On the Use of Variable-Size Fuzzy Clustering for Classification

Vicenç Torra[1] and Sadaaki Miyamoto[2]

[1] Institut d'Investigació en Intel·ligència Artificial,
Campus de Bellaterra, 08193 Bellaterra, Catalonia, Spain
vtorra@iiia.csic.es
[2] Department of Risk Engineering,
School of Systems and Information Engineering,
University of Tsukuba, 305-8573 Ibaraki, Japan
miyamoto@esys.tsukuba.ac.jp

Abstract. Hard c-means can be used for building classifiers in supervised machine learning. For example, in a n-class problem, c clusters are built for each of the classes. This results into $n \cdot c$ centroids. Then, new examples can be classified according to the nearest centroid.

In this work we consider the problem of building classifiers using fuzzy clustering techniques. In particular, we consider the use of fuzzy c-means, as well as some variations. Namely, fuzzy c-means with variable size and entropy based fuzzy c-means.

Keywords: Clustering, Classification, Fuzzy c-means, Variable-size fuzzy c-means, entropy-based fuzzy c-means.

1 Introduction

Clustering [6] and classification [3] are common tools in machine learning [8]. In both cases, sets of examples are considered. In supervised machine learning there is a highlighted attribute that classifies the examples into categories. This attribute determines the class of the examples. This is not the case of unsupervised machine learning. In such framework, all attributes are considered as equal, when knowledge is extracted.

Then, in supervised machine learning, tools have been developed for finding models for the relevant attribute. That is, models are built that permit to assign a class (*i.e.*, assign a value to the relevant attribute) to each new example for which such class is not known. Several different types of models exist based on different assumptions. Examples include, neural networks, (fuzzy) rule-based systems, statistical models, etc.

In unsupervised machine learning, methods have been developed to extract knowledge from the data. Clustering is one of the tools used for extracting such knowledge, as it permits to build structures in which similar objects are put together in clusters.

Besides of this use of clustering as an unsupervised machine learning tool. Clustering can also be used for supervised learning. For example, clustering has been

V. Torra et al. (Eds.): MDAI 2006, LNAI 3885, pp. 362–371, 2006.

used for building models based on the k-means [5] and for building fuzzy rules. In this latter case, clustering is used to build (fuzzy) partitions of the examples.

In this paper we will consider the use of some fuzzy clustering techniques [6, 11] and their use in a classification tool. We propose a method to be used with fuzzy c-means [2] and then we develop variations for the case of entropy-based c-means and their variations with variable size.

The approach presented here includes the advantages that, from the conceptual view, present the alternative clustering methods against fuzzy c-means, and this one with respect to k-means [11].

The structure of the paper is as follows. In Section 2 we review some clustering techniques and an approach to build classification models. Then, in Section 3 we introduce our approach for using fuzzy clustering techniques. Then, in Section 4 we present an application and the experiments performed.

2 Clustering and Classification

In this section we give a review of some aspects of clustering and classification that will be used later on in this work. First we consider a few clustering algorithms, and then we show how to build classifiers using clustering methods.

Here, we consider n examples in a given p dimensional space. We will denote these examples by $x_k \in \mathbb{R}^p$ for $k = 1, \ldots, n$. When the class of each example is known, we will denote this information as follows: $\kappa = \{\kappa_1, \ldots, \kappa_{|\kappa|}\}$ corresponds to the classes; $|\kappa|$ is the number of classes and $\kappa(x_k)$, or simply κ_k, denotes the class for example x_k.

2.1 Fuzzy Clustering

As explained in the introduction, clustering methods are to obtain a set of clusters from a set of examples. In this case, the only information considered is a set of examples x_k in a p dimensional space. Therefore, no information on the class of x_k is required here (although, as we will show latter, this information might be available).

Some of the most well-known algorithms for clustering are the *hard c-means* (also known as k-means) and the *fuzzy c-means*.

Both methods assume that we know a priori the number of clusters to be built. Such number will be denoted here by the parameter c. Then, the algorithms find a partition of the set of examples (the set $\{x_k\}$) into c different clusters. In hard c-means the partition is a classical one. That is, examples are assigned to only one cluster. Instead, in fuzzy c-means the partition is fuzzy. That is, examples can belong at the same time to different clusters. In this case, the membership to the clusters is not complete but only partial. This situation is modelled using the so-called fuzzy sets and fuzzy memberships.

