

A Comparison of Distance-based Semi-Supervised Fuzzy c-Means Clustering Algorithms

Daphne Teck Ching Lai and Jonathan M. Garibaldi
School of Computer Science, University of Nottingham
Email: dtl@cs.nott.ac.uk and jmg@cs.nott.ac.uk

Abstract—There are many issues to be considered in the design of distance-based fuzzy semi-supervised clustering (FSSC) algorithms. To identify these issues, we compare the performance of four such algorithms. We describe the properties of these algorithms, highlighting their key differences, and then experimentally compare their performance on common datasets. Several experimental conditions are investigated. Firstly, two forms of initialisation of the membership values of unlabelled patterns are used; $1/c$ and 0. Secondly, the algorithms are run with varying proportions of labelled patterns in the datasets, ranging from 2% to 40%. We find that no algorithm outperforms the others in all the datasets. We also observe that small modifications in similar objective functions can improve clustering, and that most of the algorithms perform slightly better with zero initialisation of unlabelled patterns. An interesting observation is that the increase in labelled patterns does not always improve clustering. From these results, we conclude that the number and scale of dimensions in the data set, initial partition matrix, distance metrics and objective functions, together, affect clustering results. In addition, we conclude that not all initially labelled patterns are good candidates for supervision.

I. INTRODUCTION

Clustering organises similar unlabelled patterns into clusters and is an unsupervised form of learning. In classification, however, labelled patterns are required to teach the algorithm the characteristics of classes, which is used to organise and tag unlabelled patterns. Thus, classification is supervised. Both techniques are widely applied in pattern recognition and data mining, often for investigation or decision-making. Good reviews of both techniques, particularly covering approaches and design issues, can be found in [1] and [2].

In fuzzy clustering [3], [4] and classification [5], membership values are assigned to patterns to indicate the degree of belongingness each pattern has to clusters or classes respectively. These give a more realistic representation of patterns, as compared to a binary-valued approach.

In reality, labelled patterns are often scarce because they are difficult or expensive to record. To increase accuracy, clustering algorithms can take advantage of available labelled patterns as examples to learn from. These algorithms that make use of unlabelled data and available labelled patterns are termed *semi-supervised clustering* algorithms. There are three main types of fuzzy semi-supervised clustering (FSSC) algorithms; constraint-based [6], distance-based [7], [8], [9] and hybrids [10]. Distance-based FSSC algorithms are found to be popular and are approached using different techniques such as RBF mapping [11], evolutionary clustering [12] and adaptive metrics [13].

Different FSSC algorithms have previously been evaluated on different datasets using various ranges of labelled patterns, which makes fair comparison difficult. This motivated our study into a comparison of four distance-based semi-supervised Fuzzy c-Means (FCM) algorithms proposed by Pedrycz and Waletzky [14], Zhang et al. [15], Li et al. [16] and Endo et al. [17]. These were chosen because we found them to be simple in operation and they have been shown to produce good results. Our objectives are to (i) compare their performances with varying quantities of labelled patterns, and (ii) explore how issues that are common to fuzzy semi-supervised clustering, such as the number of dimensions, choice of datasets, initial membership values, objective functions, distance metrics and labelled patterns affect their clustering performance. Some of these issues have already been discussed in [18] in an investigation into different distance metrics of similar objective functions. We, however, compared algorithms with different objective functions that use three different distance metrics, and algorithms with similar objective functions that employed different forms of balance between supervised and unsupervised learning. In addition, we compared experimental results from two forms of initialisation for the cluster membership of unlabelled patterns; $1/c$ and 0.

One way of comparison is by using cluster validity indices, such as those discussed in [19] and particularly in FSSC algorithms [18]. In this preliminary work, however, we use classification rate to compare clustering performance among the algorithms, as was used in [14], [15], [16], [17]. We assume that algorithms that perform well are those that achieve a high percentage of correct classifications with a low percentage of labelled patterns.

II. SEMI-SUPERVISED FCM CLUSTERING ALGORITHMS

FCM was introduced by Dunn [20] and formalised by Bezdek [4]. Gustafson and Kessel [21] introduced fuzzy covariance to generalise the distance metric, which represents patterns in a more natural manner. Early semi-supervised fuzzy clustering algorithms [22], [23], [14] adopted these techniques and produced good clustering results. Today, many FSSC algorithms [24], [13], [10], [16], [17], [15], [8] exhibit new ways to exploit FCM. Further developments of FSSC clustering are found in [25], where Pedrycz discussed the idea of fuzzy semi-supervised clustering as a knowledge-based guidance mechanism to provide additional hints about data sets. Information can be represented in different levels of

Objective Function	$J_k = \sum_{i=1}^c \sum_{k=1}^N u_{ik}^p d_{ik}^2 + \alpha \sum_{i=1}^c \sum_{k=1}^N (u_{ik} - f_{ik} b_k)^p d_{ik}^2$
Centroid	$v_i = \frac{\sum_{k=1}^N u_{ik}^2 \mathbf{x}_k}{\sum_{k=1}^N u_{ik}^2}$
Partition Matrix	$u_{ij} = \frac{1}{1+\alpha} \left\{ \frac{1+\alpha(1-b_j \sum_{l=1}^c f_{lj})}{\sum_{l=1}^c (\frac{d_{lj}}{d_{ij}})^2} + \alpha f_{ij} b_j \right\}$
Distance Metric	Mahalanobis

