Supervised Machine Learning by Generation of Rules: One Class Against All Others

A. Borgi and A. Halioui

***Abstract*—Several methods and techniques for data classification are available within the framework of supervised induction. In supervised machine learning by generation of rules, classification rules are automatically built at the learning phase. At classification phase, an inference engine uses rules to classify new objects. SUCRAGE is a machine learning system under this framework where rule is about: IF [premise] THEN [conclusion]. "Conclusions" are hypotheses about the membership of a "premise" in a given class; while "premise" regroups correlated features. The key of this classification is, then, about discovering the right dependencies to construct the right rules. We propose here a new form of discovering dependencies within data sets. This method consists of generating *n* classifiers; each one is specialized in the recognition of one class. The final rule base is a conjunction of all these particular classifiers. A classifier is a group of rules where dependencies are discovered from two data subsets. The first one contains instances belonging to the same class. The second one regroups the instances of all other classes. This method, also known by OAA (One-Against-All) or OVA (One-Vs-All), is used in various learners such as in decision trees or SVM. We present here experimental tests with several datasets. The obtained results and their comparison with other methods of discovering dependencies are also given.**

***Index Terms*—Supervised learning, Correlation, Rules generation, One against all**

I. Introduction

IN inductive training process we try to derive a complete and correct description of a phenomenon, starting with labeled data and specific observations of this phenomenon to make conclusions and predictions. In our case, the process is about training data instances to obtain a rules' base. SUCRAGE [1] is a supervised learning system by rules generation; rules are generated from a multi-attribute selection. This selection is done by research for dependencies and correlations between the components of training vectors [2]. In mono-attribute methods, the construction of rule's conditions is done by one attribute at a time. This could ignore the fact of dependencies that could exist between data attributes. Unlike mono-attribute methods, SUCRAGE, as a multi-attribute method, allows the exploitation of the eventual predictive power of a block of attributes. In discovering dependencies between features, two methods were proposed in the first version of SUCRAGE [1]; either discovering correlations within all training vectors (all classes merged research) or within vectors of one class (Intra-classes research). In this precise context, we propose a method for discovering dependencies by confronting instances of one class to all others. This could release discriminative information about a class.

This paper is organized as follow. First we introduce SUCRAGE, a supervised classification system by generation of rules. Then we present our method of discovering dependencies among all previous others. Finally we provide experimental support for this method and conclude with the principal prospects of this study.

II. SUCRAGE – SUpervised Classification by Rules Automatic GEneration

SUCRAGE is a multi-attribute supervised classifier by rules generation. The process of classification is done through two phases: Training and Recognition. In the training phase we try to generate a base of rules from labeled examples. The second phase classifies new (unlabeled) instances with the generated rules. A rule based classifier generally uses a set of IF-THEN rules for classification. In SUCRAGE an IF-THEN rule is an expression of the form:

$$\text{IF } A_1 \text{ AND } A_2 \text{ AND} \ldots\ A_K \text{ THEN } y, \alpha$$

--A_i: a condition of the form X_j in [a,b].

--X_j : the jth component of the vector representing an example.

--[a,b] : the interval resulting from discretization of the fields of an attribute, here from X_j.

--y: an assumption on the membership to a class.

--α : a degree of belief representing the uncertainty of a conclusion.

SUCRAGE is a polythetic training method looking for regrouping attributes into blocks. The selection of a block of attributes is done by a research of correlations between the components of the training vectors. Locating dependencies is done by statistical measures [3], [4], [5]. In [6] different measures are proposed to identify correlations and dependencies within a sample of datasets:

--Numerical features: a linear correlation of *Bravais Pearson* is used;

A. Halioui, "Institut Supérieur d'Informatique et de Gestion de Kairouan (ISIGK)", Kairouan, Tunisia (e-mail: halioui_ahmed@live.com).

A. Borgi, "Institut National des Sciences Appliquées et de Technologie (INSAT)", Tunis, Tunisia (e-mail: amel.borgi@insat.rnu.tn).

--Ordinal features: *Kendall's* and *Spearman's* rank correlations are used;

--Symbolic features: *Chi squared* independence test, *V Cramer* factor and *Symmetrical Uncertainty* are used.