Here we will use u_{ik} to denote the membership of element x_k to the i-th cluster. In the case of hard c-means as elements are either in the cluster or not in the cluster, we have that u_{ik} is either 0 or 1 (boolean membership). Moreover, as the elements can only belong to one cluster, we have that $\sum_{i=1}^{n} u_{ik} = 1$.

Instead, in fuzzy clustering we have that as membership is partial we have that u_{ik} is in the interval $[0, 1]$. In this latter case, $u_{ik} = 0$ corresponds to non-membership and $u_{ik} = 1$ corresponds to full membership to cluster i. Values in-between correspond to partial membership (the largest the value, the greatest the membership). Nevertheless, in this latter case, the constraint $\sum_{i=1}^{n} u_{ik} = 1$ is maintaned. If this equality holds for all examples k, and we have that $u_{ik} \in [0, 1]$ for all i and k, we say that u defines a fuzzy partition.

In this section we will review first the fuzzy c-means (FCM) algorithm. Then, we will consider one of its variations: fuzzy c-means with variable size (VFCM). And finally, we will also describe an alternative method for fuzzy clustering known as entropy-based fuzzy c-means (EFCM).

Most fuzzy clustering algorithms are defined in terms of a minimization problem with some constraints. In the case of fuzzy c-means [2, 9], the minimization problem is the following one:

$$J_{FCM}(U, V) = \sum_{i=1}^{c} \sum_{k=1}^{n} (u_{ik})^m ||x_k - v_i||^2 \tag{1}$$

with constraints:

- $u_{ik} \in [0, 1]$
- $\sum_{i=1}^{c} u_{ik} = 1$ for all k

For conciseness, we will denote the values u that satisfy these two constraints by M.

With respect to the notation used above, we have that v_i is recalled as the centroid of the i-th cluster (cluster center/cluster representative), and that m is a parameter ($m \geq 1$) that expresses the desired level of fuzziness. This is, m determines the degree of fuzziness in the membership functions. With values of m near to 1, solutions tend to be crisp (with the particular case that $m = 1$ corresponds to the crisp c-means). Instead, larger values of m yield to clusters with increasing fuzziness in their boundaries.

Local optimal solutions of the fuzzy c-means problem are obtained using an iterative process, that interleaves two steps. The first one that estimates the optimal membership functions of elements to clusters (considering the centroids as fixed) and another that estimates the centroids for each cluster (having the memberships as constant). This process is defined as follows:

Step 1: Generate an initial U and V
Step 2: Solve $min_{U \in M} J(U, V)$ computing:

$$u_{ik} = \Big(\sum_{j=1}^{c} \Big(\frac{||x_k - v_i||^2}{||x_k - v_j||^2} \Big)^{\frac{1}{m-1}} \Big)^{-1}$$

Step 3: Solve $min_V J(U, V)$ computing:

$$v_i = \frac{\sum_{k=1}^{n} (u_{ik})^m x_k}{\sum_{k=1}^{n} (u_{ik})^m}$$

Step 4: If the solution does not converge, go to step 2; otherwise, stop

As the method leads to a local optimal, different initial values can lead to different solutions.

The so-called entropy-based fuzzy c-means (EFCM) is an alternative fuzzy clustering method (proposed in [10], see also [11]). The main difference between fuzzy c-means and entropy-based fuzzy c-means is the way in which fuzziness is introduced. In this case, a parameter λ ($\lambda \geq 0$) is used to force a fuzzy solution. Formally speaking, the method is defined in terms of the optimization of the following objective function:

$$J_{EFCM}(U,V) = \sum_{k=1}^{n}\sum_{i=1}^{c}\{u_{ik}||x_k - v_i||^2 + \lambda^{-1}u_{ik}logu_{ik}\} \tag{2}$$

Again, the objective function is subject to the constraints $u_{ik} \in [0,1]$ and $\sum_{i=1}^{c} u_{ik} = 1$ for all k.

