COMPUTER METHODS AND PROGRAMS IN BIOMEDICINE XXX (2013) XXX–XXX

journal homepage: www.intl.elsevierhealth.com/journals/cmpb

Fuzzy and hard clustering analysis for thyroid disease

Ahmad Taher Azar[a,*], *Shaimaa Ahmed El-Said*[b], *Aboul Ella Hassanien*[c]

[a] *Faculty of Engineering, Misr University for Science and Technology (MUST), 6th of October City, Scientific Research Group in Egypt (SRGE), Egypt*

[b] *Electronics and Communications Department, Faculty of Engineering, Zagazig University, Zagazig, Sharkia, Egypt*

[c] *Faculty of Computers and Information, Cairo University, Scientific Research Group in Egypt (SRGE), Egypt*

ARTICLE INFO

Article history:
Received 8 September 2012
Received in revised form
21 December 2012
Accepted 6 January 2013

Keywords:
Thyroid disease
K-means clustering
K-medoids clustering
Fuzzy C-means
Gustafson–Kessel algorithm
Gath–Geva algorithm

ABSTRACT

Thyroid hormones produced by the thyroid gland help regulation of the body's metabolism. A variety of methods have been proposed in the literature for thyroid disease classification. As far as we know, clustering techniques have not been used in thyroid diseases data set so far. This paper proposes a comparison between hard and fuzzy clustering algorithms for thyroid diseases data set in order to find the optimal number of clusters. Different scalar validity measures are used in comparing the performances of the proposed clustering systems. To demonstrate the performance of each algorithm, the feature values that represent thyroid disease are used as input for the system. Several runs are carried out and recorded with a different number of clusters being specified for each run (between 2 and 11), so as to establish the optimum number of clusters. To find the optimal number of clusters, the so-called elbow criterion is applied. The experimental results revealed that for all algorithms, the elbow was located at $c = 3$. The clustering results for all algorithms are then visualized by the Sammon mapping method to find a low-dimensional (normally 2D or 3D) representation of a set of points distributed in a high dimensional pattern space. At the end of this study, some recommendations are formulated to improve determining the actual number of clusters present in the data set.

1. Introduction

A knowledge-discovery system employs a wide class of machine-learning algorithms [1] to explore the relationships among tuples, and characterize the nature of relationships that exist between them. Classification and clustering are two most commonly encountered knowledge-discovery techniques that are applied to extract knowledge [1]. Cluster analysis is an unsupervised learning method frequently used in exploratory data analysis to make a preliminary assessment of the data structure, to discover hidden structures in the data sets, and to extract (or compress) the information by drawing data prototypes. In recent years, with the development of data mining, clustering has been widely applied in various fields, such as graphic recognition, machine learning, market analysis and medical diagnosis. Clustering essentially refers to the assignment of patterns into groups (clusters) so that the objects belonging to the same group are more similar to each other than those within different groups [2,3]. Basically, Hierarchical and partitioning methods are the most popular clustering techniques [4–7]; hierarchical methods yield a complete hierarchy, i.e., a nested sequence of partitions of the input data, whereas partitioning methods seek to obtain

* *Corresponding author.*
E-mail addresses: ahmad_t_azar@yahoo.com (A.T. Azar), eng.sahmed@windowslive.com (S.A. El-Said), aboitcairo@gmail.com (A.E. Hassanien).

0169-2607/$ – see front matter © 2013 Elsevier Ireland Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.cmpb.2013.01.002

a single partition of the input data into a fixed number of clusters, usually by optimizing an objective function. The hierarchical clustering methods can be further classified into agglomerative methods and divisive methods [8]. Agglomerative methods start with each object as a cluster, recursively take two clusters with the most similarity and merge them into one cluster. Divisive methods, proceeding in the opposite way, start with all objects as one cluster, at each step select a cluster with a certain criterion [9] and bipartition the cluster with a dissimilarity measure.

Partitioning clustering can be divided into hard (or crisp) and fuzzy methods [10]. In partitioning hard clustering methods, each object of the data set must be assigned to precisely one cluster; hence, the clusters in a hard partition are disjoint. The major drawback of the hard clustering technique is that it may lose some important information which leads such a grouping becomes meaningless. *K*-means clustering [11] and partitioning around medoids [12,13] are well known techniques for performing hard partitioning algorithms. *K*-means clustering iteratively finds the k centroids and assigns every object to the nearest centroid, where the coordinate of each centroid is the mean of the coordinates of the objects in the cluster. Unfortunately, *K*-means clustering is known to be sensitive to the outliers although it is quite efficient in terms of the computational time. For this reason, *K*-medoids clustering are sometimes used, where representative objects called medoids are considered instead of centroids. Because it is based on the most centrally located object in a cluster, it is less sensitive to outliers in comparison with the *K*-means clustering. Fuzzy partitioning clustering furnishes a fuzzy partition based on the idea of the partial membership of each pattern in a given cluster. This gives the flexibility to express that data points belong to more than one cluster at the same time. Furthermore, these membership degrees offer a much finer degree of detail of the data model. Aside from assigning a data point to clusters in (equal) shares, membership degrees can also express how ambiguously or definitely a data point should belong to a cluster. Recently, De Carvalho and Tenório [14] introduced fuzzy *K*-means clustering algorithms for interval-valued data based on adaptive quadratic distances. These adaptive quadratic distances change at each algorithm iteration and can be either the same for all clusters or different from one cluster to another. Sato-Ilic [15] proposed a novel fuzzy clustering method for high dimension low sample-size interval-valued data with an adaptable variable selection. De Almeida et al. [16] proposed fuzzy Kohonen clustering networks for interval data. It is a kind of self-organizing fuzzy neural network that can show great superiority in processing the ambiguity and the uncertainty of data sets or images. Zhaoa et al. [17] presented some novel intuitionistic fuzzy clustering techniques, called intuitionistic fuzzy MST (IFMST) clustering algorithms, to deal with intuitionistic fuzzy information.

1.1. *Problem statement and research motivation*

Grekousis and Thomas [18] stated that successful clustering is achieved through the execution of clustering algorithms. The use of different algorithms leads to different results, but there is no single best approach for selecting the best algorithm, just as no algorithm offers any theoretical proof of its certainty. An intriguing problem in clustering that is yet to be solved is the relationship between the problem to be solved, that is, the data to be clustered, and the performance of various clustering algorithms. As Openshaw and Gillard [19] have shown, the choice of a suitable method is not an easy task, and there is no method that is obviously better than any other; for the results produced by each algorithm are always unreliable up to a point. Clustering performance depends strongly on the characteristics of the data to be clustered. It is well known that data quality is often a problem and that it has negative effects on the quality of the results. The robustness of the outliers and the poor-quality of the data are certainly important characteristics of any algorithm used in census data clustering. Clustering methods should be capable of providing satisfactory results and tackling the challenges presented by census data. There is no single clustering method that works best on all given problems. One method may be suitable in one area of application, but the quality of the results depends upon the algorithm used. The theoretically better choice of method does not always offer practical advantages. Furthermore, the fact that a particular method is successfully applied in one application does not necessarily mean that it can be applied with the same degree of success in another application. On the other hand, the preoccupation of researchers with a particular method or even with a particular algorithm also gives rise to difficulties. For all these reasons, the choice of the best clustering method depends on a multitude of factors, such as: a complete understanding of the aims of any clustering procedure, the examination of problems that result from the use of one algorithm, and analyses to determine whether another clustering method would consequently be preferable, and the application of criteria regarding the measurement of the performance of clustering algorithms in certain special applications [20].

The main contribution of this paper is to evaluate the performance of hard and fuzzy clustering algorithms in the clustering problem of thyroid disease. A comparison between several clustering methods is made with respect to several validity indices. Hard and fuzzy clustering algorithms are frequently used in many applications such as image retrieval, speech recognition, and pattern recognition. As far as we know, clustering techniques have not been used in thyroid disease so far.

The rest of this paper is organized as follows. Section 2 surveys related work of thyroid disease. Section 3 provides subjects and methods that are used in this paper. Section 4 presents the settings of the clustering algorithms and their implementation. Section 5 reports the results of experimental evaluations and comparisons of the proposed clustering algorithms. Section 6 concludes the study and discusses directions for future research.

2. Related work

Computer aided diagnosis (CAD) systems are getting more and more popular. Because with the help of the CAD systems, the possible errors experts made in the course of diagnosis can be avoided, and the medical data can be examined in shorter time and more detailed as well. In fact, thyroid

Please cite this article in press as: A.T. Azar, et al., Fuzzy and hard clustering analysis for thyroid disease, Comput. Methods Programs Biomed. (2013), http://dx.doi.org/10.1016/j.cmpb.2013.01.002

Table 1 – Related work for thyroid disease diagnosis studies.