The idea is to locate privileged correlations between the attributes of training vectors and to generate rules considering these correlations: the correlated features are gathered in the same premise. Generating rules requires then some specific points:

--Which features should be gathered together?

--How should values be associated to each feature, i.e. how should intervals [a,b] be chosen?

--How is the premise of a rule built?

--How should an uncertainty of a rule be represented?

A. *Correlated Components*

The first stage consists on calculating the correlations' matrix, R, between the components of the training set vectors. Let p be the number of features, R is noted by: R= $(r_{i,j})_{\substack{1\leq i\leq p \\ 1\leq j\leq p}}$.

$$R = \begin{bmatrix} 1 & r_{1,2} & \cdots & r_{1,p} \\ r_{2,1} & 1 & \dots & r_{2,p} \\ \cdots & \cdots & \cdots & \cdots \\ r_{p,1} & r_{p,2} & \dots & 1 \end{bmatrix}$$

$r_{i,j}$ is the measure of dependency between two features X_{i} and X_{j}.

The following stage is to determine the threshold of the matrix R. We decide that X_{i} and X_{j} are correlated if the absolute value of $r_{i,j}$ is higher than a threshold θ that we fix. The decision procedure is then defined by:

If $|r_{i,j}| < \theta$ Then X_{i} and X_{j} are not correlated And $r_{i,j} = 0$;
Else $r_{i,j} = 1$.

For example, in the case of a dataset examples represented by 5- components vectors, the thresholded correlation matrix obtained is :

$$R = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 \end{bmatrix}$$

From this thresholded matrix, we can extract correlated components: the features that should be gathered together in the same premise. For this example we will then group together (X1,X2,X3) and (X4, X5).

B. *Discretization*

Discretizing numeric features is widespread and recommended in several works [7], [8]. A lot of methods are proposed to identify the way how a feature should be split. We distinguish two different categories: supervised and unsupervised methods. In a regular unsupervised discretization, each feature is cut according to a number of intervals already fixed by the expert. *M* equal subintervals are obtained, so *M* regions *rg_i* for each attribute represent the new values. In supervised discretization methods as in MDLPC [9], cut-off points are chosen properly. This could obviously influence the accuracy of our classifier. Certainly, discretization leads to a loss of information and there's no substitute for precise numeric information, but this phase is delayed in training process as far as it could be. SUCRAGE discretize features to build premises then rules (see following section).

C. *Premises construction*

After discretizing the numeric features, for each correlated component and for each attribute we build premises with all possible combinations of rg_i. Fig. 1 illustrates a premise's construction in the case of 2 correlated features $\{X_4, X_5\}$ and M=4 discretization subintervals.

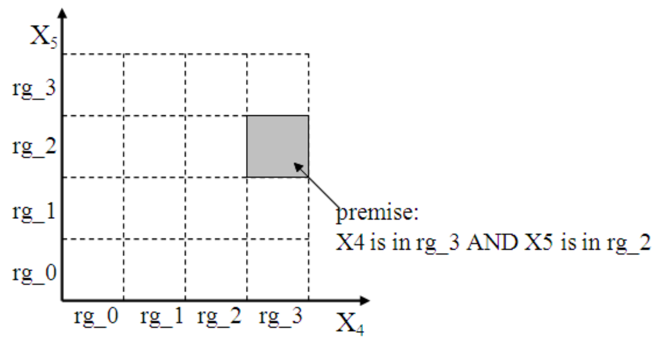

Fig. 1. Example of space's partition of two correlated attributes

D. *Rules conclusions*

After constructing premises, assigning conclusions to each premise is the next step. This step leads to generate *C* rules (*C* is the number of classes). So for each premise all possible conclusions are generated. Conclusions are assumptions on the membership of a class with some degree of certainty. The relevance of each generated rule is evaluated compared to the density of the examples belonging to the premise of the considered rule. A degree of belief is associated with each rule. In this article we propose to represent this degree by conditional probability. Such degree is estimated on the training set according to frequency approach.

