{ji} < w_{ki} \)

(5)

Case 2 : \( v_{ki} < v_{ji} < w_{ki} < w_{ji} \)

(6)

Case 3 : \( v_{ji} < v_{ki} \leq w_{ki} < w_{ji} \)

(7)

Case 4 : \( v_{ki} < v_{ji} \leq w_{ji} < w_{ki} \)

(8)

The overlapped region is eliminated using the hyperbox contraction process on a dimension-by-dimension basis, as follows

Case 1 : If \( v_{ji} < v_{ki} < w_{ji} < w_{ki} \),

\[
w_{ji}^\text{new} = v_{ji}^\text{new} = \frac{w_{ji}^\text{old} + v_{ki}^\text{old}}{2}
\]

(9)

Case 2 : If \( v_{ki} < v_{ji} < w_{ki} < w_{ji} \),

\[
w_{ji}^\text{new} = v_{ji}^\text{new} = \frac{w_{ji}^\text{old} + v_{ki}^\text{old}}{2}
\]

(10)

Case 3a : If \( v_{ji} < v_{ki} \leq w_{ki} < w_{ji} \),

\[
w_{ji}^\text{new} = v_{ji}^\text{new} = v_{ki}^\text{old}
\]

(11)

Case 3b : If \( v_{ji} < v_{ki} < w_{ki} < w_{ji} \),

\[
w_{ji}^\text{new} = v_{ji}^\text{new} = v_{ki}^\text{old}
\]

(12)

Case 4a : If \( v_{ki} < v_{ji} \leq w_{ji} < w_{ki} \),

\[
w_{ji}^\text{new} = v_{ji}^\text{new} = v_{ki}^\text{old}
\]

(13)

Case 4b : If \( v_{ki} < v_{ji} < w_{ji} < w_{ki} \),

\[
w_{ji}^\text{new} = v_{ji}^\text{new} = v_{ki}^\text{old}
\]

(14)

Compared with the FMM classification network [16], the overlap test is different. This is because placing decision boundaries accurately among different classes is emphasized in classification problems, while in clustering problems, the cluster boundary is adjusted in all dimensions so that the final cluster is more compact. Further details of the FMM learning procedure can be found in [17].

3.2. The proposed modifications

In this paper, a centroid for each FMM hyperbox is introduced, whereby two modifications with respect to the FMM learning algorithms are proposed. In the original FMM network, the hyperbox structure is governed by the minimum and maximum points. As such, no information pertaining to the centroid of data samples contained by each hyperbox is available. In this study, the centroid
of each hyperbox is created, which is based on the recursive average calculation, as follows

$$C_{ji}^\text{new} = \frac{N_j - 1}{N_j} (C_{ji}^\text{old}) + \frac{a_{hi}}{N_j}$$

(15)

where \(C_{ji}\) is the \(i\)th dimensional value of the centroid pertaining to the \(j\)th hyperbox, \(a_{hi}\) is the \(i\)th dimensional value of the \(h\)th data sample, and \(N_j\) is the number of data samples encoded by the \(j\)th hyperbox.

The creation of a centroid for each hyperbox (in addition to the minimum and maximum points) has a direct impact to the hyperbox formation procedure. This is because the centroid could lie outside the minimum or maximum points of a hyperbox after its contraction process. As such, another centroid monitoring rule is introduced, whereby the updated centroid is inspected to ensure that it falls within the designated hyperbox. Fig. 2 illustrates the scenario of centroid formation in FMM.

As shown in Fig. 2(a), the centroids of two hyperboxes, where the blue and green stars represent hyperboxes 1 and 2, respectively, are contained within their min–max structures. When a new data sample (yellow dot) is provided, hyperbox 2 becomes the winner, and its hyperbox structure is expanded (from the dotted boundary) to encompass the new data sample, as shown in Fig. 2(b). Since the expansion process leads to an overlapped region, hyperbox 1 is contracted, leaving its centroid (blue star) outside the hyperbox boundary, as shown in Fig. 2(c). As a result, the newly introduced centroid monitoring rule is activated to perform two actions, i.e., (i) reinstating the previous structures (minimum, maximum, and centroid) of hyperboxes 1 and 2; (ii) creating a new hyperbox by using the new data sample (red star) as its initial hyperbox. The resulting scenario is shown in Fig. 2(d). With the centroid monitoring rule, the centroid is always contained within each respective hyperbox. It should be noted that creation of a centroid for each hyperbox is important, as the centroid information is useful in evaluating the cluster validity and the overall clustering performance. A high-level pseudo-code of the MFMM network for data clustering is shown in Table 1.

