2014 International Joint Conference on Neural Networks (IJCNN)
July 6-11, 2014, Beijing, China

Automatic Cluster Labeling through Artificial Neural Networks

Lucas A. Lopes, Vinicius P. Machado and Ricardo de A. L. Rabêlo

Abstract— **The clustering problem has been considered as one of the most important problems among those existing in the research area of unsupervised learning (a Machine Learning subarea). Although the development and improvement of algorithms that deal with this problem has been focused by many researchers, the main goal remains undefined: the understanding of generated clusters. As important as identifying clusters is to understand its meaning. A good cluster definition means a relevant understanding and can help the specialist to study or interpret data. Facing the problem of comprehend clusters – in other words, create labels – this paper presents a methodology to automatic labeling clusters based on techniques involving supervised and unsupervised learning plus a discretization model. Considering the problem from its inception, the problem of understanding clusters is dealt similar to a real problem, being initialized from clustering data. For this, an unsupervised learning technique is applied and then a supervised learning algorithm will detect which are the relevant attributes in order to define a specific cluster. Additionally, some strategies are used to create a methodology that presents a label (based on attributes and their values) for each cluster provided. Finally, this methodology is applied in four distinct databases presenting good results with an average above 88.79% of elements correctly labeled.**

I. Introduction

THE clustering problem can be considered as one of the most important among those involving unsupervised machine learning algorithms. The goal is to break a data collection in smaller structures (groups or clusters) which contain, somehow, similar elements under a particular perspective. In addition, the elements that belong to the same cluster must possess enough dissimilarity to be distinguished from other groups. This subject is well studied in the literature and discusses several problems and techniques as Genetics Algorithms [1], heterogeneous data sets [2] and many others [3][4] showing several strategies for the problem of clustering.

Nevertheless, another not very widespread aspect of this subject deals with the task of cluster labeling. The problem consists on naming the clusters according to their main features, which means to present a clear group identification. A good definition of each cluster could make it more understandable for a specialist while studying or interpreting data.

In the literature this issue is handled differently. However, a very similar problem is the need to label new elements based on groups previously defined, having already been presented in some works as [5], [6], [7]. The main difference between these works and the one proposed here is that in the first one a new element can be classified on a predefined cluster according to the technique used and in the second one an effective definition (or in other words, a label) of each cluster will be given to help the specialist. The generated labels also can be used to classify new elements although this is not the main point here.

Lucas Araújo Lopes, Vinicius P. Machado and Ricardo de A. L. Rabêlo are with the Computing Department, Federal University of Piauí, Teresina, Piauí (email: {lucaslopes, vinicius}@ufpi.edu.br, ricardor_usp@ieee.org).

The unsupervised learning techniques are applied to a collection of data (database) and, as a result, several groupings with similar elements are created. However, the conventional methods used for clustering often do not give the clusters a proper meaning. The purpose of this work is to detect which are the key features (relevant attributes) in each group as well as their possible values, in order to clarify, steer and help the analysis and labeling of groups held by experts. For this, a supervised and unsupervised learning techniques combined with a discretization method are used.

This paper is organized as follows. Section II presents two techniques related to supervised and unsupervised learning. In Section III, we present a mechanism to label clusters. Section IV reports some results and analysis from our proposed model applied on four data sets. The paper concludes with Section V.

II. Theoretical Framework

In our proposal, it is necessary to use an unsupervised learning algorithm to deal with the clustering problem. It does not make sense to use a supervised learning algorithm since there is the need to discover to which cluster each element belongs.

The technique chosen among several algorithms was *K-means* [8], due to its compact structure and efficiency. However, any other clustering algorithm can be used. In addition, in each database used here it is known, *a priori*, the number K of clusters to be generated – a parameter required by *K-means*. These numbers can be found in the original works of each database. If this parameter is not supplied, some tests ranging this value should be done. Alternatively, any other clustering technique can be used for this task: *Cobweb* [9], *Self Organizing Maps* (SOM) [10], *Fuzzy Clustering* [11], in addition to hybrid techniques [12], [13], among others.

The *K-means* is one of unsupervised learning algorithms which deals with the task of clustering. *A priori*, a number K of clusters must be reported indicating how many centroids can be generated. A centroid is a point that represents the center of a cluster. The main idea is to determine K centroids, one for each cluster. The value of K is a very important parameter here: if it is too high similar elements will not be grouped together whereas if it is too low different elements will belong to the same cluster.

Another step of this proposal will require the use of a supervised learning algorithm to detect a possible relationship

978-1-4799-1484-5/14/$31.00 ©2014 IEEE

among the features present on the problem. For this task the use of Artificial Neural Networks (ANNs) was chosen mainly for its capability of learning, ability of generalization, fault-tolerance and data organization – grouping patterns which have the same particularities – beyond being much used. ANNs are known for dealing with non-linear and/or dynamic problems. They are computational models inspired in the nervous system of living being and are known for their ability to detect patterns and their strong fault-tolerance [14].