III. Discovering correlations in SUCRAGE

The key of SUCRAGE method is to locate attributes dependencies. We present here three different scenarios of discovering these dependencies within the training vectors of a dataset. Two of them (all class merged together and intra-classes) are already presented in [1]. A new strategy of discovering correlations is, then, proposed. All these methods describe the way we use correlations measures (linear correlation, *V Cramer* … as previously described) in order to construct a rule-based classifier.

A. *All class merged together*

In an all-class merged together approach [1]; rules are built from correlated components discovered in *ALL* training vectors of a dataset. The whole instances contribute to the generation of correlated components and then to one classifier (classification rules). This method is carried out on all the units of training sets without distinction of class. Thus, the research for correlations is done within examples of various classes.

B. *Intra-classes*

This method is more original than the first one. In this method, correlations and dependencies are sought after between the instances of the same class. It's better to focus on examples belonging to the same class and seek correlations within these data sets. This method leads to better characterize a class and release some of its properties. Each class is thus characterized by its own matrix of correlations and its own set of correlated attributes [1]. For *C* classes we obtain *C* matrixes. The final correlated features gather (union) all found correlated components (from the *C* matrixes) to make one base of rules.

C. *One class against all others*

The one against all is an original technique used to solve multiclass problems for binary learners like SVM, decision trees and Neural Bayes [10]. This algorithm can also be used in a more general classification problem "multi-label classification". In [11] the classifier for a class *i* is trained to predict "Is the label *i* or not" thus distinguishing examples in class *i* from all other examples. Predictions are done by evaluating the n examples and randomizing over those which predict "yes" or over all *n* labels if all answers are "no".

We try to apply here the same scheme but with some modifications to make it fit with our framework. The main goal is to build multiple classifiers; each one is specialized in predicting a certain class *yi*. These classifiers are built from correlated components discovered from confronting examples of a class *yi* to all others together. 2*C* (the number of the class labels) different sub-datasets are used then for the research of correlated components two by two. The dataset Ω is then transformed into C different sub-datasets: $\{\Omega_{y1},\Omega_{\bar{y}1}\},\{\Omega_{y2},\Omega_{\bar{y}2}\},\dots,\{\Omega_{yc},\Omega_{\bar{y}c}\}$.

From confronting each two sub-datasets, Ω_{yi} (contains the examples of one class yi) and $\Omega_{\bar{y}i}$ (contains the examples of all classes except yi), 2 different set of correlation components, CC_{yi} and $CC_{\bar{y}i}$ found from correlations' matrix Ryi and $R\bar{y}i$. 3 kinds of correlated components are then discovered (Fig. 3). More precisely, we distinguish the following sets of correlated components:

--$CCS_{yi} = CC_{yi} \setminus CC_{\bar{y}i}$: these correlated components are only found from Ωy_i. These components could predict a class yi easily. So, we assign to premises created from these correlated components only the class yi.

--$CCS_{yi,\bar{y}i} = CC_{yi} \cap CC_{\bar{y}i}$: found from both sub-datasets Ωy_i and $\Omega\bar{y}_i$. These components could predict either examples from a class yi or else. This leads to create more general rules and premises created from this kind of correlated components are assigned to all the C classes.

--$CCS_{\bar{y}i} = CC_{\bar{y}i} \setminus CC_{yi}$: found from the non-class *yi* sub-datasets $\Omega_{\bar{y}i}$. These correlated components could predict examples not belonging to *yi*; so we assign to premises created

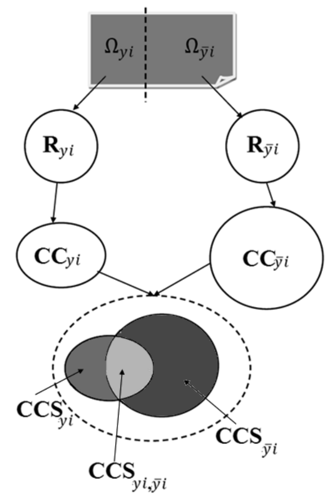

Fig. 3. One class against all others correlations

from this type of correlated components all the classes except yi.

