CLUSTERS LABELING THROUGH MULTI-LAYER PERCEPTRON ALGORITHM

Vinicius Ponte Machado*, Ricardo de Andrade Lira Rabêlo†, Lucas Araújo Lopes*

*Computer Department
Universidade Federal do Piauí (UFPI)
Teresina, Piauí, Brasil

†LABoratory of Intelligent Robotics, Automation and Systems (LABIRAS)
Universidade Estadual do Piauí (UESPI)
Teresina, Piauí, Brasil

Emails: vinicius@ufpi.edu.br, ricardor_usp@ieee.org, lucaslopes@ufpi.edu.br

Abstract— Clustering is a major problem that, among other areas, involves machine learning. As important as identifying the groups is to provide a definition for them. The present work proposes a model that uses a combination of algorithms with supervised and unsupervised learning with the goal of creating groups and identify which attributes may define them. In addition to the proposed model, this article presents the results of the all executions applied in one database.

Keywords— Machine Learning, Clustering, Labeling.

Resumo— Agrupamento é um grande problema que, entre outras áreas, envolve aprendizagem de máquina. Tão importante quanto a identificação de grupos é proporcionar uma definição para os mesmos. O presente trabalho propõe um modelo que utiliza uma combinação de algoritmos com aprendizagem supervisionada e não supervisionada com o objetivo de criar grupos e identificar quais atributos podem ser utilizados para defini-los. Além do modelo proposto, este artigo apresenta os resultados de todas as execuções aplicadas em uma base de dados.

Palavras-chave— Aprendizagem de Máquina, Agrupamento, Rotulação.

1 Introduction

The problem of clustering can be considered as one of the most important among those involving unsupervised machine learning algorithms. The goal is to partition a data collection in smaller structures (groups or clusters) which contain, somehow, similar elements under a particular perspective (Zhang et al., 2008). In addition, the elements belonging to the same cluster must possess enough dissimilarity to be distinguished from other groups. This problem is well studied in the literature involving several problems and techniques as Genetics Algorithms (Zhang et al., 2008), heterogeneous data sets (Abdullin and Nasraoui, 2012) and many others (Wang et al., 2012; de A.T. de Carvalho et al., 2012) showing several strategies which are used and presented in this paper in section 2.

However, another not very widespread aspect of this question deals with the problem of labeling. This problem lies in the fact of naming the clusters according to their common features. That is, to present a clear identification of the groups. A good definition of a group facilitates the work of the specialist while studying or interpreting data.

In the literature this problem is handled differently. However, a problem very similar is to label new elements based on groups already defined. This problems is presented in some works as (Eltoft and de Figueiredo, 1998), (Chen et al., 2005), (Chen et al., 2008). The main difference is that in these problems an new element can be classified on a predefined cluster according of the technique used. In this problem, a definition of each cluster will be given to, somehow, help the specialist. A good definition (or in other words, a label) of a cluster can be used to classify new elements too although this is not the aim here.

The unsupervised learning techniques are applied to a collection of data (database) and as a result there are several groupings. However, the methods used for grouping often fail to make its meaning clear. The present proposal in this work is to detect – combining with a supervised learning technique – what are the key features in each group, as well as their possible values in order to clarify, steer or help in any way with the analysis and labeling of groups held by experts.

2 Theoretical Framework

In our proposal it is necessary to use a unsupervised learning algorithm to accomplish the task of clustering. It doesn't make much sense to use a supervised learning algorithm because probably the attribute class will be the most important classifier and it will not be present in new elements. Therefore, it is important to use a unsupervised technique for this task.

Among several algorithms the technique chosen was *K-means* (Kanungo et al., 2002), which is relatively straightforward and also is presented on (de Lima and Machado, 2012). However, any other grouping algorithm could be used. In addi-

tion, according to the databases used for testing it is known *a priori*, the number K of clusters to be generated, which is a parameter of *K-means*.

Another step of the proposal will require the use of an algorithm with supervised learning and for that 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 (Haykin, 1998).

There are several techniques that address the problem of grouping: *Cobweb* (Fisher, 1987), *Self Organizing Maps* (*SOM*) (Figueiredo et al., 2012), *Fuzzy* (Bushong, 2007), *K-means* (Kanungo et al., 2002), in addition to hybrid techniques (Aziz et al., 2012), (Ramathilaga et al., 2011) among others.

The *K-Means* (Kanungo et al., 2002) 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 are 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 big, similar elements won't be grouped together. Whereas if it is too small, different elements will belong to the same cluster.

Artificial Neural Networks 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 (Haykin, 1998). The most basic neural network is the Perceptron (Yanling et al., 2002). The Perceptron is formed by 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) that follows the same idea of Perceptron network. The MLP is a network of the type feedforward, where there are at least two layers (a hidden one and an output one). Typically, The output values of a layer's neurons serve as input only for the neurons of the layer ahead.

3 Approach

Facing the problem of labeling presented in section 1, our proposal is to define a model with the objective of 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 Figure 1 demonstrates the proposal.