### 4. Experimental studies

The effectiveness of the MFMM network is evaluated with a series of empirical studies. The first uses two benchmark data sets to illustrate the formation and evolution of the cluster structures. The second employs eight benchmark data sets to compare the clustering results of MFMM with those published in the literature. The third uses three benchmark data sets to assess the computational times and performance of MFMM, as compared with those in the literature. In addition to benchmark studies, a real-world power quality monitoring task is employed to demonstrate the applicability of the FMM clustering network to the power systems domain. In terms of performance evaluation, different indicators are available to assess cluster quality and verify whether the resulting clustering structure represents the data samples [28]. Among them, the cophenetic correlation coefficient (CCC) is one of the useful indicators that measures the distance in the proximity matrix of data samples and the distance resulted from the cluster hierarchy [28]. The definition of CCC is as follows

$$\text{CCC} = \frac{\sum_{i<j}(Y_{ij} - y)(Z_{ij} - z)}{\sqrt{\sum_{i<j}(Y_{ij} - y)^2 \sum_{i<j}(Z_{ij} - z)^2}}$$

(16)

where \(Y_{ij}\) is the distance between objects \(i\) and \(j\) in \(Y\), \(Z_{ij}\) is the cophenetic distance between objects \(i\) and \(j\), from \(Z\), and \(y\) and \(z\) are the averages of \(Y\) and \(Z\), respectively [53]. The CCC indicator can be used to compare the quality of cluster solutions from different algorithms [53]. For a high-quality solution, the magnitude of CCC should be close to 1.

In this study, the CCC score of MFMM is calculated using hyperbox centroids. For comparison purposes, the center point (average of the minimum and maximum points of each dimension) pertaining to each hyperbox is used for CCC calculation in the original FMM clustering network, since it does not have centroid information. In addition to the CCC scores, the computational time consumed by one experimental run using an Intel Core™ i5 2.50 GHz processor with 4 GB RAM on MATLAB® R2011a is recorded, which serves as an indication of the computational complexity of the MFMM network.

#### 4.1. Benchmark study 1

In the first benchmark study, two artificial data sets, i.e., Petals [54] and R15 [55] data sets, were utilized. Created in two dimensions, these data sets have been used to assess efficacy of clustering algorithms and visualize the resulting cluster structures [56]. The details of the data sets are summarized in Table 2. The Petals data set

<table>
<thead>
<tr>
<th>Table 1</th>
<th>Pseudocode for the MFMM network.</th>
</tr>
</thead>
<tbody>
<tr>
<td>for input_pattern = 1-samples</td>
<td>determine the winning hyperbox</td>
</tr>
<tr>
<td>expand the winning hyperbox, if necessary</td>
<td>perform the hyperbox overlap test</td>
</tr>
<tr>
<td>if hyperbox overlaps</td>
<td>contract the hyperbox(es)</td>
</tr>
<tr>
<td>end</td>
<td>update the winning hyperbox centroid using recursive mean estimation</td>
</tr>
<tr>
<td>check if all centroids are within their structures (min and max points)</td>
<td>reinstate all hyperbox structures to their previous states</td>
</tr>
<tr>
<td>if the centroid is not within the hyperbox structure</td>
<td>create a new hyperbox along with its centroid</td>
</tr>
<tr>
<td>end</td>
<td>end</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Table 2</th>
<th>Description and source of data sets.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data set</td>
<td>Instances</td>
</tr>
<tr>
<td>Petals</td>
<td>100</td>
</tr>
<tr>
<td>R15</td>
<td>600</td>
</tr>
</tbody>
</table>
(shaped like a flower petal) has 100 data samples from four groups. R15 is a complex data set, with 600 instances from 15 groups.

