Classification Rules Obtained from Evidence Accumulation

Waldo Hasperué [1], Laura Lanzarini
III-LIDI (Institute of Research in Computer Sciences LIDI)
Faculty of Computer Sciences. National University of La Plata.
La Plata, Argentina, 1900
[1] Waldo Hasperué has a CIC Scholarship
{whasperue,laural}@lidi.info.unlp.edu.ar

Abstract. *This paper presents a machine learning approach applicable to Data Mining based on obtaining classification rules.*

It proposes a strategy to obtain classification rules from clusters resulting from a co-association matrix. Such matrix is obtained from the combination of different clustering methods applied to input data, and it has been selected by its result's robustness.

The proposed method has been applied to two set of data obtained from the UCI repository with really successful results. The results obtained in the classification have been compared to other existing methods showing the new proposed method superiority.

Keywords. Ensemble clustering, Evidence accumulation, Rules extraction, Data mining.

1. Introduction

At present, Computer Science advance allows us to count with a large volume of data obtained as result of the observation and/or sampling of different processes. The importance of these repositories lies in the possibility of applying intelligent processing into them, for decision taking.

In this area, Data Mining processes, which allow extracting knowledge in a non-supervised way, constitute the most accepted approach. Specially, clustering techniques have proved to have the capacity of identifying interesting patterns within the group. However, its application requires a distance measure which allows quantifying the similarity of input data.

One of the most used tools to solve clustering problems over these last years is the technique known as evidence accumulation [3]. Its main objective is to compare the result produced by different clustering strategies by means of a co-association matrix that allows measuring the similarity degree among input patterns.

There exist several studies on evidence accumulation [4] [12] [18] [20]. In all of these cases, even though the results obtained are robust and reliable, the task of establishing a representation of the obtained knowledge remains unsolved.

On the other hand, there exist several papers that extract rules of clusters obtained by different algorithms. In [13], [15], [17] and [19], new rule extraction methods are proposed, training feedforward networks; in [14] and [16] decision trees are used, in [9] genetic algorithms are used, and in [1] [2] [5] [7] [10] [11] SOM networks.

This paper proposes a strategy to obtain *classification rules* from a co-association matrix of input data whose values come from the training of a dynamic, competitive neural network using the AVGSOM method [6]. Such method has been selected basing on its capacity of preserving the topology of input data.

The next sections are organized as follows: First, section 2 presents a brief description of the general characteristics of the co-association matrix. Section 3 describes the rule extraction process. Section 4 analyzes the results obtained. Finally, section 5 presents the conclusions.

2. Co-association matrix

A co-association matrix is a structure capable of measuring input data similarity using the results obtained when running different clusterings. Those patterns belonging to the same cluster, when making different clusterings, will receive a value higher than those related in a few. The main idea is that of re-clustering the results of N different clusterings in single and final clustering.

In a set of data of m patterns, a co-association matrix T is created of mxm, where each element (i, j) of the matrix represents the similarity degree between patterns i and j, and is computed according to (1).

$$T(i,j)=\frac{\sum_{k=1}^{N} e_k}{N} \qquad (1)$$

where e_k is equal to 1 if patterns i and j belong to the same cluster in clustering k; otherwise it is 0.

From the co-association matrix it will be possible to determine the cluster quantity and the elements making up each of them.

The method used to obtain each of the initial groupings is the AVGSOM method [6] since it has proved to properly preserve the data topology, which allows obtaining an input with a high degree of reliability to build the co-association matrix. In the AVGSOM method, the quantity of neurons incorporated into the structure depends on an input pattern called *dispersion factor*. If the value used is high, the network obtained after training will have a large quantity of neurons. On the other hand, if a low dispersion factor is used, then the final network will have few neurons in its structure.

In order to get a more homogeneous result, the co-association matrix will be generated training the AVGSOM with three different values for the dispersion factor: one low, one intermediate, and one high. A low value will group the patterns of a wide neighborhood of the input space. The use of a high dispersion factor aims at reinforcing the association of a small group pattern neighborhood. This is really useful for the cases in which clusters are not clearly separated. Finally, the intermediate value dispersion factor is used so as to reinforce the relation between the clusters made up by the two previous cases.

The method for extracting clusters from the co-association matrix proposed in this paper consists in building up an ordered list with all the pattern pairs with a value higher than zero, from which only those exceeding threshold ψ will be analyzed.

