Instance Selection in the Performance of Gamma Associative Classifier

Jarvin A. Antón Vargas[1], Yenny Villuendas-Rey[1, 2], Itzamá López-Yáñez[2], Abril V. Uriarte-García[3]

[1] Universidad de Ciego de Avila, Departamento de Ciencias Informáticas, Cuba

[2] Instituto Politécnico Nacional, Centro de Innovación y Desarrollo Tecnológico en Cómputo, Mexico

[3] Instituto Politécnico Nacional, Centro de Investigación en Computación, Mexico

janton@unica.cu, {yenny.villuendas, itzama}@gmail.com, auriarteb10@sagitario.cic.ipn.mx

Abstract. The Gamma associative classifier is among the most used classifiers of the alpha-beta associative approach. It had been used successfully to solve many Pattern Recognition tasks, including environmental applications. However, as most classifiers, Gamma suffers with the presence of noisy or mislabeled instances in the training sets. This paper evaluates the impact of using instance selection techniques in the performance of Gamma classifier. The numerical experiments carried out over well-known repository datasets allows to conclude that instance selection may increase the testing accuracy of the Gamma classifier.

Keywords: Gamma classifier, instance selection, supervised classification.

1 Introduction

The Gamma associative classifier [1, 2] was proposed recently to address supervised classification tasks, including regression. It belongs to the alpha-beta associative approach to Pattern Recognitions, due to its mathematical foundations. Gamma classifier had been used effectively to solve many recognition and prediction tasks, such as the prediction of development effort of software projects [3], estimation of pollutant contamination trough time [4] and determination of air quality in Mexico City [5]. However, as Gamma classifier stores a training set and uses it to assign class labels, it is affected by the presence of noisy or mislabeled instances.

Instance selection algorithms of the error-based (or editing) approach aims at removing the instances considered as outliers or misclassified [6, 7], smoothing

decision boundaries and improving classifier accuracy. The first instance selection algorithm from the editing approach was proposed by Wilson in 1972 and named Edited Nearest Neighbor (ENN) [8]. It consist in the elimination of the instances misclassified by a k-NN classifier. Despite its simplicity, ENN have maintain a competitive performance with respect to recently proposed methods [6].

The GGE algorithm [9] is another well studied instance selection method. It was proposed by Toussaint in 2000 [9]. The GGE algorithm consist in deleting the instances connected to others of different class labels in a Gabriel Graph. It removes frontier instances, and keeps significant ones. A Gabriel graph is a directed graph such that two instances $x \in U$ and $y \in U$ form an arc if and only if $\forall z \in U\ (d((x+y/2),z) > d(x,y)/2)$, where d is a dissimilarity function. That is, two instances x and y are related in a Gabriel graph if there is no object in the hypersphere centered in the middle point of x and y, and with radius the distance between x and y.

We also considered in our study the MSEditB algorithm [10] for instance selection, which is a recently proposed algorithm, also graph-based. MSEditB was proposed in 2009 by García-Borroto et al. [10] and constructs a Maximum Similarity Graph (MSE) to determine the instances to delete.

A Maximum similarity graph is a directed graph such that each instance is connected to its most similar instances. Formally, let be S a similarity function, an instance $x \in U$ form an arc in a Maximum similarity graph with an instance $y \in U$ if and only if $d(x,y) = \max_{z \in U} d(x,z)$.

The MSEditB algorithms removes the instances having a majority of linked instances (successors and predecessors) not belonging to its class.

Most instance selection algorithms are proposed for improving the performance of the Nearest Neighbor (NN) classifier [11]. The approaches to instance selection that considered another classifiers such as Neural Networks [12] and ALVOT [13, 14] are quite specific and there are not appropriate for improving the Gamma classifier.

Instance selection algorithm proposed for Nearest Neighbor classifier [11] need a similarity function to determine neighborhood instances and to construct graph structures. To overcome this problem, Antón-Vargas et al. [15] proposed a novel similarity function based in the foundations of the Gamma classifier. However, that pioneer study does not considered the influence of feature weighting in the Gamma classifier.

This paper includes feature weighting in the performance of the Gamma associative classifier and explores the impact of instance selection in this scenario. The thorough experimental study carried out shows the significant performance gains of the proposed approach.

2 Gamma Classifier

The Gamma associative classifier belong to the alpha-beta approach of associative Pattern Recognition. That is due to it has its foundations on the Alpha and Beta operators of Alpha-Beta associative memories [16]. The Alpha and Beta operators are

defined in a tabular form considering the sets $A = \{0, 1\}$ and $B = \{0, 1, 2\}$, as shown in figure 1.