Reference	Method	Accuracy (%)
Yip and Webb [24]	Function attribute finding algorithm (FAFA) + C4.5 (Pruned)	94.38 (10-FCV)
	FAFA + C4.5 (Rules)	94.38
	Einstein	91.91
	FAFA + Einstein	93.34
Serpen et al. [25]	Multi-layer perceptron (MLP)	36.74 (test data)
	Learning vector quantizer (LVQ)	81.86 (test data)
	Radial basis function (RBF)	72.09 (test data)
	Probabilistic potential function neural network (PPFNN)	78.14 (test data)
Zhang and Berardi [26]	Multi-layer feedforward neural network	98.55 (4-FCV)
Cheong and Yoon [27]	K-NN (K-nearest neighbors)	96.90 (70–30%)
	RPA (recursive partition averaging)	96.10 (70–30%)
Ozyılmaz and Yildirim [22]	Multi-layer perceptron with back-propagation (MLP with BP)	86.33 (3-FCV)
	Radial basis function (RBF)	79.08
	Adaptive conic section function neural network (CSFNN)	91.138
Pasi [28]	Linear discriminant analysis (LDA)	81.34 (test data)
	C4.5 with default learning parameters (C4.5-1)	93.26 (test data)
	C4.5 with parameter c equal to 5 (C4.5-2)	92.81 (test data)
	C4.5 with parameter c equal to 95 (C4.5-3)	92.94 (test data)
	Multi-layer perceptron (MLP)	96.24 (test data)
	Discretized interpretable multi-layer perceptron (DIMLP)	94.86 (test data)
Myles and Brown [29]	Single-model multigroup classifier (SMC) {partial least squares discriminant analysis (PLS)–quadratic discriminant analysis (QDS)}	97.20 (5-FCV)
	One-vs-all classifier (OAC) (PLS-QDA)	93.80 (5-FCV)
	Pairwise classifier (PWC) (PLS–QDA)	97.20 (5-FCV)
	Decision pathway model (DPM) (PLS-QDA)	98.20 (5-FCV)
Hassan et al. [30]	HMM (hidden Markov model)	87.91 (100–0%)
	Self-organizing map (SOM)	88.84 (100–0%)
Pechenizkiy et al. [31]	3NN-Par (parametric)	94.20 (60–40%)
	FEDIC (feature extraction for dynamic integration of classifiers) – plain	96.10 (60–40%)
	Bayesian classifier	94.80 (60–40%)
Polat et al. [23]	The artificial immune recognition system (AIRS)	81.00 (10-FCV)
	The artificial immune recognition system (AIRS) with fuzzy weighted pre-processing	85.00 (10-FCV)
Sun et al. [32]	C4.5 base	91.57 (10-FCV)
	C4.5 AdaBoost	91.12 (10-FCV)
	Base high-order pattern and weight of-evidence rule based classifier (HPWR base)	91.66 (10-FCV)
	HPWR AdaBoost	88.17 (10-FCV)
Keles and Keles [33]	Expert system for thyroid disease diagnosis with neuro fuzzy classification (ESTDD with NEFCLASS-J)	95.33 (10-FCV)
Kukkurainen and Luukka [34]	Level set classifier	96.44
Temurtas [35]	Multilayer neural networks (MLNN) with Levenberg–Marquardt (LM)	92.96 (3-FCV)
	Probabilistic neural network (PNN)	94.43 (3-FCV)
	Learning vector quantizer (LVQ)	89.79 (3-FCV)
	MLNN with LM	93.19 (10-FCV)
	PNN	94.81 (10-FCV)
	LVQ	90.05 (10-FCV)
Kodaz et al. [36]	Artificial immune recognition system (AIRS)	94.82 (10-FCV)
	Information gain based artificial immune recognition system (IG-AIRS)	95.90 (10-FCV)
Dogantekin et al. [37]	Automatic diagnosis system based on thyroid gland: ADSTG	93.77 (10-FCV)
Dogantekina et al. [38]	GDA-WSVM (generalized discriminant analysis and wavelet support vector machine)	91.86 (test data)
Chen et al. [39]	FS-PSO-SVM (feature selection-particle swarm optimization-support vector machines)	97.49 (10-FCV)
Liu et al. [40]	Fuzzy K-nearest neighbor (FKNN)	98.82 (10-FCV)
Li et al. [41]	Extreme learning machine	97.73 (10-FCV)

function diagnosis can be formulated as the classification problem, so it can be automatically performed with the aid of the CAD systems. Machine learning techniques are increasingly introduced to construct the CAD systems owing to its strong capability of extracting complex relationships in the bio-medical data. Thyroid function diagnosis is an important classification problem [21–23]. For this purpose, a variety of methods have been proposed in the literature so far. There

Please cite this article in press as: A.T. Azar, et al., Fuzzy and hard clustering analysis for thyroid disease, Comput. Methods Programs Biomed. (2013), http://dx.doi.org/10.1016/j.cmpb.2013.01.002

have been several studies reported focusing on thyroid disease diagnosis. The classification accuracies obtained by other studies for thyroid disease dataset were presented in Table 1.

3. Subjects and methods

The thyroid gland is one of the most important organs in the body as thyroid hormones are responsible for controlling the body's metabolism. As a result, thyroid function impacts on every essential organ in the body. The thyroid gland produces two active thyroid hormones, levothyroxine (abbreviated T4) and triiodothyronine (abbreviated T3). These hormones are important in production of proteins, in the regulation of body temperature, and in overall energy production and regulation. In general, thyroid disease can be divided into two broad groups of disorders: those, which primarily affect the function of the thyroid and those, which involve neoplasms, or tumors, of the thyroid. Both types of disorders are relatively common in the general population. Most thyroid problems can be treated successfully. Abnormalities of thyroid function are usually related to production of too little thyroid hormone (hypothyroidism) or production of too much thyroid hormone (hyperthyroidism) [42]. The thyroid gland is prone to several very distinct problems, some of which are extremely common. Production of too little thyroid hormone causes hypothyroidism or production of too much thyroid hormone causes hyperthyroidism. Hypothyroidism, or an under active thyroid, has many causes. Some of the causes are prior thyroid surgery, exposure to ionizing radiation, chronic inflammation of the thyroid (autoimmune thyroiditis), iodine deficiency, lack of enzymes to make thyroid hormone, and various kinds of medication. Hyperthyroidism, or an overactive thyroid, may also be caused by inflammation of the thyroid, various kinds of medications, and lack of control of thyroid hormone production. One of the most common causes is Graves' disease. Graves' disease happens when the body makes proteins that constantly tell the thyroid to make more thyroid hormone (http://www.clevelandclinic.org/health/health-info/docs/2000/2011.asp). The seriousness of thyroid disorders should not be underestimated as thyroid storm (an episode of severe hyperthyroidism) and myxedema coma (the end stage of untreated hypothyroidism) may lead to death in a significant number of cases (http://www.clevelandclinic.org/health/health-info/docs/2000/2011.asp).

In order to perform the research reported in this manuscript, the thyroid dataset taken from the UCI machine learning respiratory were used [21–23] (http://archive.ics.uci.edu/ml) (last accessed 16.11.12). The reason for using this dataset is that because it is very commonly used among the other classification systems. The dataset which consists of the thyroid disease measurements contains three classes and 215 samples from the same hospital. These individuals were divided into three groups of known classification, based on diagnosis results, healthy individual who we call as "normal" in the following for which there were 150 cases, patients suffering from hyperthyroidism (hyper) for which there were 35 cases, from hypothyroidism (hypo) for which there were 30 cases. Table 2 describes the class distribution. The thyroid data are measurement of the thyroid gland. Each individual was characterized by the result of five laboratory tests and all the features are continuous as shown in Table 3.

4. Clustering algorithms

Clustering [43,44,2] is an unsupervised classification method. It is used when the only data available are unlabelled, and no structural information about it is available. In clustering (also known as exploratory data analysis), a set of patterns, usually vectors in a multi-dimensional space, are organized into coherent and contrasted groups or clusters, such that all data in the same group are similar to each other, while data from different clusters are dissimilar [43]. The purpose of any clustering technique is to evolve a partition matrix $U(X)$ of a given data set X (consisting of, say, n patterns, $X = \{x_1, x_2, \ldots, x_n\}$) so as to find a number, say K, of clusters $(C_1, C_2, \ldots, C_K)$. The partition matrix $U(X)$ of size $K \times n$ may be represented as $U = [u_{kj}]$, $k = 1, \ldots, K$ and $j = 1, \ldots, n$, where u_{kj} is the membership of pattern x_j to clusters C_k. The following subsections present a brief description of various proposed clustering techniques.

4.1. *Hard clustering*

Hard clustering methods are based on classical set theory, and require that an object either does or does not belong to a cluster. Hard clustering in a data set X means partitioning the data into a specified number of mutually exclusive subsets of X. The number of subsets (clusters) is denoted by c. The discrete nature of hard partitioning also causes analytical and algorithmic intractability of algorithms based on analytic functionals, since these functionals are not differentiable. The objective of hard clustering is to partition a given data set X into c clusters. The number of clusters should be known beforehand either through prior knowledge of the data or choosing a trial value. Using classical sets, a hard partition can be defined as a family of subsets $\{Ai|1 \le i \le c \subset P(X)\}$. The subsets Ai contain all the data in X, the subsets must be disjoint and none of them is empty nor contains all the data in X. Hard partitioning results in either a 0 or 1 membership function. Here μ_i is the characteristic function of the subset A_i and its value can be zero or one.

Definition. Hard partitioning space Let $X = [x_1, x_2, \ldots, x_N]$ be a finite set and let $2 \le c < N$ be an integer. The hard partitioning space for X is the set:

$$M_{hc} = \left\{ U \in R^{N \times c} |_{\mu_{ik}} \in 0, 1, \quad \forall_i, k; \qquad \sum_{k=1}^{c} \mu_{ik} = 1, \quad \forall_i; \right.$$
$$\left. 0 < \sum_{i=1}^{N} \mu_{ik} < N, \quad \forall_k \right\} \tag{1}$$

The objective of clustering is to partition the data set X into c clusters. The number of clusters should be known beforehand either through priori knowledge of the data or choosing a trial value.

Please cite this article in press as: A.T. Azar, et al., Fuzzy and hard clustering analysis for thyroid disease, Comput. Methods Programs Biomed. (2013), http://dx.doi.org/10.1016/j.cmpb.2013.01.002

Table 2 – Class distribution of the thyroid dataset.

Index	Class name	Class size	Class distribution (%)
C1	Normal	150	69.77
C2	Hyper	35	16.28
C3	Hypo	30	13.95

4.1.1. *K-means*

K-means [11] is one of the simplest unsupervised learning algorithms that solve the well-known clustering problem. Let $X = \{x_i\}$, $i = 1, \ldots, n$ be the set of n d-dimensional points to be clustered into a set of K clusters, $C = \{c_k, k = 1, \ldots, K\}$. *K*-means algorithm finds a partition such that the squared error between the empirical mean of a cluster and the points in the cluster is minimized. Let μ_k be the mean of cluster c_k. The squared error between μ_k and the points in cluster c_k is defined in [3]. The goal of *K*-means is to minimize the sum of the squared error over all K clusters [3]. Minimizing this objective function is known to be an NP-(non-deterministic polynomial-time) hard problem (even for $K = 2$) [46]. Thus *K*-means, which is a greedy algorithm, can only converge to a local minimum, even though recent study has shown with a large probability *K*-means could converge to the global optimum when clusters are well separated. *K*-means starts with an initial partition with K clusters and assign patterns to clusters so as to reduce the squared error. Since the squared error always decreases with an increase in the number of clusters K (with $J(C) = 0$ when $K = n$), it can be minimized only for a fixed number of clusters. The *K*-means algorithm requires three user-specified parameters: number of clusters K, cluster initialization, and distance metric. The most critical choice is K. While no perfect mathematical criterion exists, a number of heuristics are available for choosing K [3]. The main steps of *K*-means algorithm are as follows [2]:

1. Initializing the cluster centers to randomly selected points.
2. Assigning each point to the cluster which has the closest cluster center.
3. Updating the cluster centers.
4. Going back to stage 2.

The *K*-means algorithm is stopped if a specified error criterion is met, or after a specified number of iterations, or if fewer than a specified number of objects change clusters. Although it is simple and relatively fast, the *K*-means algorithm is a gradient descent method and thus may be trapped in local minima. So it may be necessary to run the algorithm multiple times with a different set of initial cluster centers to find the optimal solution. *K*-means is typically used with the Euclidean metric for computing the distance between points and cluster centers. As a result, *K*-means finds spherical or ball-shaped clusters in data. *K*-means with Mahalanobis distance metric has been used to detect hyperellipsoidal clusters [47], but this comes at the expense of higher computational cost [3].