With the first pattern pair of the list, the first clustering is created. For the rest of the pairs, its relation to the existing clustering is analyzed. If only one of the patterns is assigned to a group, it is also incorporated to the other. If patterns are assigned to different groups then its merging is analyzed. For this, the co-association between the elements of the pair selected from the list and each element of contrary group is controlled. If the pattern pair percentage, whose co-association is higher than ψ, exceeds a threshold γ, then clusterings are unified. Fig. 1 shows a pseudocode of the proposed algorithm.

```
REM define ψ and γ thresholds
L={(x, y, z)/z = T[x, y], z >= ψ}
C = {}
Repeat
  Obtain (x, y, z) such that z =
     max(w), ∀(a, b, w)∈ L
  cx = {r / r ∈ A, x ∈ A, A ∈ C)
  cy = {r / r ∈ A, y ∈ A, A ∈ C)
  If cx = Ø and cy = Ø then
        C = C + {{x, y}}
  end if
  If cx = Ø and cy <> Ø then
    If simil(x, cy) >= γ then
      cy = cy + {x}
    end if
  else if cx <> Ø and cy = Ø then
    If simil (y, cx) >= γ then
      cx = cx + {y}
    end if
  Else if cx<> Ø and cy <> Ø then
    If simil(y,cx) >= γ and
      simil(x,cy) >= γ then
      C = C - {cy}
      C = C - {cx}
      C = C + {cx ∪ cy}
    end if
  end if
  L = L - {(x, y, z)}
Until ( #L=0 )

Function simil (p, A)
  R = {r / T[p, r] >= ψ, ∀r ∈ A}
  return #R / #A
End
```

Figure 1. Clusterings obtaining from a co-association matrix.

3. Classification Rule Extraction

As result of the process explained in section 2, set C will be obtained, containing the input pattern clustering. For each cluster c_k, the corresponding hypercube H_k will be obtained as follows:

$$H_k = (H_{k1}, H_{k2}, \ldots, H_{km})$$
$$H_{kz} = (min_{kz}, max_{kz}) \ \forall z \in 1..m$$

where m indicates the input vector length, and min_{kz} and max_{kz} contain the minimum and maximum values found in feature z of the training patterns corresponding to cluster k, respectively.

If we wish to obtain classification rules, it will be necessary to count with additional information that allows us to associate each hypercube to a certain class. In this way, belonging to a given hypercube will mean belonging to a given class. This is represented by rules such as:

IF (*condition*) THEN *belong to class_j*

where the association between a hypercube (cluster) and a class to which the elements represented by it belong is externally (supervised) established. It is important to notice that otherwise, rules will just only be categorizations of the groups, incapable of making reference to a specific class.

The method here proposed allows obtaining an ordered sequence of rules, in which each of them identifies elements of a class. The order of rules within a sequence is determined by its precision, being the first that with the most precise classification.

In order to establish this order it is necessary to identify clusters with lesser overlapping. d_{kjz} will be used to refer to the overlapping between clusters c_k and c_j belonging to C for the feature z. The value of d_{kjz} is proportional to the overlapping between clusters and will be less than zero when there does not exist overlapping between clusters c_k and c_j for the feature z, while a positive value will indicate the opposite.

Rule extraction is an iterative process which attempts to represent each of the clusters indicated by C, starting by those presenting less overlapping.

The generated rules are incorporated in the *Rules* array. Once this process is finished, and in order to make a classification, these rules should be run in order, beginning by the array index 1 until the corresponding class is found. This mechanism implies that the used training patterns should properly characterize the input space; otherwise, it will be necessary to incorporate representatives so as to establish a "*reject*" class.

Fig. 2 shows the pseudocode with the selection of clusters or hypercubes to build the rules.

Once the cluster for which the rule is to be built up has been chosen, it is necessary to analyze its interval features considering first the most significant, i.e., those allowing its separation from most of the remaining clusters. If there exist several features with the same importance, that presenting the lesser overlapping with the rest of the clusters should be selected.