TABLE I
PEDRYCZ-97 ALGORITHM

Objective Function	$J_m = \sum_{i=1}^c \sum_{k=L+1}^N u_{ik}^m d_{ik}^2 + (1-a) \sum_{i=1}^c \sum_{k=1}^L u_{ik}^m d_{ik}^2 + a \sum_{i=1}^c \sum_{k=1}^L (u_{ik} - f_{ik})^m d_{ik}^2$
Centroid	$v_i^{(l)} = \frac{\sum_{k=1}^N (u_{ik}^{(l)})^2 \mathbf{x}_k}{\sum_{k=1}^N (u_{ik}^{(l)})^2}$
Partition Matrix	If unlabelled, $u_{ij} = \frac{1}{\sum_{l=1}^c (\frac{d_{lj}}{d_{ij}})^2} = u_{ij}^{FCM}$
Distance Metric	If labelled, $u_{ik} = (1-a) \left(\frac{1}{\sum_{l=1}^c (\frac{d_{lj}}{d_{ij}})^2} \right) + a f_{ik} = (1-a) u_{ik}^{FCM} + a f_{ik}$
Distance Metric	Mahalanobis

TABLE II
LI-08 ALGORITHM

Objective Function	$J_m = 2 \sum_{i=1}^c \sum_{k=1}^N u_{ik}^m (1 - K(x_k, v_i))$
Centroid	$v_i = \frac{\sum_{k=1}^{N_l} (u_{ik}^l)^m K(x_k^l, v_i) x_k^l + \sum_{k=1}^{N_u} (u_{ik}^u)^m K(x_k^u, v_i) x_k^u}{\sum_{k=1}^{N_l} (u_{ik}^l)^m K(x_k^l, v_i) + \sum_{k=1}^{N_u} (u_{ik}^u)^m K(x_k^u, v_i)}$
Partition Matrix	$u_{ik}^u = \frac{(1/(1-K(x_k^u, v_i)))^{1/(m-1)}}{\sum_{j=1}^c (1/(1-K(x_k^u, v_j)))^{1/(m-1)}}, 1 \leq i \leq c, 1 \leq k \leq N_u$
Distance Metric	Kernel-based distance

TABLE III
ZHANG-04 ALGORITHM

Objective Function	$J = \sum_{i=1}^c \sum_{k=1}^N u_{ik} \ x_k - v_i\ ^2 + \lambda^{-1} \sum_{i=1}^c \sum_{k=1}^N (u_{ik} - \bar{u}_{ik}) \log(u_{ik} - \bar{u}_{ik})$
Centroid	$v_i = \frac{\sum_{k=1}^N u_{ik} x_k}{\sum_{k=1}^N u_{ik}}$
Partition Matrix	$u_{ik} = \bar{u}_{ik} + \frac{e^{-\lambda d_{ik}}}{\sum_{j=1}^c e^{-\lambda d_{jk}}} \left(1 - \sum_{j=1}^c \bar{u}_{jk} \right)$
Distance Metric	Euclidean

TABLE IV
ENDO-09 ALGORITHM

granularity, allowing it to represent heterogeneous data in both parametric and non-parametric models.

A. The Selected Algorithms

Tables I, II, III and IV illustrate the components in the semi-supervised FCM algorithms selected for study in this paper; Pedrycz-97 [14], Li-08 [16], Zhang-04 [15] and Endo-09 [17], respectively. The notation used is as follows: N , c , U and L denote the number of patterns, the number of clusters, the partition matrix and the number of labelled patterns, respectively. v_i , u_{ik} , f_{ik} (u_{ik}^l and \bar{u}_{ik}) and d_{ik} indicate, respectively, the prototype of cluster i , the membership value for pattern k in cluster i , the membership value of labelled pattern k in cluster i , and the distance between pattern k and prototype of cluster i . The equations for calculating the centroid and partition matrix are found in these tables. They are derived using a standard optimization technique called the Lagrange multipliers, the

derivations are found in their respective papers and will not be discussed here. A semi-supervised FCM clustering algorithm iterates a series of steps to minimize the objective function until a termination criterion is satisfied. These steps are as follows:

- 1) Initialise c , U with known memberships and matrix F containing membership values of labelled patterns.
- 2) Calculate the centroids (prototypes of the clusters), v_i .
- 3) Calculate the similarities between clusters and patterns, d_{ik} using a chosen distance metric.
- 4) Update partition matrix U .
- 5) If $\|U - U'\| < \epsilon$ (ϵ being a threshold value), stop. Else go to step 2.

