New unsupervised classification scheme for datasets with presence of noise

José Felix Cabrera Venegas, Jarvin Antón Vargas, Yenny Villuendas Rey, Cornelio Yáñez-Márquez

Abstract. In this work an unsupervised classification algorithm, capable of handling data sets that have noisy elements is proposed. The proposed algorithm is based on a strategy to eliminate noise and then it applies a hierarchical agglomerative algorithm to obtain the groups. To determine the performance of the proposed algorithm, numeric experiments were performed and the quality of the proposal with respect to other algorithms were analyzed by using synthetic data sets developed specifically for this purpose; and also numerical databases from the Machine Learning Repository of the University of California at Irvine. The experimental results show that the new proposal has a stable performance and a good performance in noise elimination.

Keywords: Unsupervised clasification, noise detection.

1 Introduction

In recent years Pattern Recognition (PR) has gained some popularity thanks to automation of solutions to many problems of real life. In this area an important role is played by the Unsupervised Classification (UC) methods. The purpose of UC techniques is to reveal the structure of a set of data by creating groups of objects (clusters), associating each object with their most similar so that the objects in a cluster are different from the objects of other clusters [1].

In the literature, many UC algorithms have been proposed, but most of these attempts to group the data taking into account the presence or absence of certain features in the database. Many of the existing databases contains instances with rare or erratic behavior and many clustering algorithms are sensitive to such data, generating groups of poor quality. A noisy data (outlier) is an observation that deviate much from other observations and encourages suppose that was generated by different mechanisms [2]. Although there are proposals address this problem, many of them still exhibit the same deficiencies that must be settled.

Many UC algorithms try to minimize the influence of outliers or remove them all together, however this may result in a significant loss of information in certain situations. Fraudulent transactions such as credit card outliers are typical examples that may indicate fraudulent activity to be detected. That is why many applications of RP focus on the detection of noise, which is the main

result of the analysis of the data [3, 4] and not as such the classification of all data, on the other hand there are UC techniques engaged in the labeling of objects into groups and allow dealing with noise marking also with a label [5].

The state of the art techniques for the detection of noise is classified into several categories: distribution based techniques, based on depth, based on distances, based on density and based on groups, etc. [3]. Methods based on distribution and depth needs to adjust the data to statistical distributions or to assign depth to the data in a space of representation; and both become inefficient with the growing number of data. For these reasons there have been proposed methods that belong to other categories.

In [6], Knorr and Ng first proposed the notion of noise based on distance, which is a non-local approach and cannot find local outliers in databases with complex structure. Later Breuning et al. [7] presented the local outliers detection method based on density, Local Outlier Factor (LOF), supported by the distance of a point to their k nearest neighbor, declaring the n greater distance points as outliers. These proposals have the disadvantage of requiring the number of outliers to identify.

In addition to the actual noise detection techniques, there exist UC algorithms that allow detection of noise as a consecuence of the clustering process, as in the case of DBSCAN proposed by Ester et al. in [5]. DBSCAN discover the groups in dense regions of objects in the data space which are separated by regions of low density (elements of the latter regions represent noise). This method makes it possible to find groups of arbitrary shapes automatically generating the number of groups. However, it has the limitation of being sensitive to the initial value of its parameters, which determines the density of the groups to find and it does not works well finding groups of different densities and high-dimensional data.

Besides DBSCAN, other approaches have been published for noise management, one of the most recent is the APCOD (Automatic PAM Clustering algorithm for Outlier Detection) published in 2012 [4]. This method uses the classical clustering algorithm based on particional k-medioids PAM [8] and an internal index for automatic determination of the number of groups. Then using the LOF concept [7] are determined noisy data. This method has the disadvantage of not function properly with databeses of non-convex groups, as well as most centroids based methods. In addition, the number of outliers (noisemakers) to search is a parameter of the algorithm.

Actually, none of the above algorithms can group each data accurately, so some special data can be noise and be label as belonging to a group. Moreover, many of these and other proposals are limited in terms of assumptions about how to find groups, the dimensionality of the data and the density of the groups to obtain.