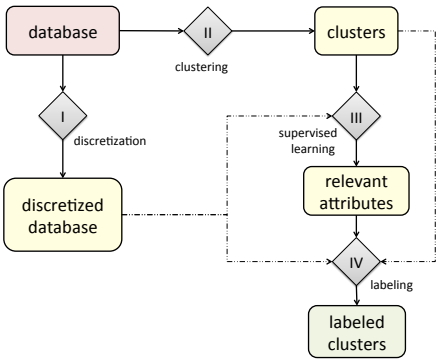

Figure 1: Proposal.

Initially, there is a database as entry. This database may have different types of data that according to their type (discrete/continuous) could be necessary to use a discretization model (I).

In order to the second algorithm obtains a better performance towards the continuous values, 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. 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.

3.1 Discretization (I)

Phase I consists on discretizing data: for the attributes that can take on different values among a continuous domain new discrete values will be established. This way, the algorithm with supervised learning will be able to more easily identify a possible relationship between attributes, showing better results in their classification, when dealing with continuous values.

In the literature (Cerquides and de Màntaras, 1997; Wang, 2000) there are several discretization methods. The two methods most commonly used are Equal Width Discretization (EWD) and Equal Frequency Discretization (EFD).

The discretization model used in this work is the EFD and uses some ranges (three, in this case) of values that contains 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 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 (Figure 2):

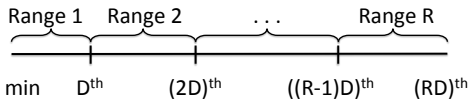

Figure 2: Ranges.

This model allows the unsupervised algorithm to work with ranges of values facilitating the detection of relevant attributes. A discretization method also ensure that the final label involving continuous attributes will be given in ranges of values instead of specific values.

The way in which it was used – three ranges of values defined by an equal quantity of distinct elements – is something to be discussed according to the circumstances. Discretized values are stored and will be used later during the steps III (training) as entry of the supervised algorithm and IV (labeling) as the limits of the intervals – ranges of values.

3.2 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 2, being the *K-means* the algorithm used here. In this step, we have a database as input and as output its elements grouped in K clusters.

3.3 Supervised Training (III)

In each generated cluster a supervised algorithm will be applied. The idea in this step is to detect which attributes are relevant to the group. For this, an ANN with supervised learning is applied for each attribute, where it is treated as an attribute class (output) and the others as network input, in order to find out which attributes may classify the group correctly. Figure 3 exemplifies this step, taking as an example a cluster in which its elements have *three attributes*.

For each attribute of the elements belonging to a given cluster an ANN will be created that will have as input the other attributes and will present as output the estimated value for the attribute concerned. Every ANNs of a same cluster has the same elements, varying only the way in which they are used in the network as input or output. The input values are not exact values, but the discretized values (if necessary) as calculated in step (I). The network output value will correspond to a range of values among the specified and according to the discretization model.

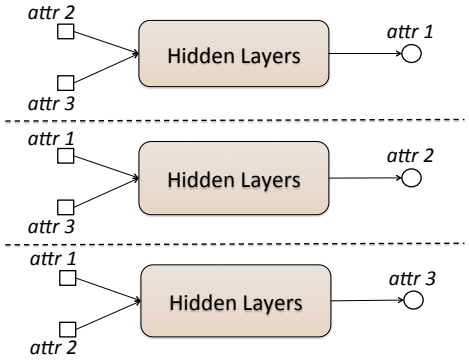

Figure 3: ANNs as supervised algorithm.

Considering any cluster, the database will have their elements divided into two parts (randomly for each network): training and testing. This process is known as cross-validation (Leisch et al., 1998) and is 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, there is a hit. Otherwise, there is the occurrence of an error.

That way, each ANN is created to represent and assess the importance of each attribute. In a wider way, each cluster will then have a hit rate for each ANN. That is, a hit rate for each evaluated attribute. That way we can know which attribute is relevant in relation to the other for a given cluster: is the one that got higher hit rate in the testing phase. For greater confidence regarding the attribute, there is an average of I iterations in this step. Each iteration corresponds to an ANN for each attribute.

3.4 Labeling (IV)

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

Another parameter, variation V, will select the other attributes that have a hit rate with variation of at most V (given in percentage) in relation to the main attribute. That way, we will have a

set of attributes that were seen as relevant to the definition of such cluster.

After setting the group of relevant attributes we confirmed which of the values (defined in the discretization step) dominates the group. That is, we detect what each attribute value range features more frequently in any cluster in that attribute taken as relevant. That way, 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.

The following algorithm demonstrates in a natural language, the proposed operations:

Require: Database
Ensure: *Labeled* clusters

```
Load database;
Discretize each continuous attribute (if necessary);
Perform clustering algorithm (unsupervised);
for each cluster do
    for each iteration i = 1 to I do
        Define training and test sets;
        for each attribute do
            Perform training (supervised);;
            Calculate the hit rate;
        end for
    end for
    Calculate the average hit rates;
end for
Label;
```

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).

4 Results

For the implementation of the proposed model we used the tool *MATLAB*[1], which enables the use of supervised and unsupervised algorithms presented in section 2 among others.