B. Differences

Both Pedrycz-97 and Li-08 use Mahalanobis distance to update their partition matrix. In the objective function of

Pedrycz-97, all patterns participate in unsupervised and supervised learning with parameter α to maintain a balance between the two types of learning. Li-08 is an improvement to avoid the redundant unsupervised learning of all labelled patterns in Pedrycz-97. Instead, it takes all unlabelled patterns and $1-a$ of labelled patterns to participate in unsupervised learning, and a of labelled patterns to undergo supervised learning. Though presented separately for labelled and unlabelled patterns in Li-08, the equation to update the partition matrix is similar to Pedrycz-97 except for their balance parameters, a and α , respectively. Unlabelled patterns are not used in the calculation of Li-09’s centroids, unlike the other algorithms. Like Pedrycz-97, the objective function of Endo-09 trains both labelled and unlabelled patterns in both unsupervised and supervised fashion, but using the Euclidean distance metric. The supervised training function, however, is entropy-regularised. Zhang et al. retained the objective function from the original FCM but replaced the Euclidean distance metric with a Gaussian kernel-based one. Unlike Pedrycz-97 and Li-08, in Zhang-04 only unlabelled patterns undergo supervised learning. This means that labelled patterns never gets updated or improved. As the unlabelled patterns do not contribute to the initial centroid, the labelled patterns act as seeds to form initial clusters, applied in FSSC learning [26] and non-fuzzy semi-supervised learning [27]. The trouble with using membership values of labelled patterns in both the calculation of centroids and the partition matrix, is that it makes the algorithm highly dependent on the initial membership values of labelled patterns, which can negatively impact the clustering results if they contain errors. This highlights the importance of balance parameters like α and a . In Endo-09’s partition matrix equation, the membership values of labelled patterns themselves, denoted by \bar{u}_{ik} , are both learning components and balance parameters. In doing so, the reliance on initial membership values of labelled patterns is higher than the other three algorithms.

C. Summary

Three distance metrics are used by the four algorithms; Mahalanobis distance by Pedrycz-97 and Li-08, kernel-based distance by Zhang-04 and Euclidean distance by Endo-09. Pedrycz-97, Li-08 and Endo-09 algorithms use different approaches to incorporate the supervised learning element into the objective function of the original unsupervised FCM algorithm. We can observe this element in the second component of Pedrycz-97’s and Endo-09’s objective functions, and third component in Li-08’s. The extended objective functions are then used to derive calculations of centroids and the partition matrix. Rather than using this extension in the objective function, Zhang-04 incorporated supervised learning during the calculation of the centroids. Thus, variations of both unsupervised and supervised learning components feature in these algorithms.

III. EXPERIMENTS

Each algorithm, written in R, was run on ten datasets taken from non-linearly separable datasets Iris, Wine, XOR, Wisconsin Original Breast Cancer (WOBC), Pima Indians Diabetes

dataset	N	n	c
Iris-2	150	2	3
Iris-4	150	4	3
Wine-2	178	2	3
Wine-13	178	13	3
XOR-2	200	2	4
WOBC-8	699	8	2
WOBC-2	699	2	2
PID-8	768	8	2
PID-2	768	2	2
WDBC-30	569	30	2

TABLE V
DATASETS USED IN THE EXPERIMENTS, WHERE N , n AND c ARE THE NUMBER OF PATTERNS, FEATURES AND CLUSTERS RESPECTIVELY.

(PID) and Wisconsin Diagnostic Breast Cancer (WDBC). Note that in Iris, the first class is linearly separable but the other two are not. The specifications of the datasets are shown in Table V. Apart from the XOR dataset which is manually built, all datasets are obtained from the UCI Machine Learning Repository [28]. For each dataset, experiments were repeated with different percentages of labelled patterns, 2%, 4%, 6%, 8%, 10%, 15%, 20%, 25%, 30% and 40%. Using the semi-supervised FCM algorithms, we can perform classification by having the labels replaced with numerals and the labelled patterns ordered such that the clusters and the numerically labelled classes will match. In this way, the unlabelled patterns after clustering can be matched with the numerically labelled classes from the datasets to verify correctness before counting the number of correctly matched patterns.

The following settings and modifications were made:

- Zhang et al. [15] used initial membership values of 1’s and 0’s to represent labelled and unlabelled patterns respectively. The assignment of initial membership values are not mentioned in the other algorithms. Like [18], however, we assume unlabelled patterns to hold equal initial memberships of all the clusters, having membership values of $1/c$ (where c is the number of clusters). The idea here is to improve learning using not only labelled patterns, but also unlabelled patterns.
- The membership value of each arbitrarily chosen labelled pattern belonging to a cluster is 0.9, with 0.1 membership divided among the other clusters. This is applied for the initial partition matrix of all datasets.
- From the Iris, Wine, WOBC and PID datasets, we created 2-dimensional datasets from these to observe the effects of number of attributes in datasets on clustering. We did not do this for WDBC because we felt that arbitrarily reducing a 30-dimensional image dataset to 2 dimensions was unlikely to improve clustering.
- Dimensions with missing values in the WOBC datasets were removed.
- The boolean matrix in Pedrycz-97 was removed since labelled and unlabelled patterns can be detected from the F matrix.
- We modified Endo-09 to compute prototypes from the initial partition matrix, like Pedrycz-97, rather than manually

initialising them as the original algorithm did.

- Endo-09 ran into an infinity problem with datasets Wine-13, PID-8, PID-2 and WDBC-30. This will be discussed in the next section. We modified these datasets to Wine-10, PID-6, PID-2* and WDBC-23. The modified PID-2 contains two attributes that are different from the previous version.
- The stopping criterion used in all four algorithms is taken from Pedrycz-97 [14] as follows: $\|U' - U\| = \sum_{i=1}^c \sum_{k=1}^N (u'_{ik} - u_{ik})^2$, where the threshold value is set to 0.01.
- In Pedrycz-97, α is calculated by N/L while a in Li-08 is calculated by $1 - L/N$, where L is the number of labelled patterns and N is the total number of patterns.
- Following the original algorithms, p in Pedrycz-97 and m in Zhang-04 and Li-08 are set to 2 and λ in Endo-09 is set to 1.
- At least c labelled patterns, each an example of a different cluster, is required for the clustering to run properly (although semi-supervised algorithms for incomplete labelled patterns already exist [27]).
- Separate experiments were conducted on all algorithms and datasets with zero membership values for unlabelled patterns.
- Separate experiments with manual initialisation of Endo-09 prototypes were conducted.