The algorithm of one class against all others is given below:

for $i \in \{1,\dots,C\}$:

1) Regroup all the examples belonging to the class yi in Ωy_i and all others into $\Omega\bar{y}_i$.
2) Look for correlations (as described before) on each sub-dataset. Two correlations' matrixes are then calculated Ryi and $R\bar{y}$i.
3) Threshold Ryi and $R\bar{y}$i and extract correlated components CC_{yi} and $CC_{\bar{y}i}$ from each matrix.
4) Let $CCS_{yi} = CC_{yi} \setminus CC_{\bar{y}i}$ and $CCS_{\bar{y}i} = CC_{\bar{y}i} \setminus CC_{yi}$
   a) CCS_{yi} allows to create premises labeled with only yi.
   b) $CC_{\bar{y}i}$ allows to create premises labeled with all classes yk besides y_i ($k \in \{1,\dots,i-1,i+1,\dots,C\}$).
   c) $CCS_{yi,\bar{y}i} = CC_{yi} \cap CC_{\bar{y}i}$ allows to create rules labeled with all classes.

--CCS_{yi} components allow to construct a base of rules denoted by BR_{yi}.

--$CC_{\bar{y}i}$components allow to construct a base of rules $BR_{\bar{y}i}$.

--$CCS_{y,\bar{y}} = \bigcup_{i=1}^{C} CCS_{yi,\bar{y}i}$. These components allow to construct the base of rules $BR_{y,\bar{y}}$.

--$BR_f = \bigcup_{i=1}^{C}(BR_{yi}\ U\ BR_{\bar{y}i}) + BR_{y,\bar{y}}$ is the final classifier.

--Each $BR_{yi}\ U\ BR_{\bar{y}i}\ U\ BR_{y,\bar{y}}$ represents a specific classifier for the recognition of yi.

Consequently, *C* classifiers are created. Each one is specialized in the recognition of one class. All classifiers represent the final learner, which we use in training.

The whole process is summarized in the Fig.4.

The number of rules always depends on the fixed threshold θ and M (the cardinal of a subdivision). Some rules will simply not be generated if the coefficient of belief reaches 0. We are more interested in the accuracy of the classifier than in the number of generated rules. Several methods of number rules base optimization have already been discussed in [12], [6].

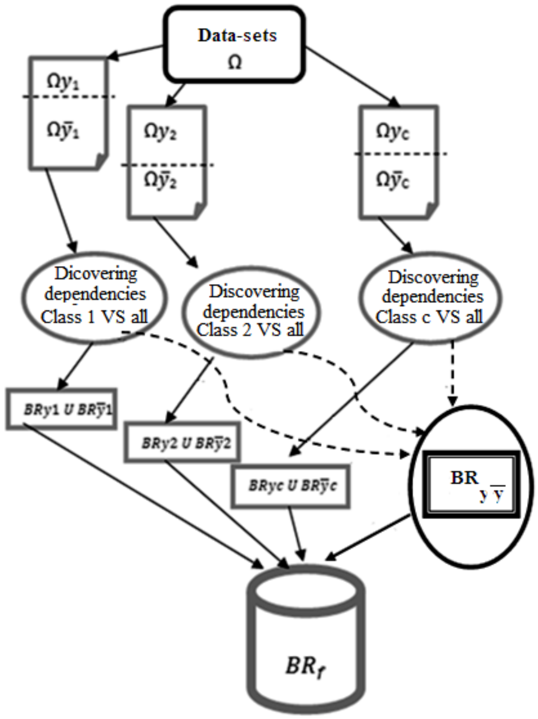

Fig. 4. One against all others classification

IV. Experimental results

In order to assess the accuracy of the one class against all method, it was tested on several datasets (numeric and symbolic) from the UCI Machine Learning system Repository[1] and PROMISE Repository[2]. We present here the details of the results obtained with the Iris dataset. Other results, with different datasets, are presented in annex. The accuracy is computed as the percentage of correct classifications. All experiments were carried out using a 10-fold cross-validation approach.