The most basic neural network is the Perceptron [15]. The Perceptron is composed of an artificial neuron that receives incoming signals. These signals are multiplied by numerical weights – which represents its knowledge – and processed by a function offering a way out.

There are several types of ANNs but we focus on Multilayer Perceptron network (MLP) [16]. The MLP is a feedforward network where there are at least two layers (a hidden one and an output one). Typically, the output values of a neurons layer serve as input only for the neurons of the next layer. In this work, a MLP network will be used to find a possible relationship between attributes so that all the others (input) will try to predict one (output).

III. Proposed Model

Our proposal, facing the labeling problem presented in Section I, is to define a model to labeling clusters. An algorithm with unsupervised learning is initially applied with the aim of forming various groups among the elements concerned. For each formed group a second algorithm will be assigned but this time with a supervised learning process that will allow the identification of relevant features.

The schema of Fig. 1 demonstrates the steps.

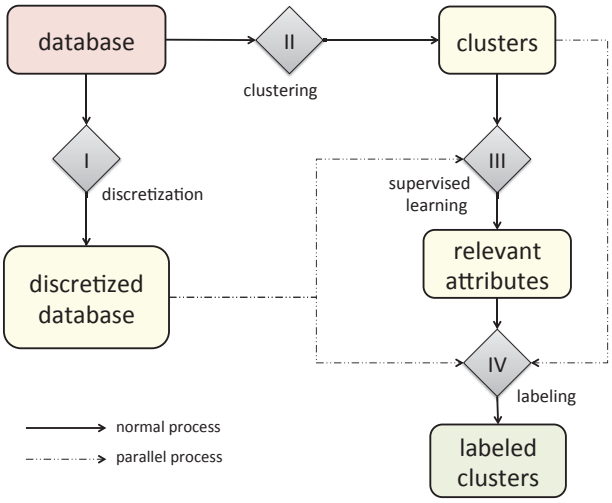

Fig. 1. Steps from proposed model.

Initially, there is a database as entry. This database may have different types of data (discrete/continuous) and sometimes a discretization model (I) may be applied. In summary, it is necessary apply this process when there is the need to estimate a range of values and not a specific one. In order to obtain a better performance and, mainly, make it possible to infer a range of values towards some attributes we conducted a discretization process where the different possible values boil down to intervals or ranges.

The second step (II) is performed by a unsupervised algorithm which performs the task of clustering. In this work, the discretization model is not related on clusters generation – only on steps III and IV. Once the clusters are generated a supervised algorithm is applied (III) in each group in order to detect which are the relevant attributes to the definition of each cluster. Finally, the labeling (IV) is performed in each cluster. Each step is detailed hereafter.

A. Discretization (I)

The step I consists in data discretization: for the attributes that can take on different values among a specific domain, new discrete values will be established. Thus, the supervised learning algorithm will be able to more easily identify a possible relationship between attributes showing better results in their classification in exchange of information loss. Also, the discretization process makes possible to infer a range of values in the final step, converting back the value discretized. It is important to remember that this step is only necessary for some attributes. When it is not necessary, then a skip to step II might be done.

In the literature ([17], [18]) there are several discretization methods. The two methods most commonly used are Equal Width Discretization (EWD) and Equal Frequency Discretization (EFD).

The EWD discretization model uses some means to discretize data. For example, to discretize the data in four ranges of values there will be necessary three means. The first mean (m) is the simple arithmetic mean between the lowest and the highest value of the attribute concerned. The second ($leftM$) and the third ($rightM$), both simple arithmetics, can be calculated using the first mean (m) with the lowest or highest value, respectively. Thus, for each attribute there will be 4 ranges of values (Fig. 2).

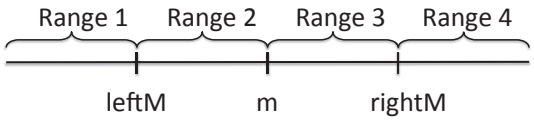

Fig. 2. 4 ranges in a EWD model.

- Range 1: value less than or equal to the second mean ($leftM$);
- Range 2: value greater than the second mean ($leftM$) and less than or equal to the first mean (m);
- Range 3: value greater than the first mean (m) and less than or equal to the third mean ($rightM$);
- Range 4: value greater than the third mean ($rightM$).

The other model is the EFD which deals with ranges of values that contain the same quantity of distinct values among the provided elements. Given a number E of distinct

elements and a number R of ranges we can define each range containing $D = E/R$ (rounded on down) distinct elements. Observe that E must be equal or greater than R and both values must be greater than 0.

Before defining the minimum and the maximum value of each range, it is still necessary to sort the values of the distinct elements. After that, the first range has its minimum as the lowest value sorted and its maximum as the value indicated by the Dth value sorted creating an interval that can be represented as $[min, Dth]$. A next range, rising from $r = 2$ to R, will start with values greater than the maximum of the previous range, $((r-1) * D)$-th, and go on until the value presented by the $(r * D)$-th sorted value. The interval created can be represented as $]((r-1) * D)$-th, $(r * D)$-th$]$ and all this process can be presented as follow, in Fig. 3.