IV. RESULTS AND DISCUSSION

Table VI and Figure 1 show the percentage of correct classifications produced by each algorithm on different datasets. The symbol * in this table indicates that the algorithm runs into an infinity problem. Overall, Li-08 performs better than the other algorithms, achieving more than 80% correct classifications in seven out of ten datasets with 4% of labelled patterns. With 4% of labelled patterns, it produces similar results to Pedrycz-97 in the Wine dataset and Zhang-04 in the WDBC dataset but, outperformed all in the XOR dataset. With 40% labelled data, it produces nearly 100% correct classification in six out of ten datasets. Pedrycz-97 achieves more than 80% correct classification in six out of ten datasets with 6% labelled patterns, while Zhang-04 achieves this with 15% labelled patterns and Endo-09 with 30% labelled patterns. We observe that Li-08 performs slightly better than Pedrycz-97 with its modified objective function.

The results of Pedrycz-97 and Li-08 are based on a Mahalanobis distance metric, which measures similarity using the sum of squared errors and the correlation with the membership values of the patterns. The presence of the inverse covariance matrix helps to normalise dimensions of different scales, preventing dominance from dimensions with greater scales. This may give a more realistic representation of the patterns. Zhang et al. replaced the Euclidean distance metric in FCM with a Gaussian kernel-induced distance metric. The Gaussian kernel measures similarity by $K(x, y) = \exp(-\|x - y\|^2 / \sigma^2)$, with the kernel width defined by the variance σ^2 . This variance keeps the kernel value of patterns normalised. The idea behind

kernel methods are their ability to solve non-linear problems by mapping the input space into higher dimensional space (the ‘kernel trick’ [29]), which is applied to distances [30]. We see this applied in Zhang-04 with prior information of data space computed in the variance. Both Mahalanobis and kernel-based distance metrics are competitive, each doing better than the other in different datasets with 2% of labelled patterns.

The ‘curse of dimensionality’ problem is apparent in the results produced by Endo-09. The Euclidean distance metric does not reflect scale differences among dimensions in high-dimensional datasets. Dimensions with smaller scales will have less influence on the distance in the presence of dimensions with larger scales. This results in biased distance values that decrease accuracy in classification. With increasing number of dimensions, distance increases exponentially. This makes Euclidean distance tricky to use in an exponential function, derived from the Endo-09 objective function. Large distance values yield a zero value in the exponential function of Endo-09 partition matrix update equation, which in turn returns infinity. Hence, the infinity problem arises in datasets with high dimensions, large scale differences in dimensions and/or large scales in dimensions such as Wine-13, PID-8, PID-6 and WDBC-30. When we removed three dimensions (those with higher scales) from the Wine dataset, the results improved drastically. Du and Urahama tackled this problem using a parameter δ , which is computed by the inverse sum of average distances with its nearest neighbours and average distances with its furthest neighbours [11].

The algorithms do not always produce better classifications using datasets that have more dimensions. For example, as can be seen in Figure 1, Pedrycz-97, Li-08 and Zhang-04 produce better results for WDBC-2 than for WDBC-8. This is because some features have higher discrimination power compared to others. Those that have low discrimination power add more computational cost and can deteriorate clustering results [2]. Through the selection of a subset of features, those that have high discrimination power can be retained while others discarded. Kong et al. [8] incorporated feature selection with fuzzy semi-supervised learning using a feature weight variable in its objective function.

All algorithms were able to cluster most of the non-linearly separable datasets, achieving more than 80% correct classifications with at most 30% labelled patterns. Zhang-04 and Endo-09 achieve less in the XOR dataset and none of the algorithms could achieve more than 80% classification with 40% labelled patterns in the PID datasets except Pedrycz-97. According to [31], the two classes in the PID dataset are not well separated, which makes it a challenge for the algorithms.

Interestingly and unexpectedly, increases in the number of labelled patterns do not always improve classification, indicated by the troughs in the graphs of Figure 1. This observation holds for all algorithms except Pedrycz-97 in Wine-2 dataset with 8% to 10% labelled patterns. Having said this, there is still a general trend of increasing correct classifications with increase in labelled patterns. This may suggest that not all labelled patterns are good candidates to guide clustering.