2 Proposed solution and experimentation planning.

We introduce a method with the ability to group data with noise by combining a strategy of noise detection and subsequent application of a classical clustering technique. Some clustering algorithms such as DBSCAN [5] allows to group data with noise but they require input parameters which are often difficult to determine. Moreover, there are other methods as APCOD [4] that first performs the grouping and then remove the noise, this often gives good results in the detection of noise but flawed in obtaining accurate clusters. The new proposal presented below is based on the design of a strategy to remove noise and the application of an agglomerative clustering algorithm. Unlike previous approaches, the proposed method allows greater accuracy in the production of "natural groups"of a data set with the presence of noise.

2.1 New proposal for data clustering in the presence of noise.

The proposed new method consists of two stages: noise elimination and clustering finding. For detection and noise elimination was designed a new strategy based on the concept of density using the k-nearest neighbors. In the case of obtain the clusters, it was adopted a classic strategy of UC as is the agglomerative single-link.

Outliers or noisy objects can be considered as objects or small groups located in regions of low density in contrast to the dense and larger structure formed by objects in the cluster. In this sense, there is an approach to identify outliers considering the number of existing objects around.

To identify regions of low density is necessary to determine for each object p the density $\delta_{Hr}(p)$ around it by counting the number of objects in the $H_{Hr}(p)$ hyperspace limited by certain hiperradio Hr.

This argument is not new; it has been used in the algorithms based on density, but their application needs of parameters such as the size of the neighborhood or hyperspace (Hr) and the threshold for which an object is considered noise. The novelty of the proposed method is the automatic determination of these parameters using the average nearest neighbor distance of all the objects d^{-}_{1-NN} and the average density for all δ^{-}_{Hr} objects using an iterative algorithm that remove the noise. The formulas for calculating these measures are offered in the following:

$$H_{Hr}(p) = \{q \in D \setminus \{p | d(q,p) \leq Hr\}\} \tag{1}$$

$$\delta_{Hr}(p) = |Hr(p)| \tag{2}$$

$$d^{-}_{1-NN} = \sum_{i=1}^{|D|} \frac{d_{1-NN}(i)}{|D|} \tag{3}$$

$$\delta^{-} = \sum_{i=1}^{|D|} \frac{\delta_{Hr}(i)}{|D|} \tag{4}$$

After having set noise-free data, we proceed to perform clustering using the single-link agglomerative strategy. The pseudo code of the proposed method is given:

Algorithm 1 Proposed Algorithm

Require: D: Dataset.
K: K-nearest neighbor.
G: Amount of non empty groups to obtain.
Ensure: C: Structuring in groups
First Stage: Noise detection
1: **repeat**
2: Calculate the nearest neighbor distance for all objects.
3: Calculate the average nearest neighbor distance for all objects.
4: **until** visit all K neighbors
5: Identifies and removes all the noisy elements (where $\delta_{Hr}(i) < \frac{1}{3}\delta^{-}_{Hr}$)
Secund Stage: Grouping data
6: Get the new free noise data base.
7: Run Agglomerative Algorithm Single Link.

Thus our proposal works in two stages, the first stage regarded to the noise detection and elimination and the second stage dedicated ti obtain clusters of the remaining data.

2.2 Experimental protocol

To carry out the tests we used nine synthetic databases, created by the user in a (CreatorData2D) tool implemented in Matlab (see the characteristic of each one in Table 1 and Fig. 2 shows the graphic representation) and eight actual databases from machine learning Repository of the University of California at Irvine (UCI Machine Learning Repository) [8], see table 2. The datasets of UCI repository it is known that in many cases have datasets with outliers that do not have labeled, it is for this we used this datasets too.

The choice of these datasets is that these are labeled databases, which facilitates the evaluation of the quality of the clusters obtained by the algorithms, using external validation indexes.

The algorithms against which the proposed method is compared are DBSCAN [5] and APCOD [4], being superior to other conventional techniques for detecting noise as LOF [7]. One of the most important design aspects of the experiments are the parameters needed to execute the methods. Like all

Table 1. Features synthetic data bases.

Datasets	Attributes	Groups	Class	Instances
ele	2	2	2	176
ruido	2	3	3	444
onda	2	2	2	306
anillo	2	2	2	806
base 1	2	3	3	910
base 2	2	3	3	472
base 3	2	3	3	518
base 4	2	3	3	637
base 5	2	3	3	451

Table 2. Features real data bases.