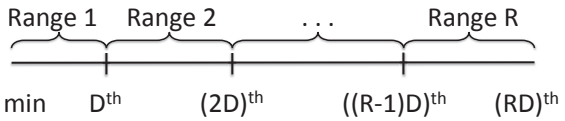

Fig. 3. Ranges in a EFD model.

The use of a discretization model allows the unsupervised algorithm to work with ranges of values facilitating the detection of relevant attributes and also making possible infer a set of values for the generated label.

How the discretization model – and its amount of ranges – will be applied is something to be discussed according to the circumstances of each problem. The discretized values are stored and will be used later during the steps III (training) as input to the supervised algorithm and IV (labeling) as the bounds of the intervals – ranges of values.

B. Clustering (II)

After discretization, the generation of clusters occurs (step II). The problem of grouping is quite studied and there are some strategies already mentioned in Section II, where *K-means* was the algorithm applied here. In this step we have a database as input and its elements grouped in K clusters as output.

C. Supervised Learning (III)

A supervised algorithm will be applied in each generated cluster. In this step, the idea is to detect which attributes are relevant – detecting a relationship among the attributes – to the group. For this, an ANN with supervised learning is applied for each attribute where it is considered as an attribute class (output) and the others as network inputs in order to find out which attributes may classify the group correctly. Fig. 4 illustrates this step, taking as an example a cluster in which its elements have three attributes.

For each attribute of the elements from a given cluster will be created an ANN. These ANNs will present as output the estimated value for the attribute concerned and will have as input the other attributes. Each ANN of a same cluster works

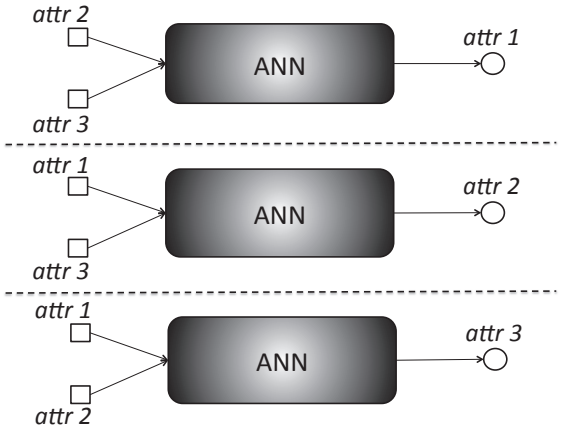

Fig. 4. ANNs as supervised algorithm.

with the same elements varying only on the way that their attribute values are used in the network – input or output.

Considering any cluster, the database will be divided into two sets (randomly for each network): training and testing. These sets will be used in a process known as cross-validation [19] – *holdout* method – and used by the network to its own learning. The testing process will be used to measure the efficiency of the network in relation to its learning obtained during the training process. After learning, during the testing phase, if the output value of the network is equal to the value corresponding to the attribute range for its value concerned then there is a hit. Otherwise, there is the occurrence of an error.

Therefore, each ANN is created to represent and evaluate the importance of its output (which is an attribute). In a wider way, each cluster will have a hit rate for each ANN – a hit rate for each evaluated attribute. Thus, we can know which attribute is relevant in relation to the others for a given cluster: is the one that got higher hit rate in the testing phase. This attribute is relevant because it can summarize a combination of other attributes. For greater confidence regarding the attribute there is an average of M performances in this step. In each performance, an ANN is created for each attribute and the final value used for is the mean of all M performances.

D. Labeling (IV)

The last step (IV) is to appoint the clusters according to its attributes. After the training stage each cluster will have the attributes average hit in M performances. The ANN(s) having the highest hit rate average indicates the most relevant attribute(s).

Another parameter, variation V, will select the others attributes that have a hit rate with variation of at most V (given in percentage) in relation to the main attribute. Thus, we will have a set of attributes that can be seen as relevant to the definition of such cluster.

After setting the group of relevant attributes we confirm which of the values (defined in the discretization step) dominates the group. In other words, we detect what each

attribute value range features more frequently in any cluster in that attribute taken as relevant. Therefore, we have the precision of each attribute importance (hit rate) as well as their likely values (ranges). Those two pieces of information are very important to labeling and they will be used to name the clusters. The Algorithm 1 demonstrates in a natural language all the proposed model in this section.

Algorithm 1 Labeling

Require: Database
Ensure: *Labeled* clusters

```
Load database;
Discretize each continuous attribute (if necessary);
Perform clustering algorithm (unsupervised);
for each cluster do
    for each performance m = 1 to M do
        Define training and test sets;
        for each attribute do
            Perform training (supervised learning);
            Calculate the hit rate;
        end for
    end for
    Calculate the average of hit rates;
    Choose the relevant attributes (bests average of hit
    rates) included in a variance V;
    Count the majority values presented;
    Convert the values discretized in ranges (if necessary);
end for
Show labels;
```

At the end of the process the label of each cluster will be the set of relevant attributes selected in a variation V, with their respective values or range of values.