The parameter λ plays a role similar to m in fuzzy c-means. Here, the smaller the λ, the fuzzier the solutions. Instead, when λ tends to infinity, the second term becomes negligible and the algorithm yields to a crisp solution.

The way to solve EFCM is an iterative process, as for the FCM, but with different expressions for computing the memberships u_{ik} and the centroids v_i. More concretely, the following expressions are considered:

$$u_{ik} = \frac{e^{-\lambda||x_k - v_i||^2}}{\sum_{j=1}^{c} e^{-\lambda||x_k - v_j||^2}} \tag{3}$$

$$v_i = \frac{\sum_{k=1}^{n} u_{ik}x_k}{\sum_{k=1}^{n} u_{ik}} \tag{4}$$

FCM and EFCM lead to different solutions. A relevant difference is that the centroids have a membership equal to one in the FCM while in the EFCM it might have a lower membership. It can be easily observed that given a unique set of centers, the memberships and the shape of the clusters would be different in both cases due to the way memberships u_{ik} are computed.

A variation of these clustering methods was introduced [12] so that the size of each cluster is variable. The variation consists on a variable for each cluster roughly corresponding to its size. The rationale of such introduction was to reduce misclassification when there are clusters of different size. In standard FCM, two adjacent clusters have equal membership function (equal to 0.5) in the mid-point between the two centroids.

Formally speaking, the size of the i-th cluster is represented with the parameter α_i (the largest is α_i, the largest is the proportion of elements that belong to the i-th cluster). A similar approach was given by Ichihashi, Honda and Tani in [7].

When such parameters for variable size are considered, the expressions to minimize for FCM and EFCM are as follows:

$$J_{FCM}(\alpha,U,V) = \sum_{i=1}^{c}\alpha_i\sum_{k=1}^{n}(\alpha_i^{-1}u_{ik})^m||x_k - v_i||^2$$

$$J_{EFCM}(\alpha, U, V) = \sum_{k=1}^{n}\sum_{i=1}^{c}\{u_{ik}||x_k - v_i||^2 + \lambda^{-1}u_{ik}log(\alpha_i^{-1}u_{ik})\}$$

Both objective functions are minimized considering the constraints given above for the membership values u_{ik}, and adding additional constraints for α_i. The new constraints are the following ones:

- $\sum_{i=1}^{c}\alpha_i = 1$
- $\alpha_i \geq 0$ for all $i = 1, \ldots, c$

These fuzzy clustering algorithms are also solved by an iterative process, but now including an additional step for estimating the parameters α_i. In the case of the FCM, the values of α are estimated by (this corresponds to [*Step 3.1*] in the algorithm for FCM):

$$\alpha_i = \Big[\sum_{j=1}^{c}\Big(\frac{\sum_{k=1}^{n}(u_{jk})^m||x_k - v_j||^2}{\sum_{k=1}^{n}(u_{ik})^m||x_k - v_i||^2}\Big)^m\Big]^{-1}$$

In such algorithm, the expression for u_{ik} in Step 2 should be replaced by (the expression for v_i in Step 3 is valid):

$$u_{ik} = \Big(\sum_{j=1}^{c}\Big(\frac{\alpha_j}{\alpha_i}\Big)\Big(\frac{||x_k - v_i||^2}{||x_k - v_j||^2}\Big)^{\frac{1}{m-1}}\Big)^{-1} \tag{5}$$

In the case of variable-size EFCM, the expression of v_i is still valid but the following expressions are required for u_{ik} and α_i:

$$u_{ik} = \frac{\alpha_i e^{-\lambda||x_k - v_i||^2}}{\sum_{j=1}^{c}\alpha_j e^{-\lambda||x_k - v_j||^2}} \tag{6}$$

$$\alpha_i = \frac{\sum_{k=1}^{n}u_{ik}}{n}$$

2.2 Classification

In this section we review the use of hard c-means for building classifiers. For this purpose, we consider a set of examples x_k in a p dimensional space, and for each example its class $\kappa(x_k)$.

Then, given a set of examples x_k, the classification model is built considering the following two steps:

1. For each $\kappa_i \in \kappa$, define X_κ as those x_k such that its class is κ (*i.e.*, $\kappa(x_k) = \kappa$):

$$X_\kappa := \{x|\kappa(x_k) = \kappa\}$$

2. Apply hard c-means to each X_κ, and construct for each X_κ c different clusters. Therefore, we obtain $c \cdot |\kappa|$ centroids. We will use v_κ^r for $r = 1, \ldots, c$ to denote the c centroids obtained for class κ.