$\alpha : A \times A \rightarrow B$		
x	y	$\alpha(x, y)$
0	0	1
0	1	0
1	0	2
1	1	1

$\beta : B \times A \rightarrow A$		
x	y	$\beta(x, y)$
0	0	0
0	1	0
1	0	0
1	1	1
2	0	1
2	1	1

Fig. 1. Operators Alpha and Beta.

To perform classification tasks, the Gamma associative classifier incorporates the unary operator u_β and the generalized gamma similarity operator, γ_g, both based on the Alpha and Beta operators. The unary operator u_β receives as an input a binary n-dimensional vector, and returns a number $p \in \mathbb{Z}^+$ according to the following expression:

$$u_\beta = \sum_{i=1}^{n} \beta\,(x_i, x_i), \tag{1}$$

the generalized gamma similarity operator receives as input two binary vectors $\mathbf{x} \in A^n$ y $\mathbf{y} \in A^m$, with $n, m \in \mathbb{Z}^+, n \leq m$, and also a non-negative integer θ, and returns a binary digit, as follows:

$$\gamma_g(x, y, \theta) = \begin{cases} 1 & if\ m - u_\beta[\alpha(x, y) mod 2] \leq \theta \\ 0 & otherwise \end{cases}. \tag{2}$$

That is, the γ_g operator returns 1 if the input vectors differentiates at most in θ bits, and returns zero otherwise.

The Gamma classifier uses the modified Johnson-Möbius code [1] to codify numeric instances, due to the generalized gamma similarity operator receives as input two binary vectors. In the following, we explain the parameters considered in the Gamma classifier.

w.- Is the vector of feature weights, which indicates the relative importance of each variable for the classification process.

initial θ.- Denotes the initial value of θ (typically zero). It indicates the maximum allowed difference between two patterns for the generalized similarity operator.

ρ.- Is the stopping parameter, referred as the maximum value allowed to θ, that permits to continue the search for a unique maximum. When ρ=θ, the Gamma classifier stops the iterations and will assign an arbitrary label. In reference [1] are proposed suggested values for this parameter.

ρ_0.- Is the pause parameter. In the pause, the Gamma classifier performs an evaluation of the pattern to classify, to determine or not its pertinence to the unknown class. In reference [1] are proposed suggested values for this parameter.

d.- Is the variable that is evaluated to decide if the pattern to classify belongs to the unknown class, or if it belongs to any of the known classes.

u.- Is the comparison threshold to determine if the pattern to classify belongs to the unknown class, or if it belongs to any of the known classes. In reference [1] are proposed suggested values for this parameter.

The steps of the functioning of Gamma classifier are shown in Figure 2.

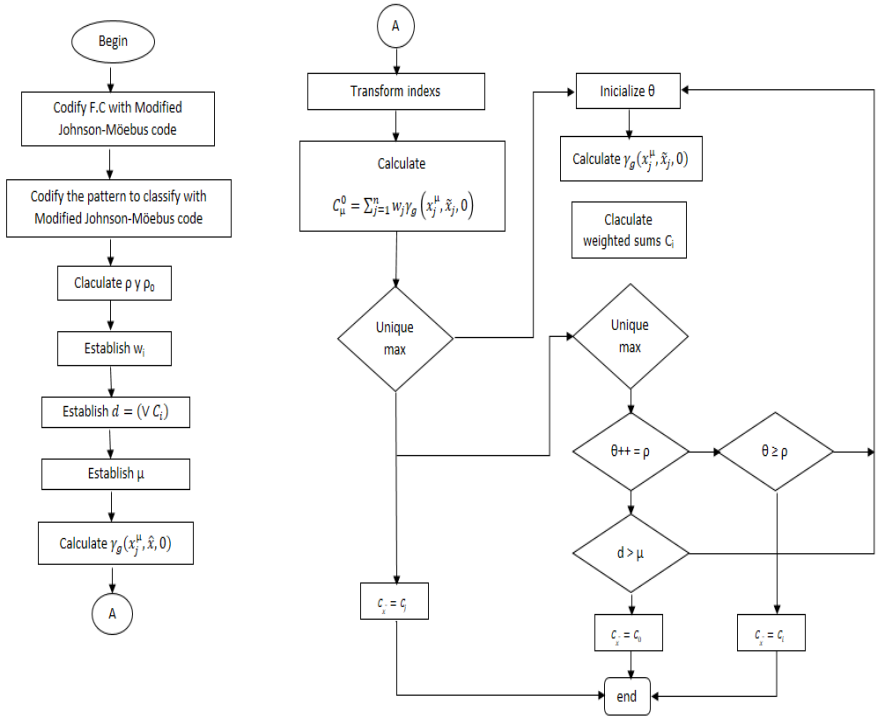

Fig. 2. Schema of the classification process with the Gamma classifier.