Datasets	Attributes	Groups	Class	Instances
iris	4	3	3	110
lymphography	18	4	4	148
cmc	9	10	10	1473
haberman	3	3	3	306
hayes-roth-train	4	3	3	110
liver-disorders	7	2	2	345
tae	5	3	3	151
wine	13	3	3	178

algorithms require knowledge of the number of groups to be formed, the value assigned to this parameter will match, for each database, with the number of classes. This allows taking labels classes of each database as a ground truth and; then compare the latter with the grouping obtained by the algorithms.

The rest of the parameters required by DBSCAN and APCOD are shown in Table 3 below. They were selected because these values are reported in literature as giving better results.

A Toshiba Laptop with the following characteristics was used for implementing and testing algorithms: Operating System Windows 7 Ultimate, 64-bit Microprocessor Intel Core i3 at 2.40GHz, with 2.65 GB RAM DDR3 memory at 665 MHz Dual-Channel, Motherboard Vendor PSKC8C-08G00R. It was also used MATLAB 7.0 Software [9], for numerical calculations with vectors, matrices, both real and complex scalar numbers, strings and other complex data structures, due to it has a basic code and several specialized libraries (toolboxes).

We use the Rand [10] index to compare the performance of the algorithms, over both synthetic and real databases.

Table 3. Parameters used in each algorithm.

Algorithms	Params
DBSCAN	Eps: radio of the the vicinity of each point - 0.3
	MinPts: minimum number of points in a neighborhood - 2
APCOD	α y β parameters for large and small groups alfa 0.75 y beta - 5

To process the data and draw conclusions on the performance of the algorithms the Wilcoxon test for related samples was used. It allows to stablished whether or not exist significant differences in the performance of each of the methods, with a 0.05 significance value. It should be noted that this test is recommended by Demsar [11] for such comparisons. In the next section the results of experimentation are described.

3 Experimentation and Results

In order to show the performance of the algorithms compared with the proposal (APCOD and DBSCAN), each of them in the test databases. Then the resulting clustering were evaluated using the Rand external index [10].

Rand is one of the indices of external validation most commonly used in the comparison of clustering algorithms. This index seeks to maximize the number of objects in the same group in the clusting to evaluate (CE), the ground truth clustering (GC) and the number of objects in different groups of CE and GC. Therefore, while higher is the value of Rand, more similar are clustering to the classes in the database analyzed. The Rand index is given by the following equation:

$$Rand = \frac{a+d}{\frac{N(N-1)}{2}} \tag{5}$$

where a is the number of pairs of objects in the same clusters in GC and CE, d is the number of pairs of objects in different clusters in GC and CE, N is the total number of objects.

The results of Rand index of the proposed algorithm over the others algorithms using synthetic databases are shown in Table 4 and real databases results are shown in Table 5.

Table 4 shows that the method of grouping DBSCAN have better performance (higher value of Rand) in six of the nine databases used in experimentation. The proposed algorithm wins in the database (base2) and ties in the database (ele) being secondly after DBSCAN while APCOD get third place.

To determine if the results are statistically significant Wilcoxon test is used for two related samples, which is recommended by Demsar [11]. For this, it was

Table 4. Rand index values of the HCOD algorithm and different methods applied to synthetic data bases.

Synthetic data bases	Proposed	APCOD	DBSCAN
ele	1.00000	0.70688	1,00000
ruido	0.98698	0.98191	0.99897
anillo	0.99504	0.49953	0.99998
onda	0.57495	0.51223	1,00000
base 1	0.92787	0.67001	0.99473
base 2	0.93999	0.63637	0.31098
base 3	0.97609	0.79633	0.98003
base 4	0.62388	0.56639	0.66244
base 5	0.85726	0.82193	0.92052

compared the proposed algorithm with each other. In each of the cases, it is set to null hypothesis that there are no significant differences between the performance of HCOD and the other algorithms. It is formulated as an alternative hypothesis, that the proposed algorithm has higher value of the Rand index than the other algorithms. The value of meaning that is adopted is 0.05 for 95% confidence. The Table 5 shows the results of the test.