As the sequence of data samples could affect the formation of hyperboxes in FMM-based networks, a total of three runs, each with a randomized sequence of data samples, were conducted to illustrate the effects of data sequences on hyperbox structures. All data samples were normalized between 0 and 1 before presenting them to MFMM (in accordance with the requirement of FMM-based models). The hyperbox structures and the corresponding centroids at four stages of the MFMM learning cycle, i.e., 15%, 30%, 50%, and 100% of the total number of training data samples, were examined. Figs. 3 and 4 show the evolution of MFMM hyperbox structures in three separate runs with randomized sequences of data samples for the Petals and R15 problems, respectively.

As shown in Fig. 3, a total of four clusters were produced by MFMM at the end of all three runs, i.e., the same as the underlying data distributions of the Petals data set. It could be observed that the hyperbox structures differed in the early stages of learning (at 15% and 30%). However, the hyperbox structures started to indicate the underlying data distributions when the number of data samples learned by MFMM became large (at 50%). Finally, MFMM was able to represent stably the underlying data distributions with four hyperboxes (at 100%), in response to randomized sequences of data samples. A similar trend could be observed for the R15 data set, as shown in Fig. 4. Upon receiving more and more data samples, the hyperbox structures expanded incrementally to capture the underlying distributions of the incoming data samples. At the end of the learning cycle, all data samples were appropriately clustered within the hyperbox structures. The centroids formulated also showed good representations pertaining to the density of the data samples. The centroid was updated in each learning cycle so that it could represent the most populated region of the respective cluster. In short, MFMM was able to form data clusters accurately according to the underlying data distributions.

<table>
<thead>
<tr>
<th>Data set</th>
<th>Clusters</th>
<th>CCC</th>
<th>Time (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Petals</td>
<td>4</td>
<td>0.51 (0.02)</td>
<td>0.19</td>
</tr>
<tr>
<td>R15</td>
<td>15</td>
<td>0.77 (0.03)</td>
<td>0.59</td>
</tr>
</tbody>
</table>

Table 3 Results of benchmark data sets.

From the evolution trends depicted in Figs. 3 and 4, the ability of MFMM to create hyperboxes (i.e., data clusters) on the fly without requiring a pre-defined number of clusters, such as in fuzzy c-means and k-means algorithms, is clearly shown. This incremental learning capability enables MFMM to undertake complex problems where the underlying data characteristics change with time, e.g., in non-stationary environments. Even when the sequence of data samples is randomized, MFMM manages to capture the underlying data distributions, and formulates the hyperbox structures with the appropriate centroids accordingly. Based on the centroid information, the CCC scores can be calculated. Table 3 shows the average CCC scores from all three runs for both Petals and R15 data sets. As can be observed, the CCC score for the petals data set is slightly above 0.5, while that of R15 yields a higher CCC score of 0.77. The standard deviations (in parentheses) of the CCC scores are small, indicating that the performance of MFMM is stable across three runs. The computational time consumed by MFMM is short too, i.e., all below 0.6 s.

4.2. Benchmark study 2

In the second benchmark study, a total of eight data sets downloaded from [57,58] were used. The data sets contain different number of features, ranging from 2 to 60, and with up to 5300 data samples, as shown in Table 4. Three methods were used in [28], i.e., a base hierarchical clustering (Base), evidence accumulation (EA), and heuristic method, and the performances were evaluated with

![Fig. 3.](image-url) Evolution of the hyperbox structures in MFMM using the Petals data set.
experimental matrices

Comparison of CCC.

Overall, a consensus was formed with the arithmetic mean of cophenetic difference [28]. These three methods were used for comparison with MFMM. In order to have a fair comparison, the experimental procedure in [28] was followed in this study.

As shown in Table 5, the results of MFMM and FMM are compared with those from three clustering methods reported in [28]. Overall, MFMM performed better than FMM as well as the three methods in [28]. On average, the results from both Base and EA in [28] are the worst, with the lowest CCC score of 0.04 for the Banana data set. Comparatively, the Heuristic method [28] and MFMM yielded CCC scores of 0.30 and 0.83 for the Banana data set, respectively. The results of FMM were close to those from MFMM.