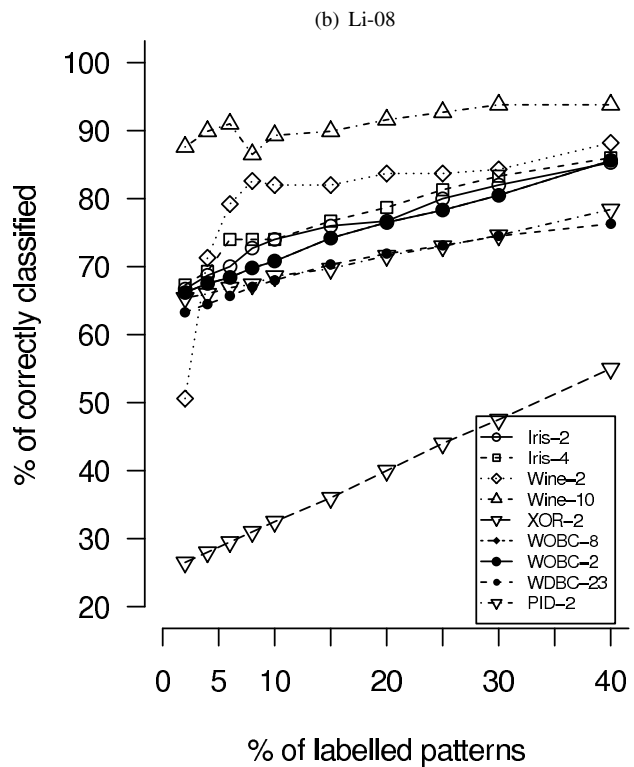
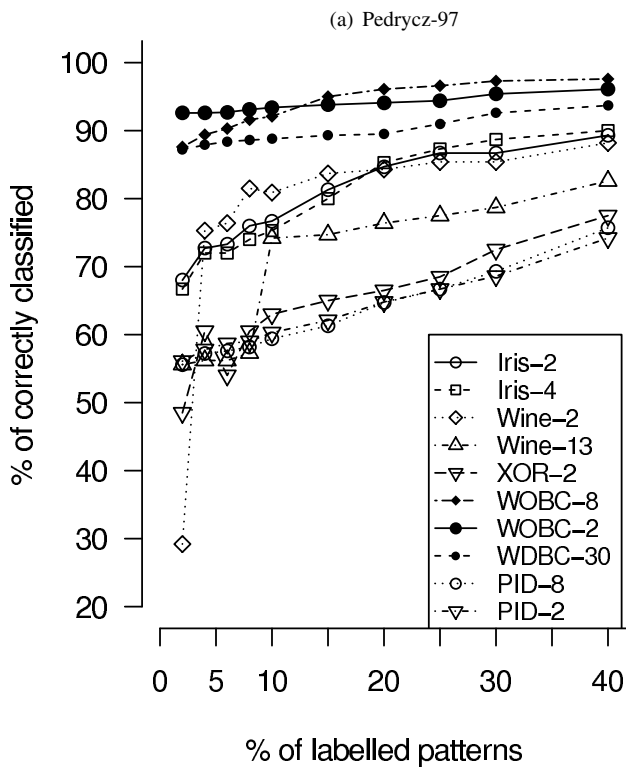
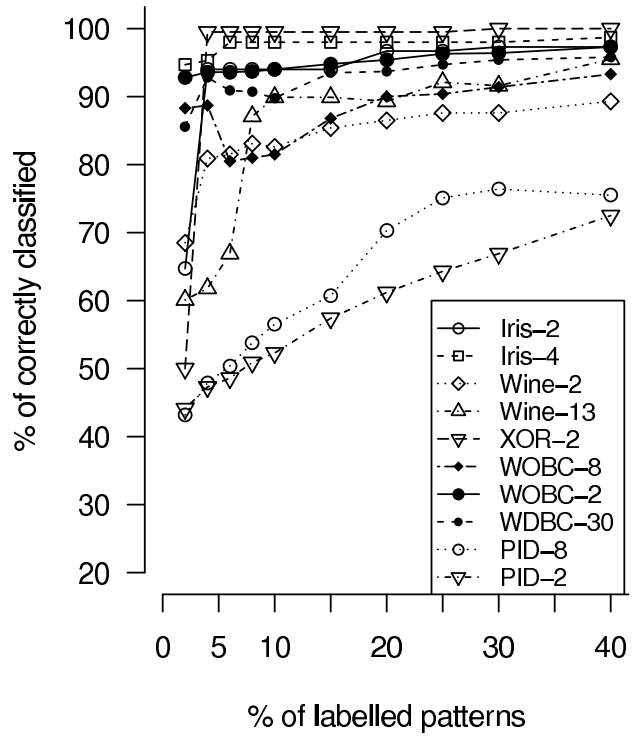
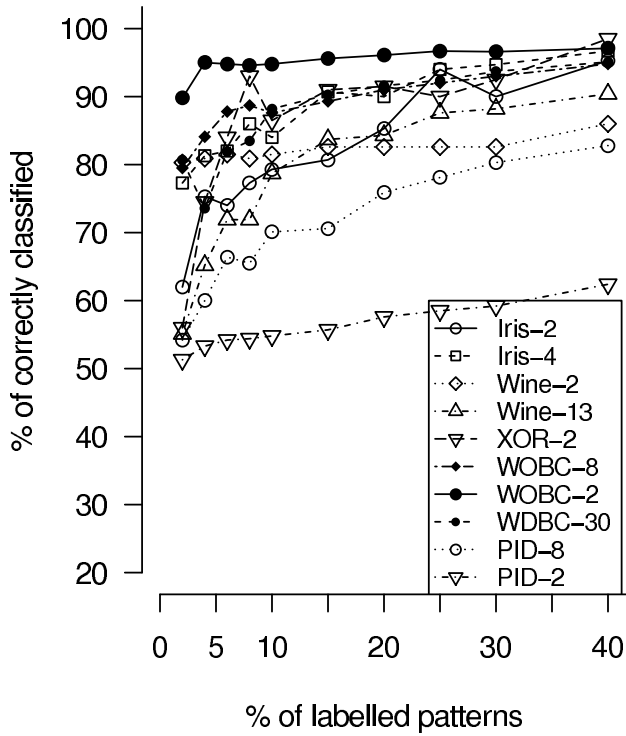