To automatically determine the values of the w vector of features weights, Ramírez et al. [17] proposed the use of Differential Evolution metaheuristic. They use a real-valued codification strategy and classifier accuracy over the training set as heuristic evaluation function, to evolve the features weights vector.

According to the classification strategy of the Gamma classifier, Antón-Vargas et al. [15] proposed a similarity function named GBS to compare pairs of instances, regarding the θ parameter.

The Gamma based similarity (GBS) uses the generalized gamma operator, but it considers the standard deviation of the feature instead of the θ parameter. Let be X and Y to instances, the Gamma based similarity between them is computed as [15]:

$$GBS(X,Y) = \sum_{i=1}^{p} \gamma_g\,(x_i, y_i, \sigma_i), \tag{3}$$

where p is the amount of features describing the instances, σ_i is the standard deviation of the i-th feature, and x_i and y_i are the binary vectors associated with the i-th feature in instances X and Y, respectively.

3 Experimental Results

We select some of the most representative instance selection algorithms and perform the test over six databases from the Machine Learning repository of the University of California at Irvine [18]. Table 1 shows the characteristics of the selected databases.

Table 1. Databases used in the experiments.

Databases	Objects	Attributes	Classes
balance-scale	625	4	3
ecoli	336	7	8
heart-statlog	270	13	2
ionosphere	351	34	2
iris	150	4	3
vehicle	846	18	4

We selected error-based editing methods due to their ability of smoothing decision boundaries and to improve classifier accuracy. The selected methods are the Edited Nearest Neighbor (ENN) proposed by Wilson [8], the Gabriel Graph Editing method (GGE) proposed by Toussaint [9] and the MSEditB method, proposed by García-Borroto et al. [10].

For the application of the mentioned instance selection algorithms, we used the Gamma Based Similarity (GBS) function proposed in [15].

We also used the Differential Evolution approach to compute features weights for the Gamma classifier, as proposed in [17].

All algorithms were implemented in C# language, and the experiments were carried out in a laptop with 3.0GB of RAM and Intel Core i5 processor with 2.67HZ. We cannot evaluate the computational time of the algorithms, because the computer was not exclusively dedicated to the execution of the experiments.

To compare the performance of the instance selection algorithms, it was used the classifier accuracy. The classifier accuracy is measure as the ratio of correctly classified instances. It was also computed the Instance retention ratio (IRR) for every algorithm, in order to determine the amount of selected instances. Table 2 and 4 show the results according to classifier accuracy and instance retention ratio, respectively. Best results are highlighted in bold.

In table 2, we show the accuracy of the weighted Gamma classifier without selecting instances (Gamma) and the accuracy of the weighted Gamma classifier trained using the instances selected by ENN, GGE and MSEditB, respectively.

Table 2. Accuracy of the weighted gamma classifier before and after the selection of instances.

Databases	Gamma	Instances selected by		
		ENN	GGE	MSEditB
balance-scale	0.760	0.685	0.770	**0.811**
ecoli	0.486	**0.504**	0.447	0.439
heart-statlog	0.837	0.837	0.833	**0.844**
ionosphere	**0.761**	0.641	0.359	0.000*
iris	**0.933**	**0.933**	**0.933**	0.927
vehicle	**0.584**	0.579	0.551	0.564

* the MSEditB method deletes all the instances.

As shown, the instance selection algorithms were able to improve the Gamma classifier accuracy in two of the compared databases, and to obtain the same accuracy with fewer instances in one database. Still, for the ionosphere and vehicle datasets, no improvement were obtained.

In addition, it is important to mention that for the ionosphere dataset, the MSEditB algorithm delete all the instances, considering that the entire dataset was mislabeled or noisy.

However, to determine the existence or not of significant differences in algorithm´s performance it was used the Wilcoxon test [19]. It was set as null hypothesis no difference in performance between the gamma classifier without instance selection (Gamma) and the gamma classifier with instance selection algorithms, and as alternative hypothesis that latter had better performance. It was set a significant value of 0.05, for a 95% of confidence. Table 3 summarizes the results of the Wilcoxon test, according to classifier accuracy.