Then, using the centroids (v_κ^r for $r = 1, \ldots, c$) obtained in the previous step, the classification of new examples ex into the classes κ in κ is done applying the following algorithm:

$minDist = \infty$
For all r do
    For all κ do
        $minDist = \min(minDist, d(ex, v_\kappa^r))$ (distance between ex and the centroid v_κ^r)
    end loop
end loop
Assign ex to the class κ if $minDist = v_\kappa^r$ for some r.

This method corresponds to building a voronoi map in two steps. First, the examples of each class are partitioned and, second, their centroids are put together to define the map.

3 Classification Model Based on Fuzzy Clustering

We have extended the model described in Section 2.2 to incorporate fuzziness. Now, as in the case of crisp partitions, we start grouping the objects according to its class. This is, computing $X_\kappa := \{x | \kappa(x_k) = \kappa\}$. Then, the fuzzy clustering algorithm is applied to each class X_κ. This leads to a set of centroids for each class. We will use v_κ^r for $r = 1, \ldots, c$ to denote the c centroids obtained for class κ. As in the case of the c-means, we obtain $c \cdot |\kappa|$ centroids.

The computation of the class of new objects in the p dimensional space differs from the case of the c-means. In that case, the nearest centroid was considered as the most rellevant issue. Now, we will consider the membership value of the object into each class. Then, for testing we will select the class with the largest membership.

Nevertheless, the consideration of membership functions is not straightforward. As fuzzy c-means (and its variations) results into a fuzzy partition for each class κ, we have that the membership of a new object into the classes can be computed in, at least, two different ways. The two alternatives are described below. We use ex, as in the previous section, to denote the new example to be classified.

1. Consider the c fuzzy partitions as separated partitions. Then, compute for each class κ, and for each cluster r ($r = 1, \ldots, c$), the membership of ex to v_κ^r. Compare memberships and assign to ex the cluster and class with the largest membership.
2. Consider the c fuzzy partitions as a single partition (combined partition). This is, put all centroids together and compute a new fuzzy partition that encompass all the existing clusters. Then, determine the membership of ex to all the clusters and assign to ex the cluster and the class with the largest membership.

The computation of the new fuzzy partition in the second alternative is simple in the case of FCM. In this case, we can compute the fuzzy partition inferred from any set of centroids using the expression u_{ik} in Step 2 (Section 2). Thus, defining the set $\mathcal{V} := \cup_{r=1,\ldots,c} \cup_{\kappa} v_{\kappa}^{r}$ and then using:

$$u(ex, v_i) = \Big(\sum_{v \in \mathcal{V}} \Big(\frac{||ex - v_i||^2}{|ex - v||^2} \Big)^{\frac{1}{m-1}} \Big)^{-1}$$

for all $v_i \in \mathcal{V}$ we can determine the membership of ex to the new partition.

Similarly, when the clustering algorithm used is the EFCM, a similar process can be applied. In this case, Expression 3 should be used for computing the new memberships. So, the membership of ex to clusters v_i in $\mathcal{V}$ is defined as:

$$u(ex, v_i) = u_{ik} = \frac{e^{-\lambda||ex - v_i||^2}}{\sum_{v \in \mathcal{V}}^{c} e^{-\lambda||ex - v||^2}}$$

Instead, when variable size fuzzy c-means is considered, it is not enough to define the union of the centroids and apply the corresponding expressions for computing the membership values. As can be observed in Expressions 5 and 6, these expressions depend on the values of α (the size of the clusters), and when merging the two sets of centroids, the values of α are no longer valid. To solve this drawback we have defined a new vector of α' in terms of the previous values of α. These new values are computed as follows:

$$\alpha'_{\kappa,i} := \alpha_{\kappa,i} * |learningSetClass(\kappa)|/|learningSet|$$

where i is for all $i \in 1, \ldots, c$.

This definition of α' satisfies the constraints that $\alpha_i \geq 0$ and $\sum \alpha_i = 1$.