In general, MFMM could produce a CCC score of above 0.8 for all problems, except the Breast Cancer and Diabetes data sets.

The effects of the hyperbox size toward the MFMM results are examined too. The respective CCC scores for different hyperbox sizes, ranging from 0.1 to 0.9 with an increment of 0.2, are shown in Table 6. Notice that when the hyperbox size was large, for instance at 0.9, no CCC scores were available as only one hyperbox was produced. The CCC results varied in accordance with different hyperbox sizes as the number of clusters generated changed, and this directly affected the CCC score. As an example, the Banana data set depicted the highest CCC score at hyperbox size of 0.3 while the Wine data set at hyperbox size of 0.5.

In addition, the numbers of clusters are shown in Table 7, while the computational times are shown in Table 8. Note that when the number of clusters was two or fewer, no CCC score was produced. When the hyperbox size was small, the number of clusters produced was high. The higher the hyperbox size setting, for instance

| Table 5 | Comparison of CCC scores between FMM, MFMM, and those reported in [28]. |
|---------|---------------------|----------------|----------------|----------------|
| Banana | 0.04 | 0.04 | 0.30 | 0.79 | 0.83 |
| Breast cancer | 0.38 | 0.13 | 0.62 | 0.67 | 0.68 |
| Diabetes | 0.13 | 0.07 | 0.62 | 0.74 | 0.76 |
| Glass | 0.34 | 0.57 | 0.74 | 0.91 | 0.94 |
| Ionosphere | 0.45 | 0.10 | 0.77 | 0.92 | 0.92 |
| Liver disorders | 0.37 | 0.11 | 0.80 | 0.92 | 0.92 |
| Splice | 0.19 | 0.02 | 0.59 | 0.76 | 0.81 |
| Wine | 0.27 | 0.17 | 0.46 | 0.83 | 0.83 |

| Table 6 | CCC scores with varying hyperbox size of MFMM. |
|---------|---------------------|----------------|----------------|----------------|
| Data set | $\theta = 0.1$ | $\theta = 0.3$ | $\theta = 0.5$ | $\theta = 0.7$ | $\theta = 0.9$ |
| Banana | 0.67 | 0.83 | 0.80 | 0.69 | – |
| Breast cancer | 0.67 | 0.68 | – | – | – |
| Diabetes | 0.75 | 0.73 | 0.70 | 0.73 | 0.76 |
| Glass | 0.94 | 0.79 | – | – | – |
| Ionosphere | 0.92 | 0.87 | – | – | – |
| Liver disorders | 0.92 | 0.92 | 0.90 | – | – |
| Splice | 0.54 | 0.81 | 0.66 | 0.53 | – |
| Wine | 0.78 | 0.74 | 0.83 | – | – |
Table 7
Cluster numbers created by MFMM.

<table>
<thead>
<tr>
<th>Data set</th>
<th>(\theta = 0.1)</th>
<th>(\theta = 0.3)</th>
<th>(\theta = 0.5)</th>
<th>(\theta = 0.7)</th>
<th>(\theta = 0.9)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Banana</td>
<td>210</td>
<td>87</td>
<td>68</td>
<td>39</td>
<td>1</td>
</tr>
<tr>
<td>Breast cancer</td>
<td>106</td>
<td>86</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Diabetes</td>
<td>509</td>
<td>640</td>
<td>494</td>
<td>126</td>
<td>8</td>
</tr>
<tr>
<td>Glass</td>
<td>41</td>
<td>3</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Ionosphere</td>
<td>244</td>
<td>69</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Liver disorders</td>
<td>207</td>
<td>71</td>
<td>10</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Splice</td>
<td>2795</td>
<td>1693</td>
<td>821</td>
<td>171</td>
<td>1</td>
</tr>
<tr>
<td>Wine</td>
<td>145</td>
<td>90</td>
<td>48</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

Table 8
Computational time (in seconds) of MFMM.