4.1.2. *K-medoids*

K-medoids [12] is a clustering algorithm related to the *K*-means algorithm and the medoid-shift algorithm. Both the *K*-means and *K*-medoids algorithms are partitional (breaking the dataset up into groups) and both attempt to minimize squared error, the distance between points labeled to be in a cluster and a point designated as the center of that cluster. In contrast to the *K*-means algorithm, *K*-medoids chooses data points as centers (medoids or exemplars). *K*-medoids is also a partitioning technique of clustering that clusters the data set of n objects into k clusters with k known a priori. A useful tool for determining k is the silhouette. It could be more robust to noise and outliers as compared to *K*-means because it minimizes a sum of general pairwise dissimilarities instead of a sum of squared Euclidean distances. Given similarity values between every pair of data objects, an iterative learning process can group data objects into clusters to maximize the sum of similarity values for each cluster. That is, when the sum of similarity values is maximized, *K*-medoid algorithm groups data objects of equal to or lesser than similarity value in one cluster [48]. The decision boundaries that are generated by a given *K*-medoid model are the perpendicular bisector hyperplane of the line segment from the medoid of one cluster to another. That is, the variance of data distribution in each cluster is assumed to be uniform. However, the variances of different orientations of the data distribution in a cluster may be different [48]. The most common realization of *K*-medoid clustering is the partitioning around medoids (PAM) algorithm [12] and is as follows:

1. Initialize: randomly select k of the n data points as the medoids.
2. Assignment step: associate each data point to the closest medoid.
3. Update step: for each medoid m and each data point o associated to m swap m and o and compute the total cost of the configuration (that is, the average dissimilarity of o to all the data points associated to m). Select the medoid o with the lowest cost of the configuration.
4. Repeat alternating steps 2 and 3 until there is no change in the assignments.

Table 3 – Description of thyroid dataset attributes.

Attributes	Description	Mean	Standard deviation
F1	T3-resin uptake test (RT3U)	109.595	13.145
F2	Total serum thyroxin as measured by the isotopic displacement method (T4)	9.805	4.697
F3	Total serum triiodothyronine as measured by radioimmunoassay (T3 or T3RIA)	2.050	1.420
F4	Basal thyroid-stimulating hormone (TSH) as measured by radioimmunoassay	2.880	6.118
F5	Increase TSH after injection of TSH-releasing hormone (ΔTSH)	4.199	8.071

Please cite this article in press as: A.T. Azar, et al., Fuzzy and hard clustering analysis for thyroid disease, Comput. Methods Programs Biomed. (2013), http://dx.doi.org/10.1016/j.cmpb.2013.01.002

Since the solution of medoids are located in the sample space, which is discrete, the finding of medoids typically needs to be accomplished with heuristic search algorithms. This is different from the case of *K*-means, in which the centroid of each cluster can be updated iteratively because the centroid is a variable in continues space. Due to the fact that exhaustive search of medoids is an NP hard problem, Kaufman and Rousseeuw [13] proposed one approximate search algorithm called PAM. The main task of PAM is to find *k* objects as medoids. Clusters are then formed by assigning each of other objects to the nearest medoid. Different scalable extensions of PAM, such as CLARA [13] and CLARANS [49] are proposed to deal with large data sets.

4.2. *Fuzzy clustering*

Fuzzy cluster analysis therefore allows gradual memberships of data points to clusters in [0, 1]. This gives the flexibility to express that data points belong to more than one cluster at the same time. Furthermore, these membership degrees offer a much finer degree of detail of the data model. Aside from assigning a data point to clusters in (equal) shares, membership degrees can also express how ambiguously or definitely a data point should belong to a cluster. The concept of these membership degrees is substantiated by the definition and interpretation of fuzzy sets.

4.2.1. *Fuzzy C-means*

The fuzzy *C*-means algorithm was presented in its initial form by Dunn [50] and completed by Bezdek [51] as an alternative to earlier *K*-means clustering. FCM partitions a collection of n vector x_i, $i=1,\ldots,n$ into C fuzzy groups, and finds a cluster center in each group such that an objective function of a dissimilarity measure is minimized. The major difference between FCM and *K*-means is that FCM employs fuzzy partitioning such that a given data point can belong to several groups with the degree of belongingness specified by membership grades between 0 and 1. In FCM, the membership matrix $U=[\mu_{ij}]$ is allowed to have not only 0 and 1 but also the elements with any values between 0 and 1.

The objective function of FCM can be formulated as follows:

$$J_m = \sum_{i=1}^{c}\sum_{j=1}^{n}\mu_{ij}^{m}||x_i - c_i||^2 \tag{2}$$

where c is the number of clusters; c_i is the cluster center of fuzzy group i and the parameter m is a weighting exponent on each fuzzy membership with $m\in(1,\infty)$. It controls the weights used in the distance function and defines how fuzzy the results are, that is, the participation percentages for every point in every cluster. No theoretical or mathematical proof distinguishes the best c, but it is generally appropriate to achieve values between 1 and 3. When $m=1$, there is no fuzziness. When m tends to infinity, there is complete fuzziness, and all points display cluster membership to a considerable degree. However, experience is the best rule of thumb for selecting m. For most classifications, $1.5<m<3.0$ offers good results. If a cluster number $c\geq 2$ and a fuzzification parameter m greater than 1 are considered, then the algorithm selects memberships μ_k in every cluster for every point, so that the total sum of the memberships is one. Fuzzy partitioning is carried out through an iterative optimization of the objective function shown above (Eq. (2)), updating of membership μ_{ij} by:

$$\mu_{ij} = \frac{1}{\sum_{k=1}^{C}(||x_i - c_i||/x_j - c_k)^{2/(m-1)}} \tag{3}$$

The cluster center c_i can be obtained from:

$$c_i = \frac{\sum_{j=1}^{N}\mu_{ij}^{m}x_j}{\sum_{j=1}^{N}\mu_{ij}^{m}} \tag{4}$$

Table 4 – Fuzzy C-means algorithm.

- Select the fuzzifier exponent m ($m>1$) and initialize the fuzzy partition matrix $U=(\mu_{ij})$ randomly
- **while** termination conditions not met **do**
1. Compute the cluster centers c_i
2. Update the fuzzy partition matrix $U=(\mu_{ij})$

end while

In Table 4 a brief description of the fuzzy *C*-means algorithm is given. Algorithm can be terminated when the relative change in the cluster centers gets very small or the objective function J cannot be minimized anymore.

4.2.2. *Gustafson–Kessel algorithm*

The algorithm of Gustafson–Kessel fuzzy clustering (GK-clust) is firstly proposed in Gustafson and Kessel [52]. The numerically robust Gustafson–Kessel algorithm was described in Babuska et al. [53]. Gustafson and Kessel [52] extended the standard fuzzy *C*-means algorithm by employing an adaptive distance norm, in order to detect clusters of different geometrical shapes in one data set. The Gustafson–Kessel algorithm is based on iterative optimization of an objective functional of the *C*-means type [52].

$$J(Z;U,V,\{A_i\}) = \sum_{i=1}^{K}\sum_{k=1}^{N}(\mu_{ik})^{m}D_{ikA_i}^{2} \tag{5}$$

Here, $U=[\mu_{ik}]\in[0,\ 1]^{K\times N}$ is a fuzzy portioning matrix of the data $Z\in\mathbb{R}^n$, $V=[v_1,v_2,\ldots,v_K]$, $v_i\in\mathbb{R}^n$ is a K-tuple of cluster prototypes and m is a scalar parameter which determines the fuzziness of the resulting clusters. The distance norm D_{ikA} can account for clusters of different geometrical shapes in one data set:

$$D_{ikA_i}^{2} = (z_k - v_i)^{T}A_i(z_k - v_i) \tag{6}$$

The metric of each cluster is defined by a local norm-inducing matrix A_i, which is used as one of the optimization variables. This allows the distance norm to adapt to the local topological structure of the data. The minimization of the GK objective functional is achieved by using the alternating optimization method according to the well-known algorithm shown in Table 5 [52].

Table 5 – Gustafson–Kessel algorithm.

Given the data set Z, choose the number of clusters $1<K<N$, the weighting exponent $m>1$ (usually 2), the termination tolerance $\varepsilon>0$ (usually 10^{-3}) and the cluster volumes ρ_i (usually 1). Initialize the partition matrix randomly, such that $U^{(0)} \in M_{fK}$ (i.e., belongs to the fuzzy partitioning space)
Step 1: Compute cluster prototypes (means)
Step 2: Compute the cluster covariance matrices
Step 3: Compute the distances
Step 4: Update the partition matrix until $\|U^{(l)} - U^{(l-1)}\| < \varepsilon$

The above-mentioned numerical problems occur in step 3 of the algorithm where the cluster covariance matrix F_i is inverted. If the number of data samples is small or when the data within a cluster are linearly correlated, the covariance matrix may become (nearly) singular [53].

4.2.3. *Gath–Geva algorithm*

Gath and Geva [54] generalize the maximum likelihood estimation for the fuzzy clustering. The fuzzy maximum likelihood estimates (FMLE) clustering algorithm makes use of the FMLE distance norm [45]. This norm distance includes an exponential term, which means that the norm distance decreases faster than the inner product norm found in the *Gustafson-Kessel* algorithm. Given the data set X, choose the number of clusters by cluster validation as $1<c<N$ and the termination tolerance $\varepsilon>0$. The following steps are repeated using MATLAB software for $l=1, 2, \ldots$ The Gath–Geva algorithm is same as that of GK, only the computation of distance norm involves an exponential term and thus decreases faster than the inner-product norm. The difference between the fuzzy covariance matrix F_i in GK algorithm and the F_i defined above is that the latter does not involve the weighting exponent m, instead of this it consists of $w=1$.

This clustering algorithm can detect clusters of varying shapes, sizes and densities. The cluster covariance matrix is used in conjunction with an "exponential" distance, and the clusters are not constrained in volume. However, this algorithm is less robust in the sense that it needs a good initialization, since due to the exponential distance norm, it converges to a near local optimum [45].

5. Results and discussion

5.1. *Cluster validation indices*

After clustering results from a clustering algorithm are obtained, it is important to validate if it accurately presents the actual structure of data. Cluster validity indexes can be used to evaluate the fitness of data partitions produced by a clustering algorithm [55]. Cluster validation refers to the problem whether a found partition is correct and how to measure the correctness of a partition. A clustering algorithm is designed to parameterize clusters in a way that it gives the best. One can distinguish two main approaches to determine the correct number of clusters in the data:

- Start with a sufficiently large number of clusters, and then reducing this number by combining clusters that have the same properties.
- Cluster the data for different values of c and validate the correctness of the obtained clusters with validation measures.