3.1 Analysis

Methods based on k-means assume that clusters are crisp and that in the resulting model the region under study is splitted (in a crisp way) following a Voronoi tessellation. This tessellation is based on the centroids of the clusters obtained by the k-means.

When fuzzy clustering algorithms are used, these assumptions are changed. First of all, we soften the crisp constraint on the boundaries of each region. Thus, objects can belong at the same time to different clusters. Nevertheless, when for a given point only the largest membership value is considered, the resulting tessellation is still the Voronoi one.

Variable-size fuzzy clustering permits clusters to have different size. Roughly speaking, the larger the number of objects associated to a centroid, the larger the region of the corresponding cluster. In this case, the tessellation changes its shape and not only the centroids come into consideration but also the dimension of the cluster (the parameter α using the notation given above).

Thus, using other fuzzy clustering methods than the k-means for building a classifier, the differences on the clustering model are exported to the classifier. This is for example the case of using a fuzzy clustering method that considers variable size.

4 Experiments

We have applied our approach to the classification of gene expressions in the budding yeast *Saccharomyces cerevisiae*. In particular, we used the data downloaded from [13] and described in [4]. Each gene is described in terms of numerical values, and most of them include a label with its name and function. The file contains information on 6221 genes. This data has been used in several studies as in [1].

In our case, we have used these data to compare the four approaches described in Section 3. This is, the classification based on the FCM and the EFCM, with and without variable size.

Some preprocessing was applied to the data as there are missing values. First, data was normalized to avoid scaling problems among variables. Normalization was achieved substracting the mean of each variable and dividing by the corresponding deviation. After normalization, missing values have been replaced by zero (this corresponds to replace the original data by its mean).

In this paper we report the results obtained for the case of the gene labels equals to "mitosis" and "protein degradation".

For testing, we have splitted the data into two sets: one for learning the model and the other for testing. Given a label, we have considered two classes: (i) those genes belonging to the class (positive examples) and (ii) those that do not belong to the class. Then, we selected at random 20% of the records of each class for learning and the rest was used for testing.

For each training/test pair, we have tested the four algorithms FCM and EFCM with and without variable size. Also, in each case we have compared the two alternatives of constructing the membership function (considering the partitions as separated entities or putting them together). For each of the algorithms, several values of m, λ and c were considered. In particular, we have considered the following values for m and λ: $m = \{1.05, 1.2, 1.4\}$, $\lambda = \{40, 20, 10\}$. With respect to c, we have considered two cases, one with $c = 3$ for both positive and negative examples and another with $c = 3$ for positive examples and $c = 8$ for negative examples. The consideration of a larger number of clusters for negative examples was due to the fact that the number of negative examples is much larger than those for positive examples.

Table 1. Rate of success considering separated partitions and combined partitions. In the upper part of the table results corresponds to the "mitosis" problem and the lower part corresponds to the "protein degradation" problem.

Clustering	parameter	c	Separated	Combined
FCM	$m = 1.05$	3	0.5163	0.8426
VFCM	$m = 1.05$	3	0.6355	0.9430
ENT	$\lambda = 40.0$	3	0.0176	0.8117
FCM	$m = 1.05$	3	0.5439	0.8099
VFCM	$m = 1.05$	3	0.5417	0.9639
ENT	$\lambda = 40.0$	3	0.0262	0.7966

Table 2. The rate of success for FCM and variable size fuzzy c-means (FCM) for the "mitosis" and the "protein degradation" problem, for several executions

c	FCM-mitosis	VFCM-mitosis	FCM-P.D.	CFCM-P.D.
3	0.823	0.99051	0.806	0.98392
8	0.867	0.99051	0.845	0.98392
3	0.823	0.99051	0.884	0.98408
8	0.882	0.99051	0.926	0.98408
3	0.851	0.99035	0.808	0.98408
8	0.828	0.99051	0.836	0.98408
3	0.841	0.99035	0.854	0.98424
8	0.914	0.99051	0.829	0.98424
3	0.900	0.99019	0.784	0.98392
8	0.920	0.99019	0.823	0.98408
3	0.832	0.99051	0.864	0.98408
8	0.856	0.99051	0.910	0.98408
3	0.878	0.99067	0.837	0.98392
8	0.889	0.99051	0.877	0.98408
3	0.842	0.99051	0.875	0.98408
8	0.848	0.99051	0.900	0.98408

Each combination of algorithm/parameters was executed 8 times, selecting each time the 20% of the records at random using a different seed.