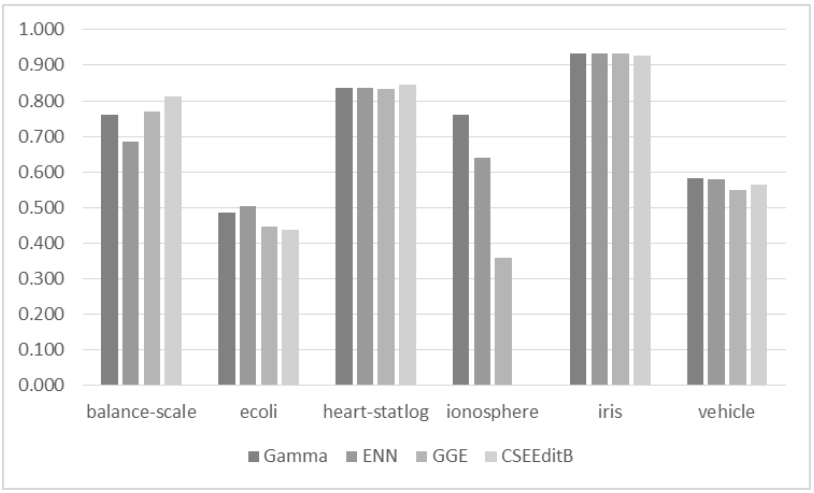

Fig. 3. Accuracy of the Gamma classifier using selected instances.

Table 3. Wilcoxon test comparing classifier accuracy.

Original Gamma vs	ENN	GGE	MSEditB
wins-looses-ties	3-1-2	4-1-1	4-2-0
probability	0.273	0.138	0.463

The Wilcoxon test obtains probability values greater than the significance level, and thus, we do not reject the null hypothesis. These results confirm the instance selection approach is able to preserve classifier accuracy, using a small amount of instances.

Table 4. Instance retention ratio obtained by the selection of instances.

Databases	ENN	GGE	MSEEditB
balance-scale	0.912	0.847	**0.777**
ecoli	0.844	0.888	**0.675**
heart-statlog	0.887	0.772	**0.763**
ionosphere	0.641	**0.359**	0.000*
iris	0.955	0.973	**0.934**
vehicle	0.838	0.851	**0.675**

*the MSEditB algorithm deletes all the instances.

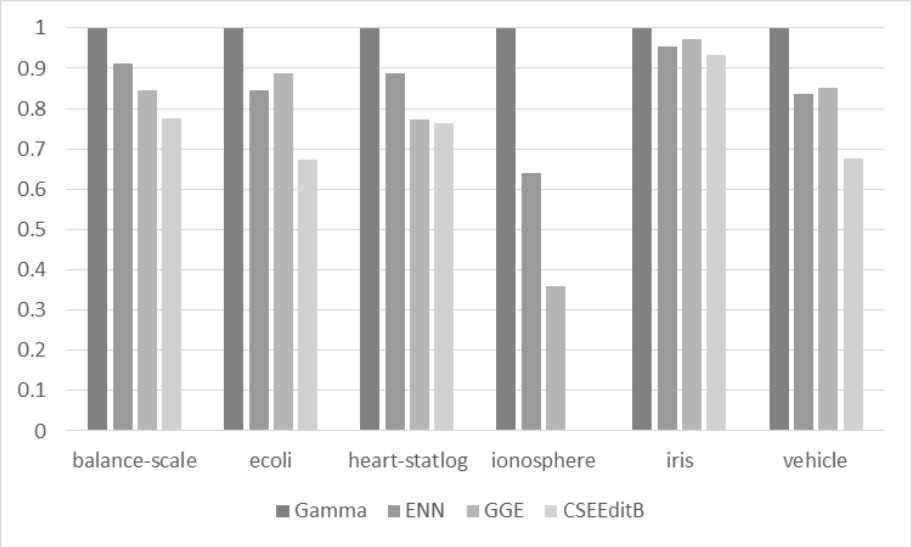

Fig. 4. Instance retention ratio obtained by the algorithms.

As shown in table 4, all instance selection methods are able to delete among the 60% and 4% of the data, without decreasing the classifier accuracy. These results confirm the proposed approach is able to obtain an adequate training set for the Gamma classifier, without losing representative objects.

Table 5. Wilcoxon test comparing instance retention ratio.