Fig. 1. Graph of percentage of correct classification against proportion of labelled patterns with $1/c$ initialisation.

dataset	With initial membership of $1/c$						With initial membership of zero					
	% labelled											
	2	4	6	8	10	20	2	4	6	8	10	20
Results produced by Pedrycz-97												
Iris-4	77.3	81.3	82.0	86.0	84.0	90.0	78.7	96.7	98.0	98.0	98.0	98.0
Iris-2	62.0	75.3	74.0	77.3	79.3	85.3	65.3	93.3	92.7	93.3	93.3	96.7
Wine-13	55.1	65.2	71.9	71.9	78.7	84.3	93.3	93.3	74.2	84.3	88.8	91.0
Wine-2	80.3	80.9	81.5	80.9	81.5	82.6	80.3	82.0	81.5	82.0	83.1	86.5
XOR-2	56.0	74.5	84.0	93.0	86.5	91.5	70.5	99.5	99.5	99.5	99.5	99.5
WOBC-8	79.5	84.1	87.8	88.7	87.6	91.0	79.3	80.3	80.5	81.1	81.7	89.7
WOBC-2	89.8	95.0	94.8	94.6	94.8	96.1	92.7	93.6	93.6	93.6	94.0	95.4
PID-8	54.2	60.0	66.4	65.5	70.1	75.9	44.3	59.1	50.8	54.4	56.6	73.7
PID-2	51.3	53.3	54.2	54.4	54.8	57.6	44.4	47.3	48.7	50.9	53.0	62.1
WDBC-30	80.7	73.6	81.9	83.5	88.2	91.6	*	85.4	88.8	87.0	89.3	90.9
Results produced by Li-08												
Iris-4	94.7	95.3	98.0	98.0	98.0	98.0	40.0	48.7	65.3	70.7	84.7	98.7
Iris-2	64.7	94.0	94.0	94.0	94.0	96.7	46.7	84.7	86.7	94.0	94.0	96.7
Wine-13	60.1	61.8	66.9	87.1	89.9	89.3	38.2	47.8	43.3	44.4	48.9	75.3
Wine-2	68.5	80.9	81.5	83.1	82.6	86.5	68.0	70.8	80.3	82.0	85.4	83.7
XOR-2	50.0	99.5	99.5	99.5	99.5	99.5	54.5	97.0	98.0	98.5	98.5	99.5
WOBC-8	88.3	88.7	80.5	81.0	81.5	90.0	67.1	87.4	92.4	90.8	91.6	92.7
WOBC-2	92.8	93.6	93.6	93.7	94.0	95.4	93.6	94.1	94.3	94.3	93.4	95.4
PID-8	43.2	47.9	50.4	53.8	56.5	70.3	46.2	62.8	66.4	70.2	73.6	75.9
PID-2	44.1	47.3	48.6	50.9	52.3	61.2	53.4	62.6	62.4	57.4	63.8	70.2
WDBC-30	85.6	92.8	90.9	90.7	89.8	93.7	*	92.8	93.1	93.3	93.1	93.7
Results produced by Zhang-04												
Iris-4	66.7	72.0	72.0	74.0	75.3	85.3	76.7	90.7	92.0	90.7	90.7	91.3
Iris-2	68.0	72.7	73.3	76.0	76.7	84.7	70.7	86.7	88.0	89.3	89.3	90.0
Wine-13	55.6	56.2	56.2	57.3	74.2	76.4	73.0	72.5	72.5	72.5	73.6	77.0
Wine-2	29.2	75.3	76.4	81.5	80.9	84.3	68.0	77.0	79.8	81.5	80.9	83.7
XOR-2	48.5	60.5	54.0	60.5	63.0	66.5	51.5	64.5	62.0	63.5	66.0	68.5
WOBC-8	87.6	89.4	90.3	91.6	92.1	96.1	96.7	63.4	96.9	95.9	96.6	97.1
WOBC-2	92.6	92.6	92.7	93.1	93.4	94.1	62.4	66.1	68.1	94.8	94.7	95.1
PID-8	55.6	57.2	57.6	58.2	59.4	64.7	54.4	52.0	56.0	56.6	57.7	62.5
PID-2	56.1	57.8	58.7	59.0	60.3	64.8	58.1	58.7	59.8	60.0	61.2	65.5
WDBC-30	87.2	87.9	88.4	88.6	88.8	89.5	81.4	83.8	91.4	91.7	91.0	91.7
Results produced by Endo-09												
Iris-4	67.3	69.3	74.0	74.0	74.0	78.7	90.0	90.7	90.7	92.7	94.0	96.7
Iris-2	66.7	68.7	70.0	72.7	74.0	76.7	88.0	88.7	88.7	88.7	89.3	92.0
Wine-10	87.6	89.9	91.0	86.5	89.3	91.6	87.1	87.1	87.1	87.6	87.1	90.4
Wine-2	50.6	71.3	79.2	82.6	82.0	83.7	40.4	80.3	80.3	80.3	80.3	83.1
XOR-2	26.5	28.0	29.5	31.0	32.5	40.0	49.5	57.5	56.5	57.5	60.5	67.5
WOBC-8	66.2	67.5	68.4	69.8	70.8	76.5	95.4	95.9	95.9	95.9	96.0	97.1
WOBC-2	66.2	67.5	68.4	69.8	70.8	76.5	93.0	93.8	94.0	94.1	94.6	96.4
PID-6	*						64.2	64.7	65.6	66.0	66.9	70.3
PID-2*	65.4	66.0	66.9	67.4	68.6	71.6	64.2	64.7	65.6	66.0	66.9	70.3
WDBC-23	63.3	64.5	65.7	67.0	68.0	71.9	51.7	52.5	53.3	54.7	55.5	60.5