To be able to perform the second approach, validation indices have to be designed. Validity indices are usually independent of clustering algorithms [56]. Many cluster validity indexes for clustering algorithms had been proposed in the literature. The first proposed cluster validity index was the partition coefficient (PC) [57–59]. Subsequently, partition entropy (PE) [58], normalization of PC and PE [60–63], and performance measure [64] were proposed. The separation coefficient proposed by Gunderson [65] seems to be the first validity index that explicitly takes into account the data geometrical properties. Indexes in this class include another parts of cluster validity indexes for clustering. These include partition density (PD) [54], XB [66], and ratio fuzzy separation/fuzzy compactness (SC) [67].

5.1.1. *Partition coefficient (PC)*

Partition coefficient [57] measures the amount of overlapping between clusters. Since the overlapping does not directly indicate the correlation of the clusters, this criterion can only be partly used. It is defined as follows:

$$\mathrm{PC}(c) = \frac{1}{N}\sum_{i=1}^{c}\sum_{j=1}^{N}(\mu_{ij})^2 \tag{7}$$

where μ_{ij} is the membership of data point j in cluster i. The optimal number of cluster is at the maximum value. This value indicates the compactness of the cluster and is between 0 and 1. In the case of PC = 1, the result of the clustering completely belongs to the crisp classification. A larger value of PC is preferable for better clustering. The disadvantages of the partition coefficient are its monotonic decrease with k and the lack of connection to data shape.

5.1.2. *Classification entropy (CE)*

Classification entropy [58] measures the fuzziness of the cluster partition only, which is similar to the partition coefficient.

$$\mathrm{CE}(c) = -\frac{1}{N}\sum_{i=1}^{c}\sum_{j=1}^{N}\log(\mu_{ij}) \tag{8}$$

Both partition coefficient and classification entropy are computed using only the elements of the membership matrix. When various clusterings techniques are evaluated, the closer the PC index approaches 1 and the more the CE index approaches 0, the better the clustering is judged to be. The PC and the CE indices are sensitive to the parameter c. The closer the PC index approaches $1/c$, the fuzzier the results are. The PC index has the disadvantage that its function continuously diminishes with regard to c and has no direct relationship with the data. The PC index does display, however, the extent to which clusters overlap each other. The CE index has similar disadvantages. As the cluster number increases, the PC index value decreases while CE index value increases. Both the PC

Please cite this article in press as: A.T. Azar, et al., Fuzzy and hard clustering analysis for thyroid disease, Comput. Methods Programs Biomed. (2013), http://dx.doi.org/10.1016/j.cmpb.2013.01.002

and CE indices estimate, up to a point, the extent to which clusters overlap.

5.1.3. *Partition index (SC)*

Partition index is the ratio of the sum of compactness and separation of the clusters. It is a sum of individual cluster validity measures normalized through division by the fuzzy cardinality of each cluster [68].

$$SC(c) = \sum_{i=1}^{c} \frac{\sum_{j=1}^{N} (\mu_{ij})^m ||x_j - v_i||^2}{N_i \sum_{k=1}^{c} ||v_k - v_i||^2} \quad (9)$$

SC is useful when comparing different partitions having equal number of clusters. A lower value of SC indicates a better partition.

5.1.4. *Separation index (S)*

On the contrary of partition index (SC), the separation index uses a minimum-distance separation for partition validity [68]. A small separation index indicates a valid optimal partition.

$$S(c) = \frac{\sum_{i=1}^{c} \sum_{j=1}^{N} (\mu_{ij})^2 ||v_j - v_i||^2}{N \min_{i,k} ||v_k - v_i||^2} \quad (10)$$

5.1.5. *Xie and Beni's index (XB)*

Xie and Beni's index [66] combines the properties of membership degree and the geometric structure of dataset. Xie measures overall average compactness against separation. Smaller Xie means more compact and better separated clusters. This index decreases monotonically when the number of clusters is very large. It aims to quantify the ratio of the total variation within clusters and the separation of clusters.

$$XB(c) = \frac{\sum_{i=1}^{c} \sum_{j=1}^{N} (\mu_{ij})^m ||x_j - v_i||^2}{N \min_{i,j} ||x_j - v_i||^2} \quad (11)$$

In this equation, the numerator is the sum of the compactness of each fuzzy cluster and the denominator is the minimal separation between fuzzy clusters. The optimal fuzzy partition is obtained by minimizing XB with respect to $c = 2, \ldots, c_{max}$.

5.1.6. *Dunn's index (DI)*

Dunn [69] proposed validity index for crisp clustering. Let there be a data set with n data objects $X = \{x_j; j = 1, \ldots, n\}$ partitioned into k clusters (C_1, $C_2, \ldots, C_k$); each cluster has a centroid v_i ($i = 1, 2, \ldots, k$). The Dunn's measure DI is defied as this index is originally proposed to use at the identification of "compact and well separated clusters". So the result of the clustering has to be recalculated as it was a hard partition algorithm.

$$DI(c) = \min_{i \in c} \left[\min_{j \in c, i \neq j} \left\{ \frac{\min_{x \in C_i, y \in C_j} d(x, y)}{\max_{k \in c} \{\max_{x,y \in C} d(x, y)\}} \right\} \right] \quad (12)$$

where d is a distance function and C_i is the set whose elements are assigned to the ith cluster. The main drawback of Dunn's index is computational complexity since calculating becomes computationally very expansive as c and N increase.

5.1.7. *Alternative Dunn's index (ADI)*

The aim of modifying the original Dunn's index was that the calculation becomes more simple, when the dissimilarity function between two clusters ($\min x \in C_i, y \in C_j\ d(x,y)$) is rated in value from beneath by the triangle-nonequality:

$$d(x, y) \geq |d(y, v_j) - d(x, v_j)| \quad (13)$$

where v_j is the cluster center of the jth cluster.

$$ADI(c) = \min_{j \in c, i \neq j} \left\{ \frac{\min_{x_i \in C_i, x_j \in C_j} |d(y, v_j) - d(x_i, v_j)|}{\max_{k \in c} \{\max_{x,y \in C} d(x, y)\}} \right\} \quad (14)$$

Note, that the only difference of SC, S and XB is the approach of the separation of clusters. In the case of overlapped clusters the values of DI and ADI are not really reliable because of re-partitioning the results with the hard partition method.

5.2. *Experimental results*

5.2.1. *Validation measures for the K-medoid algorithm*

Firstly, the optimal number of clusters has to be defined. To find the optimal number of clusters, a process called elbow criterion is used. The elbow criterion is a common rule of thumb to determine what number of clusters should be chosen. The elbow criterion says that one should choose a number of clusters so that adding another cluster does not add sufficient information. More precisely, by graphing a validation measure explained by the clusters against the number of clusters, the first clusters will add much information (explain a lot of variance), but at some point the marginal gain will drop, giving an angle in the graph (the elbow). Unfortunately, this elbow cannot always be unambiguously identified. To demonstrate the working of the elbow criterion, the feature values that represent thyroid disease, as described in Section 3, are used as input for the cluster algorithms. Several runs need to be carried out with a different number of clusters being specified for each run (between 2 and 11), so as to establish the optimum number of clusters. The results of the seven validation indices for each run of *K*-medoid algorithms are shown in Table 6.

The values of the validation methods depending on the number of clusters will be plotted. The value of the partition coefficient is for all clusters 1, and the classification entropy is always 'NaN'. This is caused by the fact that these two measures were designed for fuzzy partitioning methods, and in this case the hard partitioning algorithm *K*-medoid is used. In Fig. 1, the values of the partition index and separation index are shown.

Mention again, that no validation index is reliable only by itself. Therefore, the optimal result is only clear when all of the validation indices are plotted and examined in comparison with each other. This means that the optimum only could be detected by the comparison of all the results. To find the optimal number of cluster, partitions with fewer clusters are considered better, when the difference between the values of the validation measure is small. Fig. 1 shows that for the SC and S, the number of clusters easily could be rated to 3 and 4, clusters respectively. In Fig. 2 there are more informative plots shown. The Dunn's index and alternative Dunn's index confirms that the optimal number of clusters should be chosen to

Please cite this article in press as: A.T. Azar, et al., Fuzzy and hard clustering analysis for thyroid disease, Comput. Methods Programs Biomed. (2013), http://dx.doi.org/10.1016/j.cmpb.2013.01.002

Table 6 – Validation measures for K-medoid.

c	PC	CE	SC	S	XP	DI	ADI
2	1	NaN	2.4909	0.0116	Inf	0.0262	0.0434
3	1	NaN	1.0889	0.0083	Inf	0.0149	0.0035
4	1	NaN	0.8571	0.0065	Inf	0.0199	0.0080
5	1	NaN	0.7974	0.0051	Inf	0.0177	0.0012
6	1	NaN	0.7972	0.0064	Inf	0.0218	0.0018
7	1	NaN	0.6917	0.0051	Inf	0.0213	0.0019
8	1	NaN	0.7865	0.0061	Inf	0.0242	0.0023
9	1	NaN	0.4986	0.0038	Inf	0.0067	7.93e−4
10	1	NaN	0.5786	0.0046	Inf	0.0067	0.0011
11	1	NaN	0.6821	0.0058	Inf	0.0238	4.92e−4

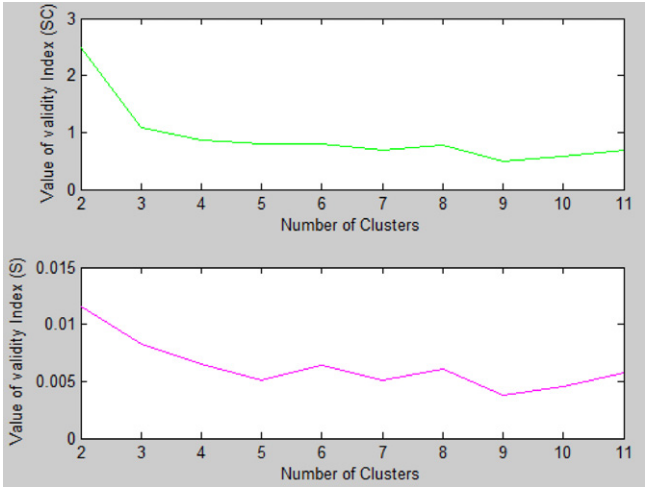

Fig. 1 – Partition index (SC) and separation index (S) for K-medoid validation.