Original Gamma vs	ENN	GGE	MSEditB
wins-looses-ties	0-6-0	0-6-0	0-6-0
probability	**0.028**	**0.028**	**0.027**

According to instance retention ratio, the Wilcoxon test rejects the null hypothesis in all cases. That is, the number of selected objects using ENN, GGE and MSEditB with the proposed gamma based similarity function, was significantly lower than the original amount of instances in the training set.

The experimental results carried out in our research show that using automatic weigh computation, as well as a similarity function based on the Gamma operator, allows to successfully apply instance selection algorithms to improve the performance of the Gamma associative classifier. The statistical tests show that instance selection algorithms are able to maintain classifier accuracy, and also to reduce the cardinality of the training sets, diminishing the computational cost of the Gamma classifier.

4 Conclusions

In this paper is explored the impact of instance selection algorithms in conjunction with automatic feature weight in the performance of the Gamma associative classifier. The numerical experiments were carried out over well- known repository data. The obtained results confirm the hypothesis that instance selection algorithms may decrease the computational cost of the Gamma classifier, while preserve the classifier accuracy. In addition, the study conclude that automatic feature weighting procedures may increase the performance of the Gamma classifier.

References

1. López Yáñez, I.: Clasificador automático de alto desempeño. MS dissertation, Instituto Politécnico Nacional-Centro de Investigación en Computación (2007)
2. López-Yáñez, I., Sheremetov, L., Yáñez-Márquez, C.: A novel associative model for time series data mining. Pattern recognition Letters, 41, pp. 23–33 (2014)
3. López-Martin, C., López-Yánez, I., Yánez-Márquez, C.: Application of Gamma Classifier to Development Effort Prediction of Software Projects. Appl. Math., 6, pp. 411–418 (2012)
4. Lopez-Yanez, I., Argüelles-Cruz, A.J., Camacho-Nieto, O., Yanez-Marquez, C.: Pollutants time-series prediction using the Gamma classifier. International Journal of Computational Intelligence Systems, 4, 680–711 (2012)
5. Yánez-Márquez, C., López-Yánez, I. Morales, G.D.: Analysis and prediction of air quality data with the gamma classifier. Progress in Pattern Recognition, Image Analysis and Applications, pp. 651–658 (2008)
6. García, S., Derrac, J., Cano, J.R., Herrera, F.: Prototype Selection for Nearest Neighbor Classification: Taxonomy and Empirical Study. IEEE Transactions on Pattern Analysis and Machine Intelligence, 34, pp. 417–435 (2012)

7. Triguero, I., Derrac, J., Garcia, S., Herrera, F.: A taxonomy and experimental study on prototype generation for nearest neighbor classification. Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions, 42, pp. 86–100 (2012)
8. Wilson, D.L.: Asymptotic properties of nearest neighbor rules using edited data, IEEE Transactions on Systems, Man and Cybernetics, SMC-2, pp. 408–421 (1972)
9. Toussaint, G.T.: Proximity Graphs for Nearest Neighbor Decision Rules: Recent Progress. 34 Symposium on Computing and Statistics INTERFACE-2002, Montreal, 1–20, Canada (2002)
10. García-Borroto, M., Villuendas-Rey, Y., Carrasco-Ochoa, J.A., Martinez-Trinidad, J.F.: Using Maximum Similarity Graphs to edit nearest neighbor classifiers. Lecture Notes on Computer Science, 5856, 489–496 (2009)
11. Cover, T.M., Hart, P.E.: Nearest Neighbor pattern classification. IEEE Transactions on Information Theory, 13, 21–27 (1967)
12. Ishibuchi, H., Nakashima, T., Nii, M.: Learning of neural networks with GA-based instance selection. IFSA World Congress and 20th NAFIPS International Conference, 4, pp. 2102–2107 (2001)
13. Medina-Pérez, M.A., García-Borroto, M., Villuendas-Rey, Y., Ruiz-Shulcloper, J.: Selecting objects for ALVOT. Progress in Pattern Recognition, Image Analysis and Applications, pp. 606–613 (2006)
14. Medina-Pérez, M.A., García-Borroto, M., Ruiz-Shulcloper, J.: Object selection based on subclass error correcting for ALVOT. Progress in Pattern Recognition, Image Analysis and Applications, pp. 496–505 (2007)
15. Antón-Vargas, J.A., Villuendas-Rey, Y., López-Yáñez, I.: Gamma classifier based instance selection. Research in Computer Science (Accepted paper) (2015)