TABLE VI

TABLE SHOWING RESULTS FROM RUNNING THE ALGORITHMS USING $1/c$ AND ZERO INITIAL MEMBERSHIP VALUES FOR UNLABELLED PATTERNS.

The choice of suitable labelled patterns, prior to clustering, lies in their features, their initial membership values and the objective function of the algorithm. The presence of labelled patterns that do not strongly belong to one class but have high membership values will hinder the clustering process, reducing accuracy. The objective function of the algorithm must have some corrective property to handle such patterns, such as in [24]. The authors extended the objective function to include the relationship between classes and clusters, and parameterised the confidence in the accuracy of data labels. The presence of labelled patterns with strong features of a cluster helps give a solid definition of a class. The idea of choosing unlabelled patterns to label before they are used to supervise clustering has been previously used in FSSC learning [6].

Two other separate experiments were conducted; one with

zero membership values for unlabelled patterns and another on Endo-09 with manual initialisation of prototypes as performed in the original work. The results using zero membership values for unlabelled patterns are shown in the right half of Table VI. The key observations are as follows:

- In contrast to the previous experiment, Endo-09 achieves more than 80% correct classification in six out of nine datasets with 4% of labelled patterns. We observe that this algorithm is more sensitive to initial partition matrix for supervision than the other algorithms. Thus, unlabelled patterns with $1/c$ membership values may confuse prototypes and hinder good clustering results. This causes restricted learning and is not favourable in situations when there are weak labelled patterns. Pedrycz-97, Li-08 and Zhang-04 perform slightly better with zero initial

membership values for unlabelled patterns in six, three and seven out of nine datasets, respectively.

- With manually initialised prototypes, the original Endo-09 produced final prototypes that are slightly different from those produced by the modified version, and so the results produced by these two versions are different. The dataset used in this experiment is provided in the original paper [17], which is too small to deduce which of the two produced better results. Manually initialising prototypes of datasets with large dimensions will be impractical and can make the clustering highly inaccurate.

V. CONCLUSION

As preliminary work in fuzzy semi-supervised clustering, this study is aimed at investigating some existing distance-based semi-supervised fuzzy c-means algorithms running on popular datasets. We compared the clustering results of four such algorithms, testing them on two forms of initialisation of unlabelled patterns membership values; $1/c$ and zero. In most cases, results were more favourable when using zero membership for unlabelled patterns rather than using $1/c$. We observed that issues such as scale differences of dimensions in the data set, distance metrics, objective functions and quality of labelled patterns affect clustering results. From our experimental results, we observe that Mahalanobis distance provides slightly better performance than Gaussian kernel-based distance, and that the Euclidean distance metric performs least well, on the selected data sets.

We recognise that algorithms that can achieve very high percentage of correct classifications with few labelled patterns are desirable. Also, the percentage of correct classifications should increase with increasing percentage of labelled patterns since more examples are available to guide the clustering. However, some labelled patterns with poor discriminating power, when used in some algorithms, can negatively impact on clustering results. Research into algorithms that can filter out ‘poorly’ labelled patterns or identify ‘good’ unlabelled patterns to label prior to the clustering process is of interest to us. It would also be interesting to incorporate such features into existing dimensionality reduction algorithms and run comparative performance tests.