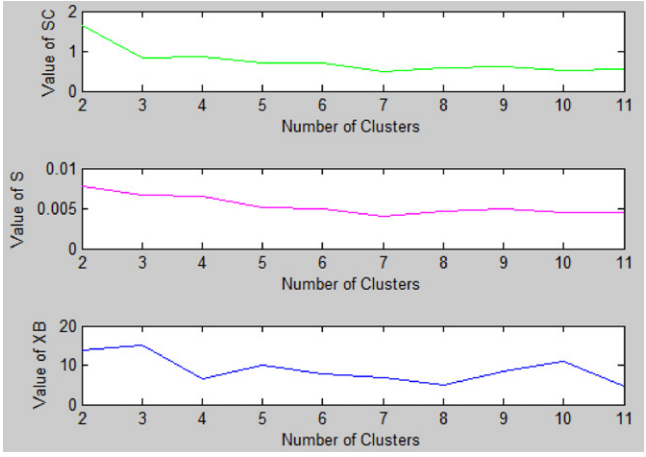

Fig. 3 – Partition index (SC), separation index (S) and XP index for K-means validation.

3 clusters. According to these indices, the best partitioning of the data according to K-medoid algorithm is achieved with 3 clusters.

5.2.2. *Validation measures for the K-means algorithm*

The numeric validity measures of the all the seven indexes for K-means are shown in Table 7. Different numbers of clusters were carried out so as to establish the optimum number of clusters where $c \in [2\ 11]$. As one can see in Tables 6 and 7, PC and CE are useless for K-means and K-medoid, while they are hard clustering methods. But that is the reason for the best results in SC, S, DI (and ADI), which are useful to validate crisp and well separated clusters.

Fig. 3 shows that for the SC and S, the number of clusters easily could be rated to 3 clusters. For the Xie and Beni index, the number should be chosen as 4 clusters.

In Fig. 4, the Dunn's index and the alternative Dunn's index confirm that the optimal number of clusters for the K-means algorithm should be chosen to 4. According to these indices,

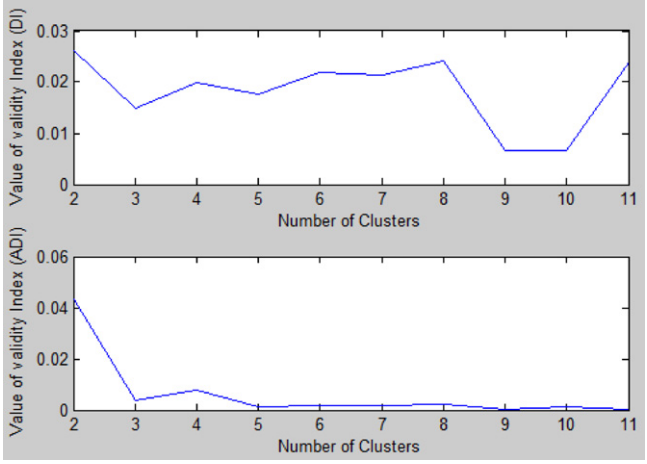

Fig. 2 – Dunn's index (DI) and alternative Dunn's index for K-medoid validation.

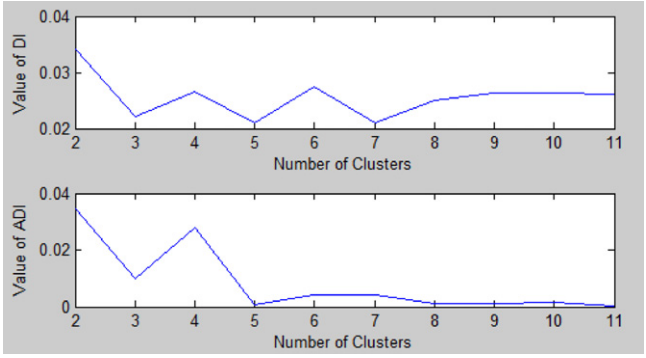

Fig. 4 – Dunn's index (DI) and alternative Dunn's index for K-means validation.

Table 7 – Validation measures for K-means.

c	PC	CE	SC	S	XP	DI	ADI
2	1	NaN	1.6482	0.0077	13.7280	0.0342	0.0346
3	1	NaN	1.0893	0.0077	11.2639	0.0101	0.0208
4	1	NaN	0.8427	0.0063	7.4240	0.0265	0.0279
5	1	NaN	0.7130	0.0049	8.4157	0.0227	4.26e−4
6	1	NaN	0.7114	0.0054	7.0513	0.0238	0.0052
7	1	NaN	0.6633	0.0053	7.7063	0.0218	0.0037
8	1	NaN	0.5227	0.0041	5.1586	0.0264	7.39e−4
9	1	NaN	0.5358	0.0041	7.2344	0.0273	4.06e−4
10	1	NaN	0.4857	0.0039	11.5008	0.0305	6.51e−4
11	1	NaN	0.5950	0.0052	3.8591	0.0378	4.06e−4

the best partitioning of the data according to K-means algorithm is achieved with 3 clusters.

5.2.3. *Validation measures for the fuzzy C-means algorithm*

It is also possible to define the optimal numbers of clusters for fuzzy clustering algorithms with elbow criterion. To illustrate this, the results of the fuzzy C-means algorithm are shown in Table 8. In Fig. 5 the results of the partition index and the classification entropy are plotted. Compared to the hard clustering methods, the validation methods can be used now for the fuzzy clustering. However, the main drawback of PC is the monotonic decreasing with c, which makes it hardly to detect the optimal number of cluster. The same problem holds for CE: monotonic increasing caused by the lack of direct connection to the data. The optimal number of cluster cannot be rated based on those two validation methods. On the score of Fig. 5, the number of clusters can be only rated to 3.

Fig. 6 gives more information about the optimal number of clusters. For the SC, the local minimum is reached at $c = 3$ while for S the local minimum is reached at $c = 4$. For the XB index, it is difficult to find the optimal number of clusters. The points at $c = 3$, $c = 6$ and $c = 10$, can be seen as an elbow. In Fig. 7, the Dunn's index also indicates that the optimal number of clusters should be at $c = 3$. On the other hand, the alternative Dunn's index, has an elbow at the point $c = 5$. However, for the alternative Dunn's index is not known how reliable its results

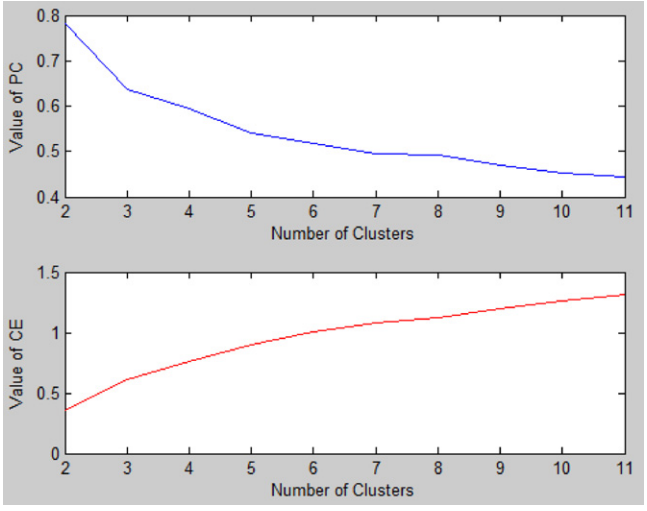

Fig. 5 – Partition coefficient (PC) and classification entropy (CE) where $m = 2.0$ for FCM validation.

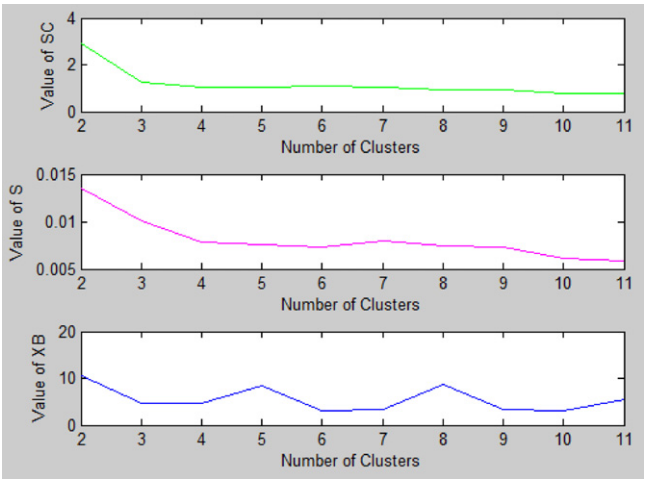

Fig. 6 – Partition index (SC), separation index (S) and XP index ($m = 2.0$) for FCM validation.

are, so the optimal number of clusters for the FCM algorithm is chosen to be 3.

5.2.4. *Validation measures for the Gustafsone–Kessel (GK) and Gath–Geva (GG) algorithms*

The Gustafsone–Kessel algorithm ran for $m = 2$. The validation indices results concern clustering of 2–11 clusters are depicted in Table 9.

Fig. 8 displays a slight change in point 5 for the PC and CE indices. In Fig. 9, the SC and S indices the local minimum is reached at $c = 3$, whilst value 4 emerges from the graph for

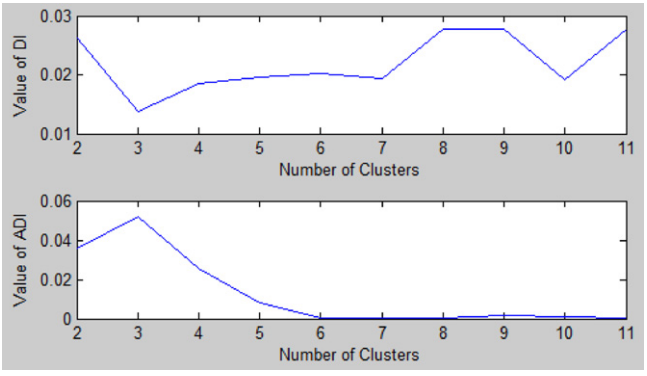

Fig. 7 – Dunn's index (DI) and alternative Dunn's index ($m = 2.0$) for FCM validation.