IV. Results

For the implementation of the proposed model we used the tool MATLAB[1], which provides the use of some supervised and unsupervised algorithms presented in Section II among others[2].

The *K-means* was used with the command *kmeans (X, k)*, where X is a matrix containing all elements (database) and k is the number of clusters to be generated. All parameters used are the default[3] by MATLAB. In the databases used here (*Glass* [20], *Scientia.Net* [21], *Iris* [22] and *Seeds* [23]) we know *a priori* the amount k of clusters that must be created, as suggested on these same works.

To perform a MLP network we used the command *feedforwardnet ()*. In this algorithm we also used the default settings, 10 neurons and 1 hidden layer, for the neural network[4]. Also, preliminary tests were done by changing the network architecture (amount of neurons and layers) without major differences in the results. In relation to learning method, 60% of the data was used for training and 40% for testing.

The parameters and the topology of the algorithms used (*K-means* and MLP), plus a discretization model and its parameters, a mean of the amount of ANNs by attribute (M) and a variation in relation to the higher hit rate by cluster (V), all presented in Section III, result in a large number of possible combinations, although the results presented here represent only a small fraction of them. The values[5] used were $M = 10$, $V = 15$ and the discretization model was the EWD and the EFD, varying in 3 to 6 ranges of values. The parameters used on MLP network (such as topology and architecture) and *K-means* (distance and centroids position) are the default by the MATLAB tool.

Then, the application of the proposed model will be shown in four databases.

A. Glasses Identification

Database regarding the identification of glasses (Identification Data Set Glass) can be found in the data store *UCI Machine Learning* [24]. The context of its application is the forensic area, where the analysis of the glass components can help to solve crimes [20].

Database has 214 elements, each containing nine continuous attributes[6]: the refractive index (*IR*) and the composition of its chemical elements given in percentage (*Na*, *Mg*, *Al*, *Si*, *K*, *Ca*, *Ba* and *Fe*), divided into 7 types of different groups that contain samples of glasses:

1) 70 elements of construction windows (processed);
2) 76 elements of construction windows (non-processed);
3) 17 elements of vehicles windows (processed);
4) 0 elements of vehicles windows (non-processed)[7];
5) 13 elements of containers;
6) 9 elements of kitchen utensils;
7) 29 elements of headlamps.

The results obtained are shown in Table I. It is observed that the labeling task is done according to the clusters generated by *K-means* and that, as shown in Table I, they differ from the form suggested of the work presented in [20]. Therefore, the labels presented here are specific and may differ at each performance according to the groups performed.

The relevance column (Rel.) represents the average hit rate of learning algorithm for the attribute concerned estimated by the ANN. In other words, it represents the relevance of such attribute to its cluster, showing only those ones held in a variation V.

As seen in Table I, for each cluster a set of attributes was suggested as well as their respective value ranges. At this point, it is necessary an analysis to verify if the elements of a given cluster obey the labeling suggested or, putting

[1]http://www.mathworks.com/products/matlab/
[2]Version used: R2012a (7.14.0.739), 64 bits (maci64).
[3]http://www.mathworks.com/help/stats/kmeans.html
[4]http://www.mathworks.com/help/nnet/ref/feedforwardnet.html
[5]Values chosen by preliminary tests.
[6]The attribute class (corresponding to a tenth attribute that identifies the type glass) has been removed from the base for the accomplishment of this work.
[7]No element of this type is present in database.

TABLE I
LABELING ANALYSIS FOR GLASS DATABASE.

Cluster	# Elem.	Result (Labels)			Analysis	
		Attr.	Rel. (%)	Range	# Errors	Hit (%)
1	74	Ba	100	$0 \sim 0.7875$	0	100
		K	100	$0 \sim 1.5525$	0	100
		Si	93.33	$72.61 \sim 74.01$	2	97.29
		Na	90.33	$12.3925 \sim 14.055$	3	95.94
2	5	Fe	100	$0 \sim 0.1275$	0	100
		Ca	100	$5.43 \sim 8.12$	0	100
3	19	K	100	$0 \sim 1.5525$	0	100
		Ba	90	$0 \sim 0.7875$	1	94.73
4	32	K	100	$0 \sim 1.5525$	0	100
		Ba	93.07	$0 \sim 0.7875$	1	96.87
		Ca	89.23	$8.12 \sim 10.81$	1	96.87
5	56	Ba	100	$0 \sim 0.7875$	0	100
		K	100	$0 \sim 1.5525$	0	100
		Na	93.47	$12.3925 \sim 14.055$	2	96.42
		Al	88.69	$1.0925 \sim 1.895$	4	92.85
		Mg	85.65	$3.3675 \sim 4.49$	6	89.28
6	28	Fe	100	$0 \sim 0.1275$	0	100
		K	93.33	$0 \sim 1.5525$	1	96.42

differently, if the values of its attributes belong to the range shown. The Table I also shows this analysis.

Assuming that only the main attributes define the labeling – the attributes that were 100% relevant on Table I – we can observe that there is no error. On the other hand, if only these attributes are considered, there is a possibility of ambiguity occurrence between labels (same relevant attributes with the same value ranges) as occurs between the pairs of clusters 1-5, 3-4 and 6-2.