<table>
<thead>
<tr>
<th>Data set</th>
<th>(\theta = 0.1)</th>
<th>(\theta = 0.3)</th>
<th>(\theta = 0.5)</th>
<th>(\theta = 0.7)</th>
<th>(\theta = 0.9)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Banana</td>
<td>20.72</td>
<td>16.46</td>
<td>14.86</td>
<td>5.16</td>
<td>4.55</td>
</tr>
<tr>
<td>Breast cancer</td>
<td>0.23</td>
<td>0.18</td>
<td>0.11</td>
<td>0.11</td>
<td>0.11</td>
</tr>
<tr>
<td>Diabetes</td>
<td>1.34</td>
<td>1.51</td>
<td>1.87</td>
<td>1.14</td>
<td>0.39</td>
</tr>
<tr>
<td>Glass</td>
<td>0.24</td>
<td>0.10</td>
<td>0.09</td>
<td>0.09</td>
<td>0.10</td>
</tr>
<tr>
<td>Ionosphere</td>
<td>0.59</td>
<td>0.14</td>
<td>0.15</td>
<td>0.14</td>
<td>0.14</td>
</tr>
<tr>
<td>Liver disorders</td>
<td>0.47</td>
<td>0.27</td>
<td>0.17</td>
<td>0.15</td>
<td>0.14</td>
</tr>
<tr>
<td>Splice</td>
<td>34.22</td>
<td>10.64</td>
<td>5.35</td>
<td>2.11</td>
<td>1.94</td>
</tr>
<tr>
<td>Wine</td>
<td>0.24</td>
<td>0.12</td>
<td>0.08</td>
<td>0.08</td>
<td>0.10</td>
</tr>
</tbody>
</table>

At 0.9, the number of hyperboxes became smaller. In some cases, only one hyperbox was produced as all data samples were able to be clustered within one hyperbox that covered almost the entire feature space. Notice that the Banana and Splice data sets consumed the longest computational time, since they contained a large number of samples. The remaining data sets showed a rapid computational time, i.e., less than half a second on average.

4.3. Benchmark study 3

In the third benchmark study, three data sets (Iris, Wine, and Yeast) downloaded from [58] were used. Table 9 shows the numbers of instances and features of each data set. As reported in [59], the number of kernels was set to 10 during the experiment using three different methods, i.e., Normalized cut (Ncut), optimized kernel k-means clustering (OKKC), affinity aggregation spectral clustering (AASC), and CKAMKC. In order to have a fair comparison, the procedure in [59] was followed, where the clustering experiments were repeated 20 times, and average of the results were taken.

Two performance indicators were used in [59], i.e., the adjusted rand index (ARI) and normalized mutual information (NMI). ARI assumes the generalized hypergeometric distribution as the model of randomness, with values between 0 and 1 [60] while NMI produces a value of 1 when two cluster structures are identical and 0 when both cluster structures are independent [61]. The mean results and standard deviation (in parentheses) of MFMM and those reported in [59] are shown in Table 10. As can be observed, MFMM yielded the best rates for both ARI and NMI in all three experiments. The most obvious increase in performance could be observed from the Wine and Yeast data sets, where results of MFMM were almost double than those reported in [59]. The standard deviations of ARI and NMI from twenty runs were between 0.015 and 0.036 as well as between 0.008 and 0.031, respectively. These small standard deviations implied that the MFMM results were stable, in response to randomized sequences of training data samples. In short, this experimental outcome indicated that the effectiveness and stability of MFMM as compared with those in [59].

To gauge the computational complexity empirically, the number of clusters of MFMM was tuned to 10 (similar to the number of kernels used in [59]), in order to compare the computational time with those in reported [59]. An Intel Core™ i5 3.40 GHz processor with 2GB RAM on MATLAB® was used in [59], while this study used an Intel Core™ i5 2.50 GHz processor with 4 GB RAM on MATLAB® R2011a. The mean computational time from 20 runs, together with the respective standard deviations (in parentheses), are shown in Table 11.

In terms of computational complexity, MFMM generally consumed more time as compared with other methods in [59]. By considering the computational time required for each iteration, MFMM was faster than OKKC, but slower than Ncut [59], AASC, and CKAMKC. As an example, in the Yeast data set, the computational time taken by AASC was 0.39 s with 3 iterations and that of CKAMKC [59] was 0.308 s with 2.1 iterations, which yielded 0.13 and 0.146 s per iteration respectively, while that of MFMM was 0.306 s per iteration.