Table 8 – Validation measures for FCM and for $m = 2.00$.

c	PC	CE	SC	S	XP	DI	ADI
2	0.7816	0.3556	2.8861	0.0134	10.5093	0.0262	0.0358
3	0.6381	0.6116	1.2580	0.0010	4.6779	0.0137	0.0520
4	0.5930	0.7561	1.0401	0.0079	4.6081	0.0184	0.0258
5	0.5397	0.9034	1.0417	0.0076	8.2816	0.0196	0.0083
6	0.5173	1.0004	1.0724	0.0073	3.0445	0.0202	5.82e−4
7	0.4956	1.0828	1.0296	0.0080	3.2118	0.0193	5.799e−4
8	0.4913	1.1230	0.9438	0.0074	8.6896	0.0278	4.942e−4
9	0.4689	1.2009	0.9348	0.0073	3.2180	0.0278	0.0018
10	0.4538	1.2642	0.7908	0.0062	3.1536	0.0191	7.22e−4
11	0.4441	1.3096	0.7585	0.0059	5.5618	0.0278	1.23e−4

Table 9 – Validation measures for GK where $m = 2$.

c	PC	CE	SC	S	XP	DI	ADI
2	0.7341	0.4183	8.4063	0.0391	5.3426	0.0154	0.0384
3	0.6486	0.6067	1.5087	0.0109	7.9546	0.0116	0.0083
4	0.6178	0.7181	1.3470	0.0089	3.8776	0.0130	0.0131
5	0.5571	0.8855	1.0877	0.0085	6.2939	0.0184	0.0032
6	0.5815	0.8717	0.9946	0.0071	4.8017	0.0125	9.255e−4
7	0.5445	0.9823	1.0598	0.0082	2.7295	0.0037	4.762e−4
8	0.5615	0.9588	1.2802	0.0095	4.3080	0.0097	6.798e−4
9	0.5112	1.0976	1.0737	0.0081	2.7321	0.0126	9.33e−5
10	0.4863	1.1738	0.7549	0.0059	2.6376	0.0187	2.2113e−4
11	0.4957	1.1744	1.0659	0.0083	2.9757	0.0120	3.25e−4

the XB index. In Fig. 10, the Dunn's index and the alternative Dunn's index confirm that the optimal number of clusters for the GK algorithm should be chosen to 3. According to these indices and considering that SC and S are more useful, when comparing different clustering methods with the same c, the best partitioning of the data for Gustafsone–Kessel algorithm is achieved with 3 clusters.

According to the validity indices, the best partitioning of the data for Gath–Geva clustering algorithm is achieved with 3 clusters as shown in Table 10. The Gath–Geva clustering algorithm has the same outputs defined at the description of *K*-means and GK-clust, but it has less input parameters (only the weighting exponent and the termination tolerance), because the used distance norm involving the exponential term cannot run into numerical problems.

Table 10 – Validation measures for GG where $m = 2$ and $c = 3$.

Index	Value
PC	0.9608
CE	0.0716
SC	2.6238
S	0.0253
XP	1.17875
DI	0.0437
ADI	0.0071

The optimal number of cluster can be determined with the validation methods, as mentioned in the previous section. The validation measures can also be used to compare the different cluster methods. As examined in the previous section,

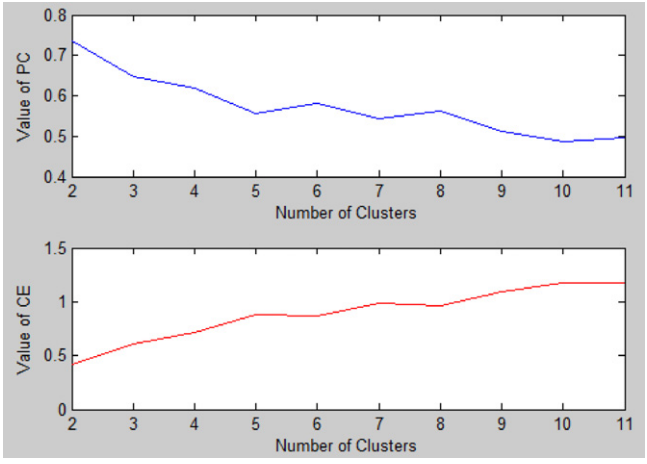

Fig. 8 – Partition coefficient (PC) and classification entropy (CE) where $m = 2.0$ for GK algorithm validation.

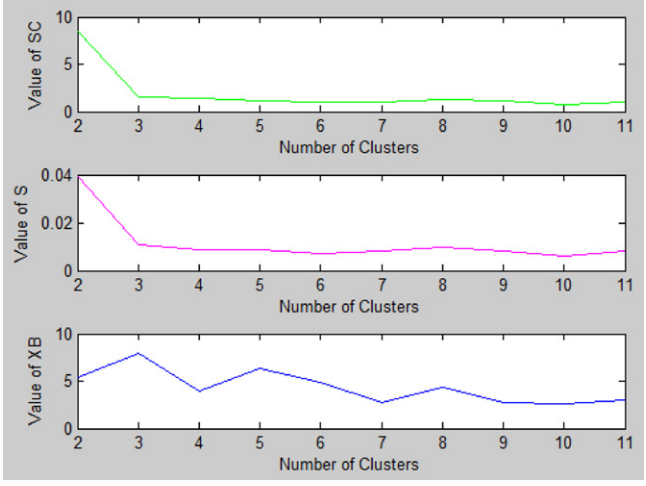

Fig. 9 – Partition index (SC), separation index (S) and XP index ($m = 2.0$) for GK algorithm validation.

Please cite this article in press as: A.T. Azar, et al., Fuzzy and hard clustering analysis for thyroid disease, Comput. Methods Programs Biomed. (2013), http://dx.doi.org/10.1016/j.cmpb.2013.01.002

Table 11 – Comparison of fuzzy clustering algorithms for c = 3.

Index	PC	CE	SC	S	XP	DI	ADI
K-medoid	1	NaN	**1.0889**	0.0083	Inf	0.0149	**0.0035**
K-means	1	NaN	1.0893	0.0077	11.2639	**0.0101**	0.0208
FCM	**0.6381**	**0.6116**	1.2580	**0.0010**	4.6779	0.0137	0.0520
GK	0.6486	0.6067	1.5087	0.0109	7.9546	0.0116	0.0083
GG	0.9608	0.0716	2.6238	0.0253	**1.17875**	0.0437	0.0071

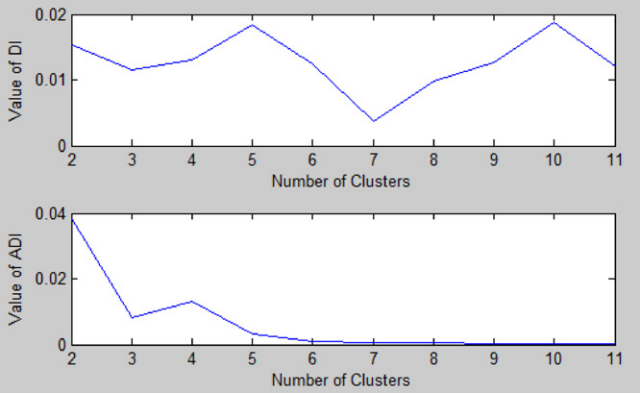

Fig. 10 – Dunn's index (DI) and alternative Dunn's index ($m = 2.0$) for GK algorithm validation.

the optimal number of all clustering algorithms is achieved with 3 clusters. In order to access the performance of the five algorithms for the partitioning of the dataset into 3 clusters, comparison of clustering algorithms will be presented in the next section.

5.3. *Comparing the clustering algorithms*

In order to access the performance of the five cluster algorithms, the validation measures can also be used to compare the different cluster methods. Table 11 shows that the PC and CE are useless for *K*-means and *K*-medoid while they are hard clustering methods. But that is the reason for the best results in SC, S, DI and ADI, which are useful to validate crisp and well separated clusters (see bold values).

As noted from Table 11 regarding the value of the SC index, which is regarded as the strongest criterion of the noise and of the separation of clusters in the comparison of various clustering methods, the *K*-medoid algorithm yields better values than other algorithms (see bold values). On the score of the values of the alternative Dunn's index, one can conclude that for c = 3 the *K*-medoid algorithm has the best results.

The fuzzy *C*-means algorithm displays smaller values of S and PC indices than other methods. Also, the CE index gives greater values for fuzzy *C*-means algorithm than other methods and it produces clusters with less overlap (Fig. 12a). Based on the CE index, the fuzzy *C*-means algorithm yields slightly better differentiated clusters. The Gath–Geva clustering algorithm has the smallest values of the Xie and Beni's index and performed much better in comparison with other methods according to XP index.

The clustering techniques are then visualized by the Sammon mapping method [70]. Sammon's non-linear mapping algorithm [71] attempts to find a low-dimensional (normally 2D or 3D) representation of a set of points distributed in a high dimensional pattern space such that the Euclidean distances between the points on the map are as similar as possible to the Euclidean distances between the corresponding points in the high-dimensional pattern space. Every cluster center is represented by a single point, in projected two-dimensional space, and is thus independent of the form of the original cluster prototype. The resulting visualization depicts clusters in input space as groups of data points mapped close to each other in the output plane [18]. In the case of fuzzy clustering, the projected clusters should be separated and not present overlapping to a great extent. Furthermore, data in a properly selected cluster prototype should fall closely to the projected cluster centers, resulting in approximately spherical (fuzzy *C*-means) or ellipsoidal (Gustafsone–Kessel)

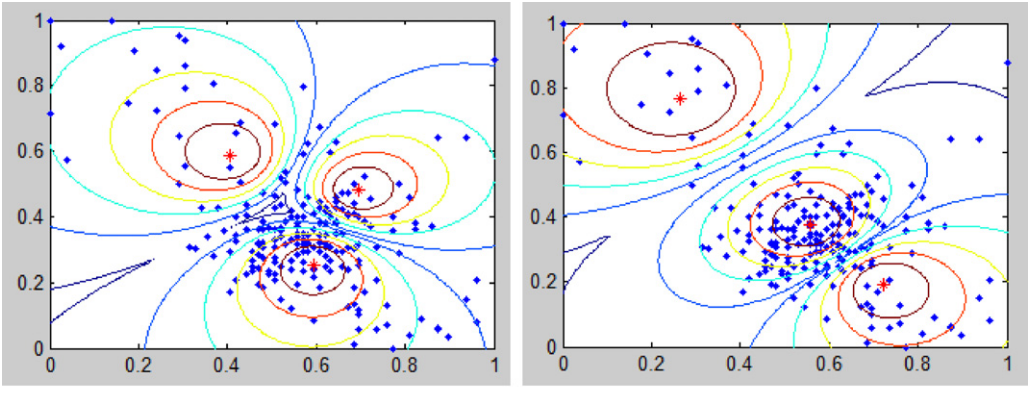

Fig. 11 – (a) Result of *K*-medoid algorithm and (b) result of *K*-means algorithm.