The results show that considering a single partition lead to better results than considering two separated partitions. Table 1 shows the number of records classified correctly for the algorithms considered for some of the tests. It can be seen that the difference between separated and combined partitions is significant, being the combined partitions better than the separated ones.

Additionally, we can see that when the partitions are combined, the FCM with variable size is the method that obtains better results. These results are valid for both the experiments on "mitosis" and "protein degradation". The results of the 8 executions (with $m = 1.2$ and c either 3 or 8) are given in Table 2.

5 Conclusions and Future Work

In this paper we have studied the use of fuzzy c-means and some of its variations for building classifiers. We have proposed a way to deal with fuzziness and fuzzy partitions when several classes are present. We have analysed the characteristics of our method with respect to the one based on k-means. We have underlined the differences between them. The approach has been applied to data from bioinformatics. In particular, we have applied our method to classify gene expressions from the yeast Saccharomyces cerevisiae.

Although the results obtained by our approach are significant for the two problems studied, the use of fuzzy clustering is not necessarily better than crisp clustering for all sets of examples. The appropriateness of the method depends on

the data and its structure. Nevertheless, we consider that the better performance of variable-size fuzzy c-means with respect to standard fuzzy c-means is rellevant.

As future work we consider the implementation of new experiments, and the study of new methods to combine the fuzzy sets resulting from several fuzzy clustering algorithms (or several executions of the same clustering method with different objects).

Acknowledgements

Partial support by Generalitat de Catalunya (AGAUR, 2004XT 00004) and by the Spanish MEC (project "PROPRIETAS", SEG2004-04352-C04-02 and grant "Salvador de Madariaga" PR2005-0337) is acknowledged.

References

1. Barreto Bezerra, G., Nunes de Castro, L., (2003), Bioinformatics Data Analysis Using an Artificial Immune Network, ICARIS 2003, Lecture notes in Computer Science 2787 22-33.
2. Bezdek, J. C., (1981), Pattern Recognition with Fuzzy Objective Function Algorithms, Plenum Press, New York, 1981.
3. Duda, R. O., Hart, P. E., Stork, D. G., (2000), Pattern Classification, Wiley.
4. Eisen, M. B., Spellman, P. T., Brown, P. O., Botstein, D., (1998), Cluster analysis and display of genome-wide expression patterns, Proc. Natl. Acad. Sci. USA, 95 14863-14868.
5. Hastie, T., Tibshirani, R., Friedman, J., (2001), The Elements of Statistical Learning, Berlin: Springer.
6. Höppner, F., Klawonn, F., Kruse, R., Runkler, T., (1999), Fuzzy cluster analysis, Wiley.
7. Ichihashi, H., Honda, K., Tani, N., (2000), Gaussian mixture PDF approximation and fuzzy c-means clustering with entropy regularization, Proc. of the 4th Asian Fuzzy System Symposium, May 31-June 3, Tsukuba, Japan, 217–221.
8. Mitchell, T., (1997), Machine Learning, McGraw Hill.
9. Miyamoto, S., Umayahara, K., (2000), Methods in Hard and Fuzzy Clustering, pp 85–129 in Z.-Q. Liu, S. Miyamoto (Eds.), Soft Computing and Human-Centered Machines, Springer-Tokyo.
10. Miyamoto, S., Mukaidono, M., (1997), Fuzzy c - means as a regularization and maximum entropy approach, Proc. of the 7th International Fuzzy Systems Association World Congress (IFSA'97), June 25-30, Prague, Chech, Vol.II 86–92.
11. Miyamoto, S., (1999), Introduction to fuzzy clustering, Ed. Morikita, Japan. (in Japanese)
12. Miyamoto, S., Umayahara, K., (2000), Fuzzy c-means with variables for cluster sizes, 16th Fuzzy System Symposium, Akita, Sept.6-8, 537–538 (in Japanese).
13. http://rana.lbl.gov/EisenData.htm