5. Real-world case study

In this section, a real-world case study on power quality monitoring is described. Before presenting the experiment, the background of power quality monitoring is first explained. Power quality, or quality of electrical power, is a growing concern especially over the last decade as it can lead to catastrophic consequences, i.e., malfunction of devices, shortened equipment life, and long production down-times [62]. Similar to other commodities, electrical power can be categorized based on its quality [63]. In general, power quality can be evaluated from two perspectives, either supplying or consuming electricity [64]. At the generation (or supplying) end, power quality refers to the generator’s ability to generate power at 50/60 Hz with small variations. On the other hand, at the transmission and distribution (or consuming) end, it refers to voltage staying in the range of plus or minus five percent [64]. Degradation in power quality is commonly caused by disturbances such as impulses, flickers, and harmonic distortion, which lead to undesirable effects such as production downtimes in industries [62]. The widespread usage of nonlinear loads, e.g. personal computers, laser printers, variable speed drives, and many other electronic systems, creates harmonics as a major source of disturbance in commercial power distribution systems [65]. As such, it is imperative to have a reliable power quality monitoring system in order to locate the source and causes of disturbances, in order to improve the quality of electrical power [62].

A literature search reveals that using data clustering methods for power quality monitoring is relatively rare, as compared with other applications such as condition monitoring in power systems. The key reported methods include a synthetic evaluation technique for power quality based on fuzzy cluster analysis [63]. Using a dynamic clustering diagram, different clusters pertaining to power quality data were obtained, and then sorted and evaluated [63]. The analysis of fuzzy cluster analysis showed its applicability to synthetic evaluation of power quality [63]. A method to determine the optimum number of clusters with respect to a set of power quality monitoring data using a data mining algorithm based on the Minimum Message Length technique was described [66]. The proposed method used simulated power system and power quality data from an actual harmonic monitoring task in a distribution system in Australia [66]. The results indicated the effectiveness of the method in determining the optimum number of clusters [66]. For wide-area power quality monitoring, a data clustering and storing technique...
was implemented [67]. The technique was used for monitoring the quality of services in wide areas and representing the data in terms of a small set of coefficients [67]. A wavelet multi-resolution analysis was utilized for feature extraction and data clustering subject to different disturbances, and good results were reported [67]. In this study, we evaluate the effectiveness of a new data clustering method, i.e., the MFMM network, to undertake power quality monitoring tasks with real data samples.

In power quality, one of the main concerns is harmonics. Harmonics result in an increased audible noise from motors and transformers, while harmonic voltage and current cause false tripping of ground fault interrupters [68]. The total harmonic distortion (THD) is a measure of the amount of harmonic content in a waveform, [68]:

\[
\text{THD} = \sqrt{\frac{V_{\text{rms}}^2 - V_{1\text{rms}}^2}{V_{1\text{rms}}^2}}
\]

where \(V_{\text{rms}}\) is the root mean square (rms) value of the total waveform, and \(V_{1\text{rms}}\) is the rms value of the first harmonic.

In this case study, a real-world power quality monitoring study in a hospital in the state of Pahang, Malaysia, was conducted. There were several power supply disruptions due to tripping of the residual current circuit breakers at a number of distribution boards in the hospital. As such, an investigation to ascertain whether the hospital required power quality improvements in order to protect sensitive hospital equipment from harmonics and premature failures was carried out. The study was important, whereby the objectives were to identify the root cause of power supply disruptions at the distribution boards and to determine whether the supplied power to medical equipment was able to cater for a satisfactory operation. As a result, a data set pertaining to power quality monitoring was collected for a period of 11 days. The process of power quality monitoring at the main switchboard and the distribution boards was conducted in accordance with the IEE standards [69]. Reliable power meters, i.e., the 1650 Three-Phase Power Quality Recorder (as shown in Fig. 5), were used to record the voltage, currents, harmonics, and THD in 10-min intervals.