It is necessary, then, to observe whether the other suggested attributes within a variation V are enough to distinguish all the labels. Thus, as seen previously in two examples, the cost to avoid the ambiguity is the reliance on less relevant attributes. This parameter should be adjusted if the amount of relevant attributes is not enough to distinguish all the clusters.

For example, to differentiate the clusters 3 and 4 we could follow all the suggested labeling. Thus, the label of the cluster 3 is represented by *K* ($0 \sim 1.5525$) and *Ba* ($0 \sim 0.7875$); and cluster 4 by *K* ($0 \sim 1.5525$), *Ba* ($0 \sim 0.7875$) and *Ca* ($8.12 \sim 10.81$).

An alternative, that could solve the problem of ambiguity, would be to use more precise value ranges or a different discretization model.

Even with the alternative used the hit rate remains high. In the example showed the use of the attribute *Ca* in the cluster 4 showed that only 1 element (of 32) does not match the label, resulting in a hit rate of 96.87%.

Following this reasoning and analyzing the others clusters we have that the lowest hit rate is present on cluster 5 (attribute *Mg*) and corresponds to 89.28%. Generally, the result was quite satisfactory reaching an average of 95.54% hit of the elements as suggested labels.

B. Scientia.Net

The *Scientia.Net* [21] is a social network prototype aimed at scientists who wish to share research with other researchers. In addition, its machine learning algorithms automatically classify content and members.

Database of *Scientia.Net* has 2000 users of 20 (distinct) knowledge areas and its elements characterized by 7 discrete attributes[8] that define their academic area:

1) Graduation;
2) Masters;
3) Master's Sub Area;
4) Doctorate;
5) Doctorate's Sub Area;
6) Postdoctoral;
7) Postdoctoral Sub Area.

The different areas are represented by numbers. The area of *Geography*, for example, is represented by number 27 in the attributes *postdoctoral*, *doctorate* and *masters* and number 20 in the attribute *graduation*. The process of discretization was not applied here since all the attributes used here are are discrete/categoric.

The results obtained are shown in Table II[9]. Therefore, as in the previous case, the labeling is done on the basis of clusters generated by the unsupervised algorithm (*K-means*) and that, as shown in Table II, resembles – mostly – the results obtained in [21]. Even tough, the labels presented here are also specific and may differ at each performance according to the grouping performed.

As in the first database, the relevance column (Rel.) represents the average hit rate of learning algorithm for the attribute concerned estimated by the ANN. Again, it represents the relevance of such attribute to its cluster, showing only those ones held in a variation V.

After the results presented by the program the analysis was done. As in Table II, for each cluster a set of attributes was suggested as well as their corresponding values. The analysis consists of observing whether the elements of a given cluster obey the labeling suggested. The Table II shows this analysis too.

By having discrete values and not continuous value ranges the groups are best defined by the unsupervised algorithm. Comparing to the previous case (glass database), there were no labels alike, except for the clusters 3 and 20.

This is due to the grouping stage once the hit rate was 100% in both clusters. In some cases a grouping can separate similar elements in different groups as well as order different elements in the same group cluster. However, in general, the grouping performed was satisfactory because most clusters were well defined, close to the suggested by the authors.

The cluster 2 was the only one presenting one of the attributes of a sub area. These attributes contain more specific

[8]The attribute class (corresponding to an eighth attribute that identifies the user area) has been removed from the base for the accomplishment of this work.

[9]For limitations of space only the most interesting clusters results were chosen to show here.

TABLE II

LABELING ANALYSIS FOR SCIENTIA.NET DATABASE.

Cluster	# Elem.	Result (Labels)			Analysis	
		Attr.	Rel. (%)	Value	# Errors	Hit (%)
1	100	Postdr.	100	27	0	100
		Graduation	100	20	0	100
		MSc.	100	27	0	100
		Dr.	100	27	0	100
2	12	Postdr.	100	17	0	100
		MSc.	100	17	0	100
		Dr.	100	17	0	100
		Sub Dr.	100	13	0	100
⋮	⋮	⋮	⋮	⋮	⋮	⋮
4	160	Postdr.	100	18	60	62.5
		MSc.	100	18	60	62.5
		Dr.	100	18	60	62.5
⋮	⋮	⋮	⋮	⋮	⋮	⋮
7	103	MSc.	98.0952	14	3	97.0874
		Dr	97.619	14	3	97.0874
		Postdr.	97.8571	14	3	97.0874
⋮	⋮	⋮	⋮	⋮	⋮	⋮
15	172	Postdr.	100	9	80	53.4884
		Graduation	100	5	80	53.4884
		MSc.	100	9	80	53.4884
		Dr.	100	9	80	53.4884
⋮	⋮	⋮	⋮	⋮	⋮	⋮

information and therefore do not tend to be a high relevant attribute. However, it was suggested in the cluster 2 and, as we can observe, it was a correct classification since the group only contains 12 elements, being a very specific group.