Our study is about comparing the effectiveness of OAA approach in comparison with the previous techniques of correlations research. All experiments were tested with different data types (numerical, categorical and mixed) and different number of discretization intervals and thresholds. The accuracy ratios (at the first cell's line) and the number of generated rules (at the second) are shown in each table. The first four tables study and compare the accuracy of one class against all others with inter-classes (all classes merged together) and intra-classes classification, within the data sets of Iris. The first three tables represent results obtained with Pearson linear correlation. Each table shows results of a fixed number of discretization intervals (3, 5 and 7). Ratios 1 and 2 show the effectiveness (percentage of Correct Classifications (CC)) of intra-class compared to the classic inter-class classification and our approach as well.

Ratio 1 = $CC_{Intra\text{-}Classes}/CC_{Inter\text{-}Classes}$

Ratio 1 = $CC_{OAA}/CC_{Inter\text{-}Classes}$

[1] ftp ://ftp.ics.uci.edu

[2] http://promisedata.org

TABLE I

Iris Dataset, Unsupervised Discretization (3), linear correlation. A comparison of OAA classification with inter and intra-classes.

Threshold	0.9	0.8	0.5
Inter-Class	**96.66**	95.33	95.33
	(23)	(20)	(20)
Intra-Class	**97.33**	**97.33**	96.66
	(25)	(36)	(66)
OAA	**97.33**	95.99	95.99
	(47)	(50)	(67)
Ratio 1	1.01	**1.02**	1.01
Ratio 2	1.01	**1.01**	1.01

TABLE II

Iris Dataset, Unsupervised Discretization (5), linear correlation. A comparison of OAA classification with inter and intra-classes.

Threshold	0.9	0.8	0.5
Inter-Class	**93.99** (38)	91.33 (39)	91.33 (39)
Intra-Class	**93.99** (38)	93.33 (57)	93.33 (122)
OAA	93.99 (76)	**94.00** (93)	86.00 (139)
Ratio 1	1.00	**1.02**	1.02
Ratio 2	1.00	**1.03**	0.94

TABLE III

Iris Dataset, Unsupervised Discretization (7), linear correlation. A comparison of OAA classification with inter and intra-classes.

Threshold	0.9	0.8	0.5
Inter-Class	**90.00**	85.33	85.33
	(49)	(58)	(58)
Intra-Class	0.8999	0.9133	**93.99**
	(47)	(73)	(182)
OAA	90.66	**93.33**	87.33
	(95)	(131)	(207)
Ratio 1	1.00	1.07	**1.10**
Ratio 2	1.01	**1.09**	1.02

From these results, we see that OAA generally leads to better performances than the standard inter-class classification and shows quite similar or better classifications than the intra-class. The better performance of OAA is notably noticed with a higher number of discretization intervals (here with 7 intervals) and lower threshold. As intra-class, OAA performs better with the worst inter-class classification results. OAA helps then to overcome these imperfections (a high number of discretization intervals and low threshold). Table IV shows that OAA performs even better when the Chi Squared test is used. OAA improves classification accuracy in "Iris" until 43% with a higher discretization interval (7).

TABLE IV

Iris Dataset, Chi Squared test. A comparison of OAA classification with inter and intra-classes.

Discretization intervals	3	5	7
Inter-Class	**93.33**	78.66	65.99
	(22)	(47)	(74)

Intra-Class	**97.33**	93.33	89.99
	(25)	(54)	(47)
OAA	**97.33**	94.00	94.66
	(69)	(132)	(187)
Ratio 1	1.04	1.19	**1.36**
Ratio 2	1.04	1.20	**1.43**

The higher number of generated rules is explained by the fact that OAA discovers more and maybe better correlations within the data sets. This could explicain the performances' improvements of our approach.

OAA shows also worse performances (for example within the iris dataset, table II, ratio 2). This could be explained by the error rate caused by a "false negative" component. A false negative (i.e., predicting "no" when the correct label is "yes") is more disastrous than a false positive (i.e., predicting "yes" when the correct label is "no") because a false negative results in 1/c (c: number of classes) probability of correct predictions, while for a false positive this probability is 1/2 [13]. Making an error on $CCS_{\overline{y}i}$ could noticeably decrease the performance of our approach.

Numeric datasets (Iris, 4-0-final_4-2-final and Diabetes) are trained with linear correlation. Mixed and symbolic datasets are tested with the Symmetrical Uncertainty coefficient (except Vote is tested with V-Cramer).