A total of 1601 samples were collected and used in this study. Twelve features comprising inputs from phases A, B, and C, of voltage harmonics, current harmonics, voltage THD, and current THD, were used for monitoring purposes. These formed the input data to the MFMM network. During the experiment, the hyperbox size was varied from 0.1 to 0.9, with an increment of 0.1. The experimental results of FMM and MFMM are shown in Table 12. It can be seen that the number of clusters reduced with the increase in the hyperbox size. As expected, the same observation occurred for the computational time. The numbers of clusters created by FMM were slightly fewer than those from MFMM, owing to the centroid formation procedure used in MFMM. Because of a more complex network structure, the computational durations of MFMM were higher than those of FMM. However, the benefit of MFMM could be observed in terms of its CCC scores, which were higher than those of FMM.

### Table 10
Performance comparison between MFMM and those reported in [59].

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Iris</td>
<td>ARI</td>
<td>NMI</td>
<td>ARI</td>
<td>NMI</td>
<td>ARI</td>
</tr>
<tr>
<td>Wine</td>
<td>(0.067)</td>
<td>(0.043)</td>
<td>(0.175)</td>
<td>(0.156)</td>
<td>(0.030)</td>
</tr>
<tr>
<td>Yeast</td>
<td>(0.011)</td>
<td>(0.005)</td>
<td>(0.076)</td>
<td>(0.064)</td>
<td>(0.005)</td>
</tr>
</tbody>
</table>

### Table 11
Comparison of computational time between MFMM and those reported in [59].

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Iris</td>
<td>Time(s)</td>
<td>#Iter</td>
<td>Time(s)</td>
<td>#Iter</td>
<td>Time(s)</td>
</tr>
<tr>
<td>Wine</td>
<td>(0.003)</td>
<td>(8.025)</td>
<td>(6.720)</td>
<td>(0.013)</td>
<td>(0.003)</td>
</tr>
<tr>
<td>Yeast</td>
<td>(0.002)</td>
<td>(32.760)</td>
<td>(3.690)</td>
<td>(0.023)</td>
<td>(0.001)</td>
</tr>
</tbody>
</table>

![Fig. 5. Power quality monitoring using three-phase power recorders.](image)
Overall, the CCC scores of FMM and MFMM ranged from 0.69 to 0.91 and 0.72 to 0.92, respectively. The best result of MFMM was produced by hyperbox size of 0.6, i.e., $\text{CCC} = 0.92$ with 21 clusters.

In addition to FMM, the fuzzy $c$-means and $k$-means clustering algorithms were implemented for comparison purposes. The overall CCC scores are shown in Table 13. The best result of fuzzy $c$-means was $\text{CCC} = 0.79$ with 33 clusters, while that of $k$-means was $\text{CCC} = 0.69$ with 163 clusters. Comparatively, MFMM (the best result of CCC $= 0.92$ with 21 clusters) outperformed fuzzy $c$-means and $k$-means clustering algorithms. In terms of computational complexity, the computational times of the $k$-means clustering algorithm were the shortest, as compared with those from fuzzy $c$-means and MFMM. In general, MFMM exhibited the advantage of better CCC scores at the expense of longer computational times.

6. Summary

In this paper, a modified FMM network for data clustering has been described. Before evaluating the usefulness of MFMM, a number of clustering methods are first reviewed. Useful modifications pertaining to FMM an efficient clustering method have been proposed. These include procedures for computing the cluster centroids. A number of benchmark data sets have been used to evaluate the evolution patterns of the cluster structures of MFMM, and to compare the MFMM performances with those reported in the literature. The CCC results of MFMM obtained from the benchmark studies are better than those from other existing clustering methods. Then, a real-world case study on power quality monitoring has been conducted. The results of MFMM are better as compared with those from the $k$-means and fuzzy $c$-means clustering algorithms. However, the structure of MFMM is more complex than that of FMM. It also requires longer computational times as compared with FMM and other methods reported in the literature. It is therefore necessary to investigate ways to improve the computational requirement of MFMM in real-world application, e.g. hardware FPGA or ASIC implementation. On the other hand, the use of transfer learning techniques with the MFMM network will be conducted, and the resulting network will be evaluated using other real-world clustering problems.

Acknowledgement

The research is supported partially by Grant RG115-12CIT from University of Malaya.

References