Please cite this article in press as: A.T. Azar, et al., Fuzzy and hard clustering analysis for thyroid disease, Comput. Methods Programs Biomed. (2013), http://dx.doi.org/10.1016/j.cmpb.2013.01.002

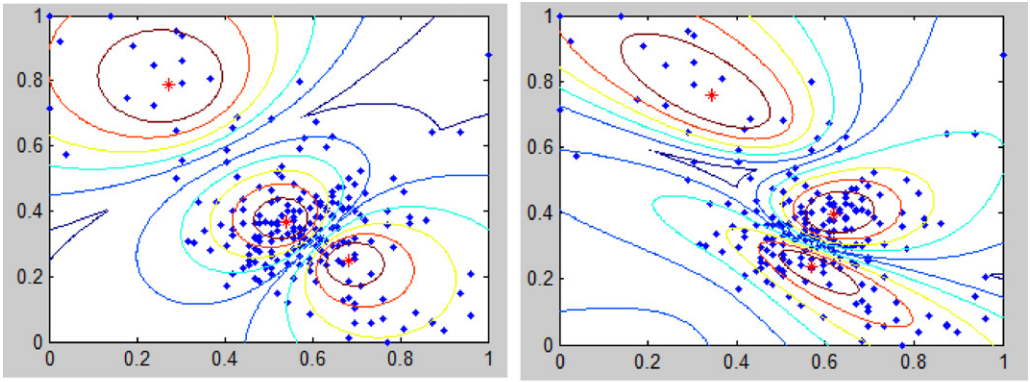

Fig. 12 – (a) Result of FCM algorithm and (b) result of Gustafsone–Kessel (GK) algorithm.

clusters. However, if all data are projected very close to cluster centers then fuzzy clustering has been turn to crisp clustering [18]. For *K*-medoid algorithm visualization (Fig. 11a), the output result is more clear and gives good results by removing noise to a considerable degree with respect to the separation of clusters than *K*-means algorithm (Fig. 11b). For *K*-means algorithm visualization, the clusters that occur display an ellipsoid structure, whilst the centers of the clusters are dispersed in such a way that the point pattern tends to be normal. Fuzzy *C*-means performed better for the thyroid disease dataset creating better-separated and meaningful clusters with high compactness. Gustafsone–Kessel (see Fig. 12) and Gath–Geva (see Fig. 13) algorithms gave satisfactory separation but not with high compactness. This is because the Gustafsone–Kessel algorithm creates ellipsoid clusters, adopting the distance norm to the local topological structure and analyzing complex datasets more efficiently. With the results of the validation methods and the visualization of the clustering, one can conclude that hard clustering methods also can find a good solution for clustering thyroid disease data set, when it is compared with the figures of fuzzy clustering algorithms.

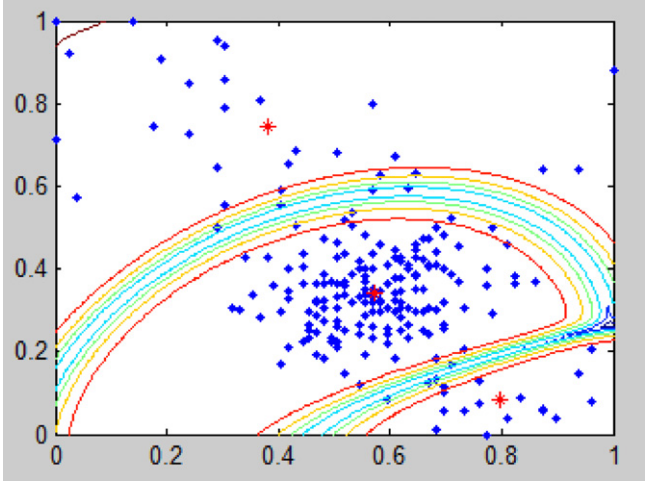

Fig. 13 – Result of Gath–Geva (GG) clustering algorithm.

5.4. *Mode of availability of software developed*

The algorithms used in this paper can be coded with the use of C++ language for hardware implementation. The key point of the implementation includes the number of clusters. This number must be fixed in a hardware implementation. The minimum number of clusters allowed is two which is valid for applications as binarization, other applications use three [72] to higher number. In hardware applications it has been fixed to two clusters leading to the smallest need of memory. Another point of the hardware implementation includes the fuzziness factor *m*. The fuzziness factor must be chosen empirically depending on the actual applications. Since the implementation of fractional exponents is such a difficult task, the 'optimum' number of *m* suitable for implementation can be obtained. The third point is the initialization of the membership matrix. In software implementation, the matrix is initialized randomly (this is the method used by the Matlab algorithm). Such a circuit would include complexity without any improvement. To solve this problem, implementation method developed by Lázaro et al. [73] of fuzzy *C*-means clustering may be used. The algorithm allows a high degree of parallelism, which makes the hardware implementation suited for real-time applications.

6. Conclusion

Thyroid disease clustering is an important classification problem. In this study, thyroid disease data set is clustered based on crisp and fuzzy algorithms. Different scalar validity indexes as partition coefficient (PC), classification entropy (CE), partition index (SC), separation index (S), Xie and Beni's index (XB), Dunn's index (DI) and alternative Dunn's index (DII) are used in performances analysis and comparison of the proposed methods. Several runs are carried out with a different number of clusters (between 2 and 11), so as to establish the optimum number of clusters. The results of the seven validation indices for each run of the clustering methods are recorded. To find the optimal number of clusters, the so-called elbow criterion is applied. The experimental results revealed that for all algorithms, the elbow was located at $c = 3$. The comparison results

Please cite this article in press as: A.T. Azar, et al., Fuzzy and hard clustering analysis for thyroid disease, Comput. Methods Programs Biomed. (2013), http://dx.doi.org/10.1016/j.cmpb.2013.01.002

disclose that the K-medoid algorithm yields better SC index values, which is regarded as the strongest criterion of the noise and of the separation of clusters in the comparison of various clustering methods, than other algorithms. On the score of the values of the alternative Dunn's index, one can conclude that for $c = 3$ the K-medoid algorithm has the best results. Based on the CE index, the fuzzy C-means algorithm yields slightly better differentiated clusters. The Gath–Geva clustering algorithm performs much better in comparison with other methods according to XP index. The clustering results for all algorithms are then visualized by the Sammon mapping method to find a low-dimensional (normally 2D or 3D) representation of a set of points distributed in a high dimensional pattern space. The visualization results show that the fuzzy C-means performed better for the thyroid disease dataset creating better-separated and meaningful clusters with high compactness. Gustafsone–Kessel, and Gath–Geva algorithms gave satisfactory separation but not with high compactness. With the results of the validation methods and the visualization of the clustering, it is clear that hard clustering methods also can find a good solution for clustering thyroid disease data set, when it is compared with the figures of fuzzy clustering algorithms. Based on this research's experiments, it is possible to formulate some recommendations. The first recommendation is to use different feature values for thyroid disease. This can lead to different clusters and thus different segments. To improve determining the actual number of clusters present in the data set, the application of more specialized methods than the elbow criterion could be applied. An interesting alternative is, for instance, the application of evolutionary algorithms. Another way of improving this research is to extent the number of cluster algorithms like hierarchical clustering, main shift clustering or mixture of Gaussians.

Acknowledgement

The authors would like to highly appreciate and gratefully acknowledge, UCI Machine Learning Repository [http://archive.ics.uci.edu/ml] [74] for obtaining the thyroid disease data set.

REFERENCES

[1] T.M. Mitchell, Machine Learning, McGraw Hill, New York, 1997.
[2] A.K. Jain, R.C. Dubes, Algorithms for Clustering Data, Prentice-Hall, Englewood Cliffs, NJ, 1988.
[3] A.K. Jain, Data clustering: 50 years beyond K-means, Pattern Recognition Letters 31 (2010) 651–666.
[4] H. Frigui, R. Krishnapuram, A robust competitive clustering algorithm with applications in computer vision, IEEE Transactions on Pattern Analysis and Machine Intelligence 21 (1999) 450–465.
[5] A.K. Jain, M.N. Murthy, P.J. Flynn, Data clustering: a review, ACM Computing Surveys 31 (1999) 264–323.
[6] Y. Leung, J. Zhang, Z. Xu, Clustering by space–space filtering, IEEE Transactions on Pattern Analysis and Machine Intelligence 22 (2000) 1396–1410.
[7] B. Everitt, Cluster Analysis, Halsted, New York, 2001.
[8] G. Karypis, E.H. Han, V. Kumar, CHAMELEON: a hierarchical clustering algorithm using dynamic modeling, IEEE Computing 32 (1999) 68–75.
[9] S.M. Savaresi, D.L. Boley, S. Bittanti, G. Gazzaniga, Cluster selection in divisive clustering algorithms, in: Proceedings of the 2nd SIAM ICDM, Arlington, VA, 2002, pp. 299–314.
[10] R. Duda, P. Hart, Pattern Classification and Scene Analysis, Wiley Interscience, New York, 1973.
[11] J.B. MacQueen, Some methods for classification and analysis of multivariate observations, in: Proceedings of the 5th Berkeley Symposium on Mathematical Statistics and Probability, vol. 1, University of California Press, 1967, pp. 281–297 http://en.wikipedia.org/wiki/Mathematical_Reviews
[12] L. Kaufman, P.J. Rousseeuw, Clustering by means of medoids, in: Y. Dodge (Ed.), Statistical Data Analysis Based on the L1-Norm and Related Methods, Amsterdam, North-Holland, 1987, pp. 405–416.
[13] L. Kaufman, P.J. Rousseeuw, Finding Groups in Data: An Introduction to Cluster Analysis, Wiley, New York, 1990.
[14] F.A. De Carvalho, C.P. Tenório, Fuzzy K-means clustering algorithms for interval-valued data based on adaptive quadratic distances, Fuzzy Sets and Systems 161 (2010) 2978–2999.
[15] M. Sato-Ilic, Symbolic clustering with interval-valued data, Procedia Computer Science 6 (2011) 358–363.
[16] C.W.D. De Almeida, R.M.C.R. De Souza, A.L.B. Candeias, Fuzzy kohonen clustering networks for interval data, Neurocomputing (2012), http://dx.doi.org/10.1016/j.neucom.2012.06.019.
[17] H. Zhaoa, Z. Xu, S. Liu, Z. Wang, Intuitionistic fuzzy MST clustering algorithms, Computers and Industrial Engineering 62 (2012) 1130–1140.
[18] G. Grekousis, H. Thomas, Comparison of two fuzzy algorithms in geodemographic segmentation analysis: the fuzzy C-means and Gustafsone–Kessel methods, Applied Geography 34 (2012) 125–136.
[19] S. Openshaw, A. Gillard, On the stability of a spatial classification of census enumeration districts data, in: P.W.S. Batey (Ed.), Theory and Methods in Urban and Regional Analysis, Pion, London, 1978, pp. 101–119.
[20] D. Michie, D.J. Spiegelhalter, C.C. Taylor, Machine Learning Neural and Statistical Classification, Oxford University Press, Oxford, 1994.
[21] K. Hoshi, J. Kawakami, M. Kumagai, S. Kasahara, N. Nisimura, H. Nakamura, et al., An analysis of thyroid function diagnosis using Bayesian-type and SOM-type neural networks, Chemical and Pharmaceutical Bulletin 53 (2005) 570–1574.
[22] L. Ozyılmaz, T. Yıldırım, Diagnosis of thyroid disease using artificial neural network methods, in: Proceedings of ICONIP'02 9th International Conference on Neural Information Processing, Orchid Country Club, Singapore, 2002, pp. 2033–2036.
[23] K. Polat, S. Sahan, S. Gunes, A novel hybrid method based on artificial immune recognition system (AIRS) with fuzzy weighted preprocessing for thyroid disease diagnosis, Expert Systems with Applications 32 (2007) 1141–1147.
[24] S.P. Yip, G.I. Webb, Empirical function attribute construction in classification learning, in: Joint Conference on Artificial Intelligence (AI'94), 1994, pp. 29–36.
[25] G. Serpen, H. Jiang, L. Allred, Performance analysis of probabilistic potential function neural network classifier, in: Proceedings of Artificial Neural Networks in Engineering Conference, vol. 7, St. Louis, MO, 1997, pp. 471–476.
[26] G. Zhang, L.V. Berardi, An investigation of neural networks in thyroid function diagnosis, Health Care Management Science 1 (1998) 29–37.