TABLE V
Benchmark results - OAA VS intra-class and inter-class calssification

	Best Scores		
Discretization intervals	3	5	7
Iris			
Inter-Class	96.66	93.99	90.00
Intra-Class	**97.33**	93.99	**93.99**
OAA	**97.33**	**94.00**	93.33
4-0-final_4-2-final			
Inter-Class	**68.29**	70.43	67.91
Intra-Class	**68.29**	**71.53**	**68.26**
OAA	67.92	70.06	**68.25**
Diabetes			
Inter-Class	**65.75**	73.42	71.22
Intra-Class	65.23	**74.34**	71.60
OAA	**65.75**	74.21	**71.73**
Labor			
Inter-Class	76.99	73.66	79.33
Intra-Class	**81.99**	**76.99**	84.33
OAA	78.66	75.33	**86.33**
Colic			
Inter-Class	**67.38**	**74.45**	**75.28**
Intra-Class	**67.38**	74.18	75.01
OAA	**67.38**	74.18	75.01

Dataset	Best scores
Breast-Cancer	
Inter-Class	70.28
Intra-Class	**73.39**
OAA	72.73
Vote	
Inter-Class	92.65
Intra-Class	**94.23**
OAA	91.26

The accuracy of the OAA approach generally exceeds that of the inter-class approach and is comparable with intra-class approach (table V, see appendix). The OAA and intra-class are generally better than the classic approach (inter-class) except with Colic and 4-0-final_4-2-final datasets (with OAA) but with the cost of higher number of rules.

V. Conclusion

We have shown that the One Class Against All Others classification generally performs better than the traditional inter-class approach. A great cardinal of subdivisions (with the regular discretization) and lower thresholds lead generally (or at least for the most cases), for both approaches (intra-class and OAA), to higher performances.

In general, OAA performances are mitigated. A major disadvantage of the OAA approach is that it takes more time in training than the other methods because of the number of treated sub-datasets. Also, because of its error rate, OAA is not the best method [14], [15], [10]. In literature, several improvements are then proposed to surmount this major disadvantage [13], [16]. These works open a way to new perspectives under the framework of discovering dependencies.

Appendix

TABLE VI
4-0-final_4-2-final Dataset, Unsupervised Discretization (7), linear correlation. A comparison of OAA classification with inter and intra-classes.

Threshold	0.9	0.8	0.5
Inter-Class	**67.91**	**67.91**	67.18
	(71)	(71)	(63)
Intra-Class	**0.6826**	67.53	67.91
	(105)	(68)	(86)
OAA	67.53	67.53	**68.25**
	(106)	(67)	(88)
Ratio 1	1.01	0.99	1.01
Ratio 2	0.99	0.99	1.02

TABLE VII
4-0-FINAL_4-2-FINAL DATASET, CHI SQUARED TEST. A COMPARISON OF OAA CLASSIFICATION WITH INTER AND INTRA-CLASSES.

Discretization intervals	3	5	7
Inter-Class	**66.49**	60.26	62.47
	(32)	(83)	(110)
Intra-Class	68.29	**70.43**	68.33
	(36)	(82)	(86)
OAA	68.66	**72.28**	67.96
	(43)	(78)	(90)
Ratio 1	1.03	**1.17**	1.09
Ratio 2	1.03	**1.20**	1.09

TABLE VIII
BREAST-CANCER DATASET, SU CORRELATION. A COMPARISON OF OAA CLASSIFICATION WITH INTER AND INTRA-CLASSES.

Threshold	0.9	0.8	0.5
Inter-Class	69.58	**70.28**	**70.28**
	(78)	(83)	(83)
Intra-Class	0.6958	70.28	**73.39**
	(78)	(90)	(240)
OAA	69.58	69.92	**72.73**
	(78)	(90)	(211)
Ratio 1	1.00	1.00	**1.04**
Ratio 2	1.00	0.99	**1.03**

TABLE IX
VOTE DATASET, V CRAMER CORRELATION. A COMPARISON OF OAA CLASSIFICATION WITH INTER AND INTRA-CLASSES.