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 used parameters are patterns[2] according to *MATLAB*. In the database used here (Fisher, 1936) we know that *a priori* (as the author himself) the amount k of clusters that must be created.

To represent the ANN of MLP type we used the command *feedforwardnet ()*. In this algorithm we also used the standard settings from the neural network[3]. In relation to learning, 60% of 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, a quantity of ANNs by attribute (I) and a variation in relation to the higher hit rate by cluster (V), presented in section 3, result in a large number of possible combinations so that the results presented here represent only a small fraction of these combinations. The values used were $I = 10$, $V = 5$ and the discretization model was the *EFD*, with 3 ranges of values defined by the same quantity of distinct elements. 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, we will show the proposed model applied in one database.

4.1 Iris Identification

Database regarding the identification of iris (Iris Data Set) can be found in the data store *UCI Machine Learning* (Bache and Lichman, 2013). 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[4]: 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 after the implementation of the proposed model are presented in Table 1.

Table 1: Results of iris database.

		Result			Analysis	
Clus.	#Ele.	Attributes	Rel. %	Range	Error	Hit %
1	62	PL	83.6	3.7 ∼ 5.1	6	90.32
		SL	79.6	5.3 ∼ 6.4	13	79.03
2	38	PW	84.37	1.7 ∼ 2.5	3	92.10
		SW	82.5	2.7 ∼ 3.4	6	84.21
		PL	82.5	5.1 ∼ 6.9	2	94.73
3	50	PW	100	0.1 ∼ 1	0	100
		PL	100	1 ∼ 1.37	0	100

It is observed that the labeling is done according to clusters generated by *K-means* and that, as shown in Table 1, they can differ from the form

[1] www.mathworks.com/products/matlab/
[2] www.mathworks.com/help/stats/kmeans.html
[3] www.mathworks.com/help/nnet/ref/feedforwardnet.html

[4] 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.

suggested of the work presented in (Fisher, 1936) (50 elements in each cluster). Therefore, the labels presented here are specific and may differ at each performance according to the group performed.

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

As seen in Table 1, 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, that is, if the values of its attributes belong to the range shown. The Table 1 shows this analysis.

Only the main attributes define the labeling. That is, the attributes that have the best percentage of relevancy (as shown in Table 1). 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 variant V is used to select more attributes (less relevant) 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, the price to pay 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 clusters.

As we can observe in Table 1, the amount of elements clustered was different from (Fisher, 1936). 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 (Fisher, 1936).

As shown in Table 1, 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% considering only the PL attribute (which is already enough to distinguish the clusters) and 79,03% considering the SL and SW (which were appointed by a variant V).

Finally, the labels suggested by the proposal is: 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; PW ranging 0.1 and 1, PL ranging 1 and 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 variant V which is a result quite satisfactory.

5 Conclusion

A model for labeling was presented in this article. A unsupervised algorithm is used for the definition of the groups and, later, an algorithm with supervised learning is applied to each attribute of each cluster. The evaluation of unsupervised algorithms allows to identify which attributes are relevant to the problem. It is important to highlight that the discretization stage, held in continuous attributes, is important to this model.

The results are quite satisfactory assuming an average above 87%. It is 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. That way the unsupervised algorithm still has strong influence on the labeling result.

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

References

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

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

Bache, K. and Lichman, M. (2013). UCI machine learning repository.

Bushong, B. A. (2007). Fuzzy clustering of baseball statistics, *Fuzzy Information Processing Society, 2007. NAFIPS '07. Annual Meeting of the North American*, pp. 66–68.

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

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

Chen, H.-L., Chuang, K.-T. and Chen, M.-S. (2008). On data labeling for clustering categorical data, *Knowledge and Data Engineering, IEEE Transactions on* **20**(11): 1458–1472.

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

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

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

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

Fisher, D. (1987). Improving inference through conceptual clustering, *Proceedings of the sixth National conference on Artificial intelligence - Volume 2*, AAAI'87, AAAI Press, pp. 461–465.

Fisher, R. A. (1936). The use of multiple measurements in taxonomic problems, *Annals of Eugenics* **7**(7): 179–188.

Haykin, S. (1998). *Neural Networks: A Comprehensive Foundation*, 2nd edn, Prentice Hall PTR, Upper Saddle River, NJ, USA.

Kanungo, T., Mount, D., Netanyahu, N., Piatko, C., Silverman, R. and Wu, A. (2002). An efficient k-means clustering algorithm: analysis and implementation, *Pattern Analysis and Machine Intelligence, IEEE Transactions on* **24**(7): 881–892.

Leisch, F., Jain, L. and Hornik, K. (1998). Cross-validation with active pattern selection for neural-network classifiers, *Neural Networks, IEEE Transactions on* **9**(1): 35–41.

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

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

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

Yanling, Z., Bimin, D. and Zhanrong, W. (2002). Analysis and study of perceptron to solve xor problem, *Autonomous Decentralized System, 2002. The 2nd International Workshop on*, pp. 168–173.

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