According to the analysis of the results the worst hit rate were obtained in 53.4884% and 62.5%. Not coincidentally this rate refers to the largest clusters with 172 and 160 elements, respectively. Indeed, the label is not wrong: it represents the majority of the group. However, the low hit rate is due to the poor definition of groups that contain 72% and 60% elements more than it was presented in the original work [21].

Observing the other groups we have that the majority (55%) of the clusters presented a hit rate of 100%. Generally, the result was quite satisfactory reaching an average of 93.357% hit of the elements as suggested labels.

C. Iris Identification

Database regarding the identification of iris (Iris Data Set) also can be found in the data store *UCI Machine Learning* [24]. The data set contains 3 classes of 50 instances each, where each class refers to a type of an iris plant.

The database has 150 elements, each containing four continuous attributes[10]: the sepal length (*SL*), the sepal width (*SW*), the petal length (*PL*) and the petal width (*PW*), given in *cm*, divided into 3 types of different groups that contain samples of iris:

1) 50 elements from Iris Setosa;
2) 50 elements from Iris Versicolor;
3) 50 elements from Iris Virginica.

The results obtained are shown in Table III.

TABLE III

LABELING ANALYSIS FOR IRIS DATABASE.

Cluster	# Elem.	Result (Labels)			Analysis	
		Attr.	Rel. (%)	Range	# Errors	Hit (%)
1	62	PL	83.6	$3.7 \sim 5.1$	6	90.32
		SL	79.6	$5.3 \sim 6.4$	13	79.03
2	38	PW	84.37	$1.7 \sim 2.5$	3	92.10
		SW	82.5	$2.7 \sim 3.4$	6	84.21
		PL	82.5	$5.1 \sim 6.9$	2	94.73
3	50	PW	100	$0.1 \sim 1$	0	100
		PL	100	$1 \sim 1.37$	0	100

The labeling is done according to the clusters generated by *K-means* and that, as shown in Table III, they can differ from the form suggested of the work presented in [22] (50 elements in each cluster). Therefore, the labels presented here are specific and may differ at each performance according to the groups performed.

The relevance column (Rel.) represents the average hit rate of learning algorithm for the attribute concerned estimated by the ANN. In other words, it represents the relevance of such attribute to its cluster.

As seen in Table III, for each cluster a set of attributes was suggested as well as their respective value ranges. As in the previous example, it is necessary an analysis to verify if the elements of a given cluster obey the labeling suggested, or in another words, if the values of its attributes belong to the range shown. The Table III also shows this analysis.

Only the main attributes define the labeling: the attributes that have the best percentage of relevancy (as shown in Table III). However, it would be possible that there were similar groups. To avoid a possibility of ambiguity occurrence between labels on groups (same relevant attributes with the same value ranges), a variance V is used to select more attributes (more relevant as possible) to distinguish these clusters.

It is necessary, then, to note the other suggested attributes within a variation V that is enough to distinguish all the labels. That way, as seen in the previous example, the cost to avoid the ambiguity is the reliance on less relevant attributes. In this case, the groups have distinct values for the same attributes.

As we can observe in Table III, the amount of elements clustered was different from the original work. One group (cluster 3) was easily separated but the other two were mixed. This was expected once one class is linearly separable from the other two and the latter are not linearly separable from each other [22].

[10]The attribute class (corresponding to a fifth attribute that identifies the type iris) has been removed from the base for the accomplishment of this work.

As shown in Table III, the cluster 3 was rated 100% correctly using the attributes *PW* and *PL* to label it. The other two groups has a minor rate of 84.21% and 79.03%. All the attributes and their respective values are different showing no ambiguity between the clusters.

Finally, the labels suggested by the proposal are: *PL* ranging from 3.7 to 5.1 and *SL* ranging from 5.3 to 6.4 for Cluster 1; *PW* ranging from 1.7 to 2.5, *SW* ranging from 2.7 to 3.4 and *PL* ranging from 5.1 to 6.9 for Cluster 2; and *PW* ranging from 0.1 to 1 and *PL* ranging from 1 to 1.37 for Cluster 3.

Observing the groups as a whole, we have an average of 87.74% of the elements classified correctly by all attributes presented in a variance V which is a result quite satisfactory.

D. Seeds Identification

Database regarding the identification of seeds (Seeds Data Set) also can be found in the data store *UCI Machine Learning* [24]. The data set contains 3 classes of 70 instances where each class refers to a different type of wheat.

The database has 210 elements characterized by 7 geometric features[11]: area, perimeter, compactness, length of kernel (LK), width of kernel (WK), asymmetry coefficient (AC) and length of kernel groove (LKG), divided into 3 types of different groups that contain samples of kernels belonging to three different varieties of wheat:

1) 70 elements from Kama;
2) 70 elements from Rosa;
3) 70 elements from Canadian.

The results obtained are shown in Table IV.

TABLE IV
LABELING ANALYSIS FOR SEEDS DATABASE.