Table 5. Wilcoxon test statistics for the Rand index of the algorithms (synthetic data bases)

	Algorithms	**Asymptotic significance**
PROPOSED ALGORITHM vs	APCOD	0.008
	DBSCAN	0.161

As shown in Table 5, the proposed algorithm outperforms the APCOD method. It is inferred from the value of asymptotic significance, which is less than 0.05, allowing reject the null hypothesis. However, there are no significant differences in performance of the proposed method with respect to DBSCAN algorithm, due to significance value of 0.161.

With regard to the actual databases in Table 6 Rand index values calculated are shown. There can be seen that the proposed algorithm has higher Rand index value than other methods in two databases (iris, hayes-roth-train) APCOD Rand has greater value in three databases (Lymphography, cmc, wine) than other algorithms and DBSCAN beats other methods in databases (haberman, liver-disorders, tae). DBSCAN and APCOD being tied by three in the first place with better performance and the proposed algorithm finishes second.

Table 6. Rand index values of the HCOD algorithm and different methods applied to real data bases.

Synthetic data bases	Proposed	APCOD	DBSCAN
iris	0.92727	0.85004	0.92344
Lymphography	0.28838	0.63164	0.23396
cmc	0.47599	0.55803	0.34430
haberman	0.57270	0.49182	0.61571
hayes-roth-train	0.65903	0.57819	0.34525
liver-disorders	0.49870	0.50266	0.50563
tae	0.51859	0.54623	0.62155
wine	0.62141	0.69009	0.65930

Applying the Wilcoxon test to these databases and using the same assumptions as used in comparison with previous methods, the table below shows the results obtained:

Table 7. Wilcoxon test statistics for the Rand index of the algorithms (real data bases)

	Algorithms	**Asymptotic significance (bilateral)**
PROPOSED ALGORITHM vs	APCOD	0.674
	DBSCAN	0.674

Table 7 shows that in the algorithms no significant differences are evident. Observed a good performance in these three methods.

4 Conclusions

In this paper, a new clustering algorithm which has the ability to handle noisy data sets of numerical type is proposed. The new proposal presented is based on the design of a strategy to remove noise and then applying an agglomerative clustering algorithm to obtain groups. Experimental results show that the proposed algorithm has good performance in synthetic databases and real data using as validation the Rand index. The proposed algorithm is stable and in some cases it has superior behavior removing noise with respect to other techniques. This method can be defined as a clustering algorithm in the presence of noise that can efficiently manipulate numeric data types obtaining efficient performance.

References

1. Xu, R. and D. W. II: Survey of Clustering Algorithms. IEEE Transactions on Neural Networks. 16, 645-678. (2005)
2. Hawkins, D. M.: Identification of outliers. New York, USA: Chapman and Hall. (1980)
3. Dianmin, Y., et al.: A Survey of Outlier Detection Methods in Network Anomaly Identification. Computer Journal. 54, 570-588. (2011)
4. Lei, D., et al.: Automatic PAM Clustering Algorithm for Outlier Detection. Journal of Software. 7, 1045-1051. (2012)
5. Ester, M., et al.: A density-based algorithm for discovering clusters in large special databases with noise. 2nd International Conference on Knowledge Discovery and Data Mining (KDD96). Portland, OR: AAAI Press. (1996)
6. Knorr, E. M. and R. T. Ng.: Algorithms for Mining Distance-Based Outliers in Large Datasets. 24rd International Conference on Very Large Data Bases. New York, USA. (1998)
7. Breunig, M. M., et al.: LOF: identifying density-based local outliers. ACM SIGMOD International Conference on Management of Data. New York, USA. (2000)
8. Asuncion, A. and D. J. Newman, : UCI machine learning repository. http://archive.ics.uci.edu/ml. University of California. Irvine, CA. (2007)
9. Jalon, J. G., J. I. Rodrguez, and J. Vidal: Aprenda MATLAB 7.0 como si estuviera en primero. 1-128. (2005)
10. Jain, A. K. and R. C. Dubes: Algorithms for Clustering Data. Prentice Hall. (1988)
11. Demsar, J.: Statistical comparison of classifiers over multiple datasets. The Journal of Machine Learning Research. 7, 1-30. (2006)