REFERENCES

- [1] A. K. Jain, M. N. Murty, and P. J. Flynn, “Data clustering: A review,” *ACM Comput. Surv.*, vol. 31, pp. 264–323, September 1999.
- [2] A. K. Jain, R. P. W. Duin, and J. Mao, “Statistical pattern recognition: A review,” *IEEE Transaction on Pattern Analysis and Machine Intelligence*, vol. 22, no. 1, pp. 4–37, 2000.
- [3] F. Hoppner, F. Klawonn, R. Kruse, and T. Runkler, *Fuzzy Cluster Analysis*. Wiley, 1999.
- [4] J. C. Bezdek, *Pattern Recognition with Fuzzy Objective Function Algorithms*. Norwell, MA, USA: Kluwer Academic Publishers, 1981.
- [5] L. I. Kuncheva, *Fuzzy Classifier Design (Studies in Fuzziness and Soft Computing)*. Heidelberg, Germany: Physica-Verlag HD, 2000, vol. 49.
- [6] N. Grira, M. Crucianu, and N. Boujemaa, “Active semi-supervised fuzzy clustering,” *Pattern Recognition*, vol. 41, pp. 1851–1861, May 2008.
- [7] J. Gao, P. ning Tan, and H. Cheng, “Semi-supervised clustering with partial background information,” in *In Proceedings of the Sixth SIAM International Conference on Data Mining*, 2006.
- [8] Y.-q. Kong and S.-t. Wang, “Feature selection and semisupervised fuzzy clustering,” *Fuzzy Information and Engineering*, vol. 1, pp. 179–190, 2009.
- [9] S. Miyamoto, M. Yamazaki, and W. Hashimoto, “Fuzzy semi-supervised clustering with target clusters using different additional terms,” in *IEEE International Conference on Granular Computing*, 2009, pp. 444–449.
- [10] N. Wang, X. Li, and X. Luo, “Semi-supervised kernel-based fuzzy c-means with pairwise constraints,” in *IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence)*, June 2008, pp. 1098–1102.
- [11] W. Du and K. Urahama, “Semi-supervised pattern classification utilizing fuzzy clustering and nonlinear mapping of data,” *Journal of Adv Comp Intelligence and Intelligent Informatics*, vol. 11, pp. 1159–1164, 2007.
- [12] H. Liu and S.-T. Huang, “Evolutionary semi-supervised fuzzy clustering,” *Pattern Recogn. Lett.*, vol. 24, pp. 3105–3113, December 2003.
- [13] M. Ceccarelli and A. Maratea, “Semi-supervised fuzzy c-means clustering of biological data,” *Fuzzy Logic and Applications*, vol. 3849, pp. 259–266, 2006.
- [14] W. Pedrycz and J. Waletzky, “Fuzzy clustering with partial supervision,” *IEEE Transactions on Systems, Man and Cybernetics*, vol. 27, no. 5, pp. 787–795, May 1997.
- [15] D. Zhang, K. Tan, and S. Chen, “Semi-supervised kernel-based fuzzy c-means,” *Lecture Notes in Computer Science: Neural Information Processing*, vol. 3316, pp. 1229–1234, July 2004.
- [16] C. Li, L. Liu, and W. Jiang, “Objective function of semi-supervised fuzzy c-means clustering algorithm,” in *IEEE International Conference on Industrial Informatics*, July 2008, pp. 737–742.
- [17] Y. Endo, Y. Hamasuna, M. Yamashiro, and S. Miyamoto, “On semi-supervised fuzzy c-means clustering,” *IEEE International Conference on Fuzzy Systems*, 2009.
- [18] A. Bouchachia and W. Pedrycz, “Enhancement of fuzzy clustering by mechanisms of partial supervision,” *Fuzzy Sets and Systems*, vol. 157, no. 13, pp. 1733–1759, 2006.
- [19] N. Pal and J. Bezdek, “On cluster validity for the fuzzy c-means model,” *IEEE Transactions on Fuzzy Systems*, vol. 3, no. 3, pp. 370–379, Aug 1995.
- [20] J. C. Dunn, “A fuzzy relative of the isodata process and its use in detecting compact well-separated clusters,” *Journal of Cybernetics*, vol. 3, no. 3, pp. 32–57, 1973.
- [21] D. E. Gustafson and W. C. Kessel, “Fuzzy clustering with a fuzzy covariance matrix,” in *IEEE Conference on Decision and Control including the 17th Symposium on Adaptive Processes*, vol. 17, January 1978, pp. 761–766.
- [22] W. Pedrycz, “Algorithms of fuzzy clustering with partial supervision,” *Pattern Recognition Letters*, vol. 3, no. 1, pp. 13–20, 1985.
- [23] A. M. Bensaid, L. O. Hall, J. C. Bezdek, and L. P. Clarke, “Partially supervised clustering for image segmentation,” *Pattern Recognition*, vol. 29, no. 5, pp. 859–871, 1996.
- [24] A. Bouchachia and W. Pedrycz, “A semi-supervised clustering algorithm for data exploration,” *Fuzzy Sets and Systems - IFSA 2003 (International Fuzzy Systems Association World Congress Istanbul, Turkey)*, vol. 2715, pp. 107–155, 2003.
- [25] W. Pedrycz, *Knowledge-Based Clustering: From Data to Information Granules*. Wiley-Interscience, 2005.
- [26] K. Li, Z. Cao, L. Cao, and R. Zhao, “A novel semi-supervised fuzzy c-means clustering method,” in *Proceedings of Chinese Control and Decision Conference, CCDC '09.*, June 2009, pp. 3761–3765.
- [27] S. Basu, A. Banerjee, and R. J. Mooney, “Semi-supervised clustering by seeding,” in *International Conference on Machine Learning*, ser. ICML '02. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2002, pp. 27–34.
- [28] A. Asuncion and D. Newman, “UCI machine learning repository,” 2007. [Online]. Available: <http://www.ics.uci.edu/~mllearn/MLRepository.html>
- [29] A. Aizerman, E. M. Braverman, and L. I. Rozoner, “Theoretical foundations of the potential function method in pattern recognition learning,” *Automation and Remote Control*, vol. 25, pp. 821–837, 1964.
- [30] B. Scholkopf, “The kernel trick for distances,” in *Advances in Neural Information Processing Systems*, 2001, pp. 301–307.
- [31] N. Päivinen, “Clustering with a minimum spanning tree of scale-free-like structure,” *Pattern Recognition Letters*, vol. 26, pp. 921–930, May 2005.