Cluster	# Elem.	Result (Labels)			Analysis	
		Attr.	Rel. (%)	Range	# Errors	Hit (%)
1	72	Perim.	81.6	13.62 ∼ 14.83	28	61.11
		Area	78.16	13.23 ∼ 15.88	20	72.22
2	77	Perim.	88.17	12.41 ∼ 13.62	15	80.51
		Area	91.39	10.59 ∼ 13.23	5	93.50
3	61	WK	100	3.33 ∼ 4.033	0	100
		Compac.	97.33	0.86 ∼ 0.91	4	93.44
		LKG	98.66	5.53 ∼ 6.55	1	98.36
		Area	100	15.88 ∼ 21.18	0	100
		Perim.	100	14.83 ∼ 17.25	0	100
		LK	100	5.78 ∼ 6.67	0	100

Again, it is observed that the labeling task is done according to the clusters generated by *K-means* and therefore, as shown in Table IV, they will be specific to them and not to the form suggested of the work presented in [23] (70 elements in each cluster). Thus, the labels presented here are specific and may differ at each performance according to the groups generated.

[11]The attribute class (corresponding to a eighth attribute that identifies the type wheat) has been removed from the base for the accomplishment of this work.

As in the previous cases, the relevance column (Rel.) represents the average hit rate of learning algorithm for the attribute concerned estimated by the ANN: it represents the relevance of such attribute to its cluster.

As seen in Table IV, for each cluster a set of attributes was suggested as well as their respective value ranges. Also, in order to verify if the elements of a given cluster obey the labeling suggested, or in another words, if the values of its attributes belong to the range shown, an analysis process was done. The Table IV shows this analysis too.

Only the main attributes define the labeling: the attributes that have the best percentage of relevancy (as shown in Table IV). However, it would be possible that there were similar groups. To avoid a possibility of ambiguity occurrence between labels on groups (same relevant attributes with the same value ranges), a variance V is used to select more attributes (more relevant as possible) to distinguish these clusters.

As shown in Table IV, the Cluster 3 has the highest rate of correct label with 93.44% of its elements labeled correctly. The Cluster 2 also presented a good percentage of correctly labeled elements: 80.51%. A low hit rate was presented for Cluster 1 with 61.11% only. Also, the clusters have not ambiguity presenting their values different from each other even when the labels have the same attributes. We can observe (Table IV) that the clusters are well defined since the values of their main attributes are different.

Finally, the labels suggested by the proposal are: *perimeter* (13.62 $\sim$ 14.83) and *area* (13.23 $\sim$ 15.88) for Cluster 1; *perimeter* (12.41 $\sim$ 13.62) and *area* (10.59 $\sim$ 13.23) for Cluster 2; and *WK* (3.33 $\sim$ 4.033), *compactness* (0.86 $\sim$ 0.91), *LKG* (5.53 $\sim$ 6.55), *area* (15.88 $\sim$21.18), *perimeter* (14.83 $\sim$ 17.25) and *LK* (5.78 $\sim$ 6.67).

Observing all the groups we have an average of 78.53% of the elements classified correctly by all labels suggested.

V. CONCLUSIONS

Facing the problem presented on Section I, an unsupervised algorithm was used for defining clusters and then an supervised learning algorithm was applied into each group to detect which attributes – and values – can define them. It is important to highlight that the discretization process has a significant purpose on this method, making possible to infer a range of values for the relevant attributes besides improving the performance of the ANNs.

It is also important to highlight that the labeling process is done in a cluster and that therefore it depends essentially on its elements. Thus, a poorly defined group will have an imprecise labeling. Therefore the unsupervised algorithm still has strong influence on the labeling result.

In face of the diversity of existing techniques for supervised and unsupervised techniques, discretization models and its parameters a significant improvement still can be reached.

The expensive cost of using ANNs still needs to be compared with other techniques – principal component analysis (PCA) or support vector machine (SVM), for example.

Although we have not done tests in big data the ANNs can easily work in parallel, making this approach scalable.

Finally, the results shown are quite satisfactory: most of the clusters evaluated in the databases shown in this article were labeled with high hit rates in an average above of 88.79% of elements labeled correctly.

ACKNOWLEDGMENT

The authors gratefully acknowledge the financial support of FAPEPI (Foundation of Research Support in Piaui), CAPES (National Council for the Improvement of Higher Education) and CNPq (National Council for Scientific and Technological Development).

REFERENCES

[1] Z. Zhang, H. Cheng, S. Zhang, W. Chen, and Q. Fang, "Clustering aggregation based on genetic algorithm for documents clustering," in *IEEE Congress on Evolutionary Computation, 2008. CEC 2008. (IEEE World Congress on Computational Intelligence).*, 2008, pp. 3156–3161.

[2] A. Abdullin and O. Nasraoui, "Clustering heterogeneous data sets," in *2012 Eighth Latin American on Web Congress (LA-WEB)*, 2012, pp. 1–8.

[3] J. Wang, Y. Jing, Y. Teng, and Q. Li, "A novel clustering algorithm for unsupervised relation extraction," in *2012 Seventh International Conference on Digital Information Management (ICDIM)*, 2012, pp. 16–21.