The selected feature interval will be used to build up the rule antecedent condition. Then, features are selected until the indicated cluster is separated from the rest. Fig. 3 sums up the way to obtain such rule. Notice that, when the rule is incorporated, the Class(t) process allows obtaining the class represented by cluster t. It is important to notice that between the clusters to be separated, overlapping may occur. The process *Adjust_Limit* aims at minimizing it, adjusting the corresponding interval (Fig.4).

```
$C = \{c_1, c_2, \ldots, c_L\}$ set of L clusters
      obtained from the co-association
      matrix
$H_k = (H_{k1}, H_{k2}, \ldots, H_{km}) \; \forall c_k \in C$
i=0
Repeat
  $minmax = \min(\max(H_{kz}), \max(H_{jz}))$
  $maxmin = \max(\min(H_{kz}), \min(H_{jz}))$
  $d_{kjz} = minmax - maxmin$, $z=1..m$;
                          $k,j=1..L$, $j \neq k$
  $priority_{kz} = \#\{d_{kjz}<0 \;/\; k,j=1..L,\; j \neq k\}$
  $maximum = \max(priority_{kz})$ $k=1..L$, $z=1..m$
  $T = \{c_t \;/\; priority_{tz} = maximum,\; c_t \in C,$
      $\exists z \in [1..m]\}$
  $Intersec_{kz} = sum(d_{kjz})$ with $d_{kjz}>0$,
      $c_k \in T, j=1..K, j \neq k, z=1..m$
  Obtain t such that ($Intersec_{tz} =$
     $\min(Intersec_{kz})$) with $c_k, c_t \in T$,
     $z=1..m$ and ($priority_{tz} = maximum$)
  Rules[++i]=Generate_rule(t, $priority_t$,
                          $Intersec_t$, $d_t$, H)
  $C = C - \{c_t\}$
Until ( #C=1 )
```

Figure 2. Cluster selection algorithm for building up rules

4. Results

The rule extraction method presented in this paper was tested with two set of data of the UCI's repository: Iris Plants Database and Wine recognition data [8].

The experiment consisted in training different dynamic competitive neural networks with the AVGSOM method for three different dispersion factors: low, intermediate, and high. Values used were 0.01, 0.2, and 0.7 respectively. In order to obtain stable results, each training was repeated 15 times, which gives a total of 45 trainings for the same database.

Table 1 shows the result of this stage applied to both data bases and compared to the distribution obtained with the assignation of original classes. The information of the class to which each pattern belongs is given in the respective databases.

```
function Generate_Rule(t, $priority_t$, $Intersec_t$, $d_t$, var H)
    Better = {s / $priority_{ts}$ = max($priority_{tz}$) with z=1..m}
    Obtain r such that ($Intersec_{tr}$ = min($Intersec_{tz}$))with z∈ Better
    Rem_Clusters= C - {$c_t$}
    Adjust_Limits(t, r, Rem_Clusters, $d_t$, H)
    Condition = {(feature_r ∈ [min($H_{tr}$), max($H_{tr}$)] ∀ Hp ∈ Ht, ∀r∈1..m }
    Rem_Clusters = Rem_Clusters - {$c_j$ / $d_{tjr}$<0, $c_j$ ∈ C, j≠t }
    Remaining_Feat= {1..m} - {r}
    While (#Rem_Clusters> 0)
        $Priority_{tz}$ = #{$d_{tjz}$<0, $c_j$ ∈ Rem_Clusters, z ∈ Remaining_Feat }
        $Intersec_{tz}$= sum($d_{tjz}$) con $d_{tjz}$>0,$c_j$∈ Rem_Clusters, z∈ Remaining _Feat
        Better= {s / $priority_{ts}$ = max($priority_{tz}$ ) con ∀z∈ Remaining _Feat}
        Obtain r such that($Intersec_{tr}$ = min($Intersec_{tz}$))with z∈Better
        Adjust_Limits (t,r, Rem_Clusters, $d_t$, H)

        Condition = Condition OR {(feature_r ∈ [min($H_{tr}$), max($H_{tr}$)] ∀ Hp ∈
                Ht, ∀r∈1..m }
            Rem_Clusters = Rem_Clusters - { $c_j$/$d_{tjr}$<0, $c_j$∈ Rem_Clusters, j≠t
                                 }
        Remaining _Feat = Remaining _Feat - {r}
    end While
    return(« if » + condition+ «then is_of_class » + Class(t))
end process
```