Threshold	0.9	0.8	0.5
Inter-Class	91.26	**92.65**	89.21
	(64)	(139)	(192)
Intra-Class	91.26	**94.23**	92.41
	(64)	(160)	(267)
OAA	91.26	**91.71**	88.98
	(64)	(139)	(225)
Ratio 1	1.00	1.02	**1.04**
Ratio 2	1.00	0.99	1.00

TABLE X
LABOR DATASET, UNSUPERVISED DISCRETIZATION (7), SU CORRELATION. A COMPARISON OF OAA CLASSIFICATION WITH INTER AND INTRA-CLASSES.

Threshold	0.9	0.8	0.5
Inter-Class	**79.33**	**79.33**	80.99
	(104)	(104)	(104)
Intra-Class	**79.33**	74.33	84.33
	(104)	(122)	(153)
OAA	**79.33**	78.00	86.33
	(104)	(119)	(149)
Ratio 1	1.00	0.93	**1.04**
Ratio 2	1.00	0.98	**1.07**

REFERENCES

[1] A. Borgi, J.M. Bazin, H. Akdag " Two Methods of Linear Correlation Search for a Knowledge Based Supervised Classification ". *Lecture Notes In Computer Science*, vol. 1415. 1998, pp. 696-707.

[2] S. Di Palma, J. Rugna, and D.A., Zighed, "Apprentissage supervisé polythétique : une évaluation". Presented at the Cinquièmes Rencontres de la Société, SFC'97, Lyon, France, 1997.

[3] A. Jakulin, and I. Bratko, "Testing the Significance of Attribute Interactions" Presented at the 21st International Conference on Machine Learning. Banff, Canada, 2004.

[4] A. Freitas, "Understanding the crucial role of attribute interaction in data mining". *Artificial Intelligence Review*, 2001, pp. 177-199.

[5] G. Saporta, *Probabilités Analyse de données et Statistiques*. 2nd ed, Ed. Technip, Paris, 2006.

[6] D. Lahbib, "Apprentissage supervisé : Optimisation d'une base de règles et recherche de corrélations entre attributs" M.S. thesis. ENSI., Tunisia, 2007.

[7] D.A. Zighed, and R. Rakotomalala, *Graphes d'induction*, HERMES Science Publications, 2000. pp. 327-359.

[8] J. Dougherty, R. Kohavi, and M. Sahami, "Supervised and Unsupervised Discretization of Continuous Features", International Conference on Machine Learning, 1995, pp. 194-202.

[9] U.M. Fayyad, and K.B. Irani, "Multi-Interval Discretization of Continuous-Valued Attributes for Classification Learning". Ed. Bajcsy and Ruzena: Chambéry, France, 1993, pp. 1022--1029. Available: http://www.cs.utsa.edu/\~{}bylander/cs6243/fayyad.pdf.

[10] R. Rifkin, and A. Klautau "In Defense of One-Vs-All Classification", *Journal of Machine Learning Research*, 2004, pp. 101–141.

[11] T. G. Dietterich, and G. Bakiri, "Solving multiclass learning problems via error-correcting output codes", *Journal of Artificial Intelligence Research*, 1995, pp. 263-286.

[12] A. Borgi, D. Lahbib , K. Ghedira, "Optimizing the Number of Rules in a Knowledge Based Classification System ". Presented at AIPR07, Orlando, Florida, USA, July 9-12, 2007.

[13] A. Beygelzimer, J. Langford, and B. Zadrozny, "Weighted One-Against-All", *American Association for Artificial Intelligence*, 2005.

[14] E.L. Allwein, R.E. Schapire, and Y. Singer, "Reducing Multiclass to Binary: A Unifying Approach for Margin Classifiers", *Journal of Machine Learning Research*, vol. 1, 2000, pp. 113-141.

[15] C.-W. Hsu, and C.-J. Lin, "A comparison of methods for multi-class support vector machines", *IEEE transactions on Neural Networks*, vol. 13, 2002, pp. 415-425.

[16] L. Yang, W. Rui, and Y. Zeng, "An Improvement of One-against-all Method for Multi-class Support Vector Machine", in *IEEE Machine Learning and Cybernetics*, Hong Kong, 2007.