[4] F. de A.T. de Carvalho, G. Barbosa, and M. Ferreira, "Variable-wise kernel-based clustering algorithms for interval-valued data," in *2012 Brazilian Symposium on Neural Networks (SBRN)*, 2012, pp. 25–30.

[5] T. Eltoft and R. de Figueiredo, "A self-organizing neural network for cluster detection and labeling," in *IEEE World Congress on Computational Intelligence. The 1998 IEEE International Joint Conference on Neural Networks Proceedings, 1998.*, vol. 1, 1998, pp. 408–412 vol.1.

[6] H.-L. Chen, K.-T. Chuang, and M.-S. Chen, "Labeling unclustered categorical data into clusters based on the important attribute values," in *Fifth IEEE International Conference on Data Mining*, 2005.

[7] ——, "On data labeling for clustering categorical data," *IEEE Transactions on Knowledge and Data Engineering*, vol. 20, no. 11, pp. 1458–1472, 2008.

[8] T. Kanungo, D. Mount, N. Netanyahu, C. Piatko, R. Silverman, and A. Wu, "An efficient k-means clustering algorithm: analysis and implementation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 7, pp. 881–892, 2002.

[9] D. Fisher, "Improving inference through conceptual clustering," in *Proceedings of the sixth National conference on Artificial intelligence - Volume 2*, ser. AAAI'87. AAAI Press, 1987, pp. 461–465.

[10] M. Figueiredo, S. Botelho, P. Drews, and C. Haffele, "Self-organizing mapping of robotic environments based on neural networks," in *2012 Brazilian Symposium on Neural Networks (SBRN)*, 2012, pp. 136–141.

[11] B. A. Bushong, "Fuzzy clustering of baseball statistics," in *NAFIPS '07. Annual Meeting of the North American on Fuzzy Information Processing Society, 2007.*, 2007, pp. 66–68.

[12] D. Aziz, M. A. M. Ali, K. B. Gan, and I. Saiboon, "Initialization of adaptive neuro-fuzzy inference system using fuzzy clustering in predicting primary triage category," in *2012 4th International Conference on Intelligent and Advanced Systems (ICIAS)*, vol. 1, 2012, pp. 170–174.

[13] S. Ramathilaga, J.-Y. Leu, and Y.-M. Huang, "Adapted mean variable distance to fuzzy-cmeans for effective image clustering," in *2011 First International Conference on Robot, Vision and Signal Processing (RVSP)*, 2011, pp. 48–51.

[14] S. Haykin, *Neural Networks: A Comprehensive Foundation*, 2nd ed. Upper Saddle River, NJ, USA: Prentice Hall PTR, 1998.

[15] F. Rosenblatt, "Neurocomputing: foundations of research," J. A. Anderson and E. Rosenfeld, Eds. Cambridge, MA, USA: MIT Press, 1988, ch. The perception: a probabilistic model for information storage and organization in the brain, pp. 89–114. [Online]. Available: http://dl.acm.org/citation.cfm?id=65669.104386

[16] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Parallel distributed processing: explorations in the microstructure of cognition, vol. 1," D. E. Rumelhart, J. L. McClelland, and C. PDP Research Group, Eds. Cambridge, MA, USA: MIT Press, 1986, ch. Learning internal representations by error propagation, pp. 318–362. [Online]. Available: http://dl.acm.org/citation.cfm?id=104279.104293

[17] J. Cerquides and R. L. de Màntaras, "Proposal and empirical comparison of a parallelizable distance-based discretization method," in *In Proceedings of the 3rd International Conference on Knowledge Discovery and Data Mining*, 1997.

[18] H. Wang, "Cmp: A fast decision tree classifier using multivariate predictions," in *In Proceedings of the 16th International Conference on Data Engineering*, 2000, pp. 449–460.

[19] F. Leisch, L. Jain, and K. Hornik, "Cross-validation with active pattern selection for neural-network classifiers," *Neural Networks, IEEE Transactions on*, vol. 9, no. 1, pp. 35–41, 1998.

[20] I. W. Evett and E. J. Spiehler, "Knowledge based systems," P. H. Duffin, Ed. New York, NY, USA: Halsted Press, 1988, ch. Rule induction in forensic science, pp. 152–160.

[21] B. V. A. de Lima and V. P. Machado, "Machine learning algorithms applied in automatic classification of social network users," *4th International Conference on Computational Aspects of Social Networks - CASoN*, 2012.

[22] R. A. Fisher, "The use of multiple measurements in taxonomic problems," *Annals of Eugenics*, vol. 7, no. 7, pp. 179–188, 1936.

[23] P. Kulczycki and M. Charytanowicz, "A complete gradient clustering algorithm," in *Proceedings of the Third International Conference on Artificial Intelligence and Computational Intelligence - Volume Part III*, ser. AICI'11. Berlin, Heidelberg: Springer-Verlag, 2011, pp. 497–504.

[24] K. Bache and M. Lichman, "UCI machine learning repository," 2013. [Online]. Available: http://archive.ics.uci.edu/ml/