Figure 3. Generation of the classification rule for cluster t

```
process Adjust_Limits(t, r, Remaining_Clusters, $d_t$, var H)
    Overlapped = {$c_j$ / $d_{tjr}$>0, $c_j$ ∈ Remaining_Clusters }
    Solutions = {}
    For each $c_j$ ∈ Overlapped
        Solutions = Solutions + {(v, $H_1$, $H_2$, $H_t$) / $H_1$, $H_2$ ∈ Ht are
              hypercubes resulting from dividing Ht, such that they do
              not overlap with $H_j$, ∀a∈1..m, v is the sum of volumes of
              $H_1$ and $H_2$}
        Solutions = Solutions + {(v, $H_1$, $H_2$, $H_j$) / $H_1$, $H_2$ ∈ $H_j$ are
              hypercubes resulting from dividing $H_j$, such that they do
              not overlap with $H_t$, ∀a∈1..m, v is the sum of volumes of
              $H_1$ and $H_2$}
    end for
    Best_solution = (v, $H_1$, $H_2$, $H_m$) ∈ Solution such that v is minimum
    H = H - {$H_m$}
    H = H + {{$H_1$, $H_2$}}
end process
```

Figure 4. Adjustment of r feature interval of cluster t

Table 2 shows the rules built from the found clusters for the two sets of data using the proposed method.

Table 3 represents the exactness of such rules in relation to the clustering resulting from the co-association matrix. For all the clusters found in each data base, its total data pattern is shown, as well as how many were properly classified with rules obtained by the method proposed in this paper.

Finally, the method proposed in this paper was compared to the results presented in [7]. In this latter, the initial clusterings were obtained directly from a dynamic, competitive neural network trained with AVGSOM with a 0.9 dispersion factor. Table 5 shows rules reliability for each of the methods. Such reliability is measured as the quantity of data patterns properly classified by the rules, dividing the total of patterns.

Table 1. Comparison between the clustering built from the co-association matrix and the original class distribution.

Iris Plants Database			
	Iris-versicolor	*Iris-virginica*	*Iris-setosa*
C1	0	0	50
C2	49	4	0
C3	1	46	0
Wine Data Base			
	A	B	C
C1	59	2	0
C2	0	2	44
C3	0	67	4

Table 2. Extracted Rules.

Data Base	Extracted Rules
Iris	If (Petal-length ≤ 1.9) then C1 Else If (Petal-width ≤ 1.7) and (Petal-Length ≤ 5.6) then C2 Else C3
Wine	If (OD280/315 of diluted wines > 2.5) and (Proline ≥ 650) and (Color intensity ≥ 3.35) then C1 Else If (Alcohol < 12.2) or (Hue > 0.96) or (Flavanoids > 1.59) or (Malic acid < 1.67) then C2 Else C3

Table 3. Exactness of the obtained rules

Data Base	bc / N	%
Iris Plants Database		
C1	50 / 50	100%
C2	53 / 53	100%
C3	47 / 47	100%
Total	150 / 150	100%
Wine Database		
C1	60 / 61	98.36%
C2	44 / 46	95.65%
C3	48 / 48	100%
Total	175 / 178	98.31%

Table 4 shows the effectiveness of rules discriminating each cluster's patterns according to the class they belong.

After applying the proposed method in several situations, we have observed that, in general, it properly carries out the expected clusterings. In addition, the use of hypercubes - as representative structure of the input space coverage on the part of each competitive neuron - eases rule construction. However, the use of hypercubes presents restrictions inherent to its shape which, in some cases, makes the method incur classification errors. Without doubts, the solution of this problem is the choice between the simplicity of the rule to obtain and the expected clustering error bound.

5. Conclusions

A strategy to obtain *classification rules* from co-association matrix of input data whose values arise from training a dynamic, competitive neural network using the AVGSOM method has been presented.

The rule extraction method proposed and used in this paper is an improved version of [7]. This last has proved to be better than other existing solutions which also obtain classification rules.

Table 4. Effectiveness of the obtained rules

	Clusters			bc / N	%
Iris	C1	C2	C3		
Versic	0	49	1	49 / 50	98%
Virgin	0	4	46	46 / 50	92%
Setosa	50	0	0	50 / 50	100%
Total				145 / 150	96.67%
Wine					
A	59	0	0	59 / 59	100%
B	2	2	67	67 / 71	94.37%
C	0	44	4	44 / 48	91.67
Total				170 / 178	95.5%

Table 5. Reliability of the obtained rules

	Method			
	Co-association		[Has06]	
Iris	150/150	100%	148/150	98.67%
Wine	175/178	98.31%	169/178	94.94%

This proposal superiority in the results is due to two factors. On the one hand, the way of establishing in a non-supervised fashion the initial clusterings by means of a co-association matrix has been improved. On the other hand, the strategy to separate overlapping hypercubes has been modified thus helping improve the precision of rules.