Please cite this article in press as: A.T. Azar, et al., Fuzzy and hard clustering analysis for thyroid disease, Comput. Methods Programs Biomed. (2013), http://dx.doi.org/10.1016/j.cmpb.2013.01.002

[27] T.S. Cheong, C.H. Yoon, memory based classifier using the recursive partition averaging, in: Proceedings of the IEEE Region 10 Conference Tencon, vol. 2, 1999, pp. 1038–1041.
[28] L. Pasi, Similarity classifier applied to medical data sets, in: International Conference on Soft Computing, Helsinki, Finland & Gulf of Finland & Tallinn, Estonia, 2004.
[29] A.J. Myles, S.D. Brown, Decision pathway modeling, Journal of Chemometrics 18 (2004) 286–293.
[30] R. Hassan, B. Nath, M. Kirley, A data clustering algorithm based on single hidden Markov model, in: Proceedings of the International Multiconference on Computer Science and Information Technology, 2006, pp. 57–66.
[31] M. Pechenizkiy, A. Tsymbal, S. Puuronen, D.W. Patterson, Feature extraction for dynamic integration of classifiers, Fundamenta Informaticae 77 (2007) 243–275.
[32] Y. Sun, M.S. Kamel, A.K.C. Wong, Y. Wang, Cost-sensitive boosting for classification of imbalanced data, Pattern Recognition 40 (2007) 3358–3378.
[33] A. Keles, A. Keles, ESTDD: expert system for thyroid diseases diagnosis, Expert Systems with Applications 34 (2008) 242–246.
[34] P. Kukkurainen, P. Luukka, Classification method using fuzzy level set subgrouping, Expert Systems with Applications 34 (2008) 859–865.
[35] F. Temurtas, A comparative study on thyroid disease diagnosis using neural networks, Expert Systems with Applications 36 (2009) 944–949.
[36] H. Kodaz, S. Ozsen, A. Arslan, S. Gunes, Medical application of information gain based artificial immune recognition system (AIRS): diagnosis of thyroid disease, Expert Systems with Applications 36 (2009) 3086–3092.
[37] E. Dogantekin, A. Dogantekin, D. Avci, An automatic diagnosis system based on thyroid gland: ADSTG, Expert Systems with Applications 37 (2010) 6368–6372.
[38] E. Dogantekin, A. Dogantekin, D. Avci, An expert system based on generalized discriminant analysis and wavelet support vector machine for diagnosis of thyroid diseases, Expert Systems with Applications 38 (2011) 146–150.
[39] H.L. Chen, B. Yang, G. Wang, J. Liu, Y.D. Chen, D.Y. Liu, A three-stage expert system based on support vector machines for thyroid disease diagnosis, Journal of Medical Systems 36 (2012) 1953–1963, http://dx.doi.org/10.1007/s10916-011-9655-8.
[40] D.Y. Liu, H.L. Chen, B. Yang, X. En Lv, L.N. Li, Design of an enhanced fuzzy k-nearest neighbor classifier based computer aided diagnostic system for thyroid disease, Journal of Medical Systems (2011), http://dx.doi.org/10.1007/s10916-011-9815-x.
[41] L.N. Li, J.H. Ouyang, H.L. Chen, D.Y. Liu, A computer aided diagnosis system for thyroid disease using extreme learning machine, Journal of Medical Systems (2012), http://dx.doi.org/10.1007/s10916-012-9825-3.
[42] AAO-HNS: The American Academy of Otolaryngology-Head and Neck Surgery, 2012. http://www.entnet.org/Health Information/Thyroid-Disorders.cfm (accessed June 2012).
[43] J. Hartigan, Clustering Algorithms, Wiley, New York, 1975.
[44] P.A. Devijver, J. Kittler, Pattern Recognition: A Statistical Approach, Prentice-Hall, London, 1982.
[45] B. Balasko, J. Abonyi, B. Feil, Fuzzy Clustering and Data Analysis Toolbox: For Use with MATLAB, MathWorks, Inc., 2005, pp. 1–74.
[46] P. Drineas, A. Frieze, R. Kannan, S. Vempala, V. Vinay, Clustering large graphs via the singular value decomposition, Machine Learning 56 (1999) 9–33.
[47] J. Mao, A.K. Jain, A self-organizing network for hyper-ellipsoidal clustering (HEC), IEEE Transactions on Neural Networks 7 (1996) 16–29.
[48] P.S. Lai, H.C. Fu, Variance enhanced K-medoid clustering, Expert Systems with Applications 38 (2011) 764–775.
[49] R.T. Ng, J. Han, CLARANS: a method for clustering objects for spatial data mining, IEEE Transactions on Knowledge-based Data Engineering 14 (2002) 1003–1016.
[50] J.C. Dunn, A fuzzy relative of the ISODATA process and its use in detecting compact well-separated clusters, Journal of Cybernetics 3 (1973) 32–57.
[51] J.C. Bezdek, Pattern Recognition with Fuzzy Objective Function Algorithms, Kluwer Academic Publishers, Norwell, MA, USA, 1981.
[52] D.E. Gustafson, W.C. Kessel, Fuzzy clustering with fuzzy covariance matrix, in: Proceedings of the IEEE CDC, San Diego, CA, 1979, pp. 761–766.
[53] R. Babuska, P.J. Van der Veen, U. Kaymak, Improved covariance estimation for Gustafson–Kessel clustering, IEEE International Conference on Fuzzy Systems 2 (2002) 1081–1085.
[54] I. Gath, A.B. Geva, Unsupervised optimal fuzzy clustering, IEEE Transactions on Pattern Analysis and Machine Intelligence 11 (1989) 73–781.
[55] K.R. Zalik, B. Zalik, Validity index for clusters of different sizes and densities, Pattern Recognition Letters 32 (2011) 221–234.
[56] K.L. Wu, M.S. Yang, J.N. Hsieh, Robust cluster validity indexes, Pattern Recognition 42 (2009) 2541–2550.
[57] J.C. Bezdek, Numerical taxonomy with fuzzy sets, Journal of Mathematical Biology 1 (1974) 57–71.
[58] J.C. Bezdek, Cluster validity with fuzzy sets, Journal of Cybernetics 3 (1974) 58–73.
[59] J.C. Dunn, Some recent investigations of a new fuzzy partition algorithm and its application to pattern classification problem, Journal of Cybernetics 4 (1974) 1–15.
[60] M. Roubens, Pattern classification problems with fuzzy sets, Fuzzy Sets and Systems 1 (1978) 239–253.
[61] J.C. Dunn, Indices of partition fuzziness and the detection of clusters in large data sets, in: M. Gupta, G. Saridis (Eds.), Fuzzy Automata and Decision Processes, Elsevier, New York, 1976.
[62] J.C. Bezdek, M.P. Windham, R. Ehrlich, Statistical parameters of fuzzy cluster validity functionals, International Journal of Computer and Information Science 9 (1980) 323–336.
[63] R.N. Dave, Validating fuzzy partition obtained through c-shells clustering, Pattern Recognition Letters 17 (1996) 613–623.
[64] E. Backer, A.K. Jain, A clustering performance measure based on fuzzy set decomposition, IEEE Transactions on Pattern Analysis and Machine Intelligence 3 (1981) 66–74.
[65] R. Gunderson, Application of fuzzy ISODATA algorithms to star tracker pointing systems, in: Proceedings of the Seventh Triennial World IFAC Congress, Helsinki, Finland, 1978, pp. 1319–1323.
[66] X.L. Xie, G. Beni, A validity measure for fuzzy clustering, IEEE Transactions on Pattern Analysis and Machine Intelligence 13 (1991) 841–847.
[67] N. Zahid, M. Limouri, A. Essaid, A new cluster-validity for fuzzy clustering, Pattern Recognition 32 (1999) 1089–1097.
[68] A.M. Bensaid, L.O. Hall, J.C. Bezdek, L.P. Clarke, M.L. Silbiger, J.A. Arrington, R.F. Murtagh, Validity-guided (Re) clustering with applications to image segmentation, IEEE Transactions on Fuzzy Systems 4 (1996) 112–123.
[69] J. Dunn, A fuzzy relative of the ISODATA process and its use in detecting compact well separated cluster, Journal of Cybernetics 3 (1974) 32–57.

Please cite this article in press as: A.T. Azar, et al., Fuzzy and hard clustering analysis for thyroid disease, Comput. Methods Programs Biomed. (2013), http://dx.doi.org/10.1016/j.cmpb.2013.01.002

[70] J.W. Sammon, A nonlinear mapping for data structure analysis, IEEE Transactions on Computers 18 (1969) 401–409.
[71] J.A. Nelder, R. Mead, A simplex method for function minimization, Computer Journal 7 (1965) 308–313.
[72] J.-H. Xue, A. Pizurica, W. Philips, E. Kerre, R.V.D. Walle, I. Lemahieu, An integrated method of adaptive enhancement for unsupervised segmentation of MRI brain images, Pattern Recognition Letters 24 (15) (2003) 2549–2560.
[73] J. Lázaro, J. Arias, J.L. Martín, C. Cuadrado, A. Astarloa, Implementation of a modified fuzzy C-means clustering algorithm for real-time applications, Microprocessors and Microsystems 29 (2005) 375–380.
[74] A. Frank, A. Asuncion, UCI Machine Learning Repository, University of California, School of Information and Computer Science, Irvine, CA, 2010 http://archive.ics.uci.edu/ml (accessed 16.11.12).

Please cite this article in press as: A.T. Azar, et al., Fuzzy and hard clustering analysis for thyroid disease, Comput. Methods Programs Biomed. (2013), http://dx.doi.org/10.1016/j.cmpb.2013.01.002