The application of this new method to two sets of data of the UCI repository has been successful and the rules obtained in each case have proved to have a higher reliability than those presented in [7].

6. References

[1] Alahakoon D, Halgamuge SK Srinivasan B. Data Mining with Self Generating Neuro-Fuzzy Classifiers. IEEE International Fuzzy Systems Conference Proceedings; 1999 Aug 22-25; Seoul, Korea. p. 1096-101.

[2] Azcarraga AP, Hsieh, M-H, Pan SL, Setiono R. Extracting Salient Dimensions for Automatic SOM Labeling. IEEE Transactions on Systems, Man, and Cybernetics-Part C 2005. 35(4): 595-600.

[3] Fred ALN, Jain AK. Data clustering using evidence accumulation**.** Proceedings. 16th International Conference on Pattern Recognition 2002. 4: 276-80.

[4] Fred ALN, Jain AK. Combining Multiple Clusterings Using Evidence Accumulation. IEEE Transactions On Pattern Analysis And Machine Intelligence 2005. 27(6): 835-50.

[5] Hadzic F, Dillon TS. CSOM: Self-Organizing Map for Continuous Data. 3rd IEEE International Conference on Industrial Informatics; 2005 Aug 10-12; p 740-5.

[6] Hasperué W, Lanzarini L. Dynamic Self-Organizing Maps. A new strategy to upgrade topology preservation. XXXI Congreso Latinoamericano de Informática CLEI; 2005 Oct 10-14; Cali, Colombia. p. 1081-7.

[7] Hasperué W, Lanzarini L. Classification Rules obtained from Dynamic Self-organizing Maps. VII Workshop de Agentes y Sistemas Inteligentes (WASI); 2006 Oct 17-21; San Luis, Argentina.

[8] Hettich S, Blake CL, Merz CJ. UCI Repository of machine learning databases. University of California, Irvine, Dept. of Information and Computer Sciences; 1998. http://www.ics.uci.edu/~mlearn/MLRepository.html.

[9] Hruschka ER, Ebecken NFF. Using a clustering genetic algorithm for rule extraction from artificial neural networks. IEEE Symposium on Combinations of Evolutionary Computation and Neural Networks; 2000 May 11-13; p. 199-206.

[10] Hsu C-C, Wang S-H. An Integrated Framework for Visualized and Exploratory Pattern Discovery in Mixed Data. IEEE Transactions on Knowledge and Data Engineering 2006. 18(2): 161-73.

[11] Huang Z, Hu Y-Q. Applying AI technology and rough set theory to mine association rules for supporting knowledge management. Proceedings of the Second International Conference on Machine Learning and Cybernetics, Xi´an; 2003 Nov 2-5.

[12] Kuncheva LI, Vetrov DP. Evaluation of Stability of k-Means Cluster Ensembles with Respect to Random Initialization. IEEE Transactions On Pattern Analysis And Machine Intelligence 2006. 28(11): 1798-808.

[13] Lu H, Setiono R, Liu H. Effective Data Mining Using Neural Networks. IEEE Transactions on Knowledge and Data Engineering 1996. 8(6): 957-61.

[14] Sato M, Tsukimoto H. Rule extraction from neural networks via decision tree induction**.** Proceedings. IJCNN '01. International Joint Conference on Neural Networks; 2001. Jul 15-19; p. 1870-5.

[15] Setiono R, Liu H. Symbolic representation of neural network 1996. 29(3): 71-7.

[16] Setiono R, Leow W. Generating Rules from trained Network using Fast Pruning. International Joint Conference on Neural Networks; 1999 Jul 10-16; p. 4095-8.

[17] Setiono R. Extracting M-of-N Rules from Trained Neural Networks. IEEE Transactions on Neural Networks 2000. 11(2): 512-9.

[18] Strehl A, Ghosh J. Cluster Ensembles: A Knowledge Reuse Framework for Combining Multiple Partitions. Journal of Machine Learning Research 2002. 3: 583-617.

[19] Taha IA, Ghosh J. Symbolic Interpretation of Artificial Neural Networks. IEEE Transactions on Knowledge and Data Engineering 1999. 11(3): 448-63.

[20] Topchy A, Jain AK, Punch W. Clustering Ensembles: Models of Consensus and Weak Partitions. IEEE Transactions on Pattern Analysis and Machine Intelligence 2005. 27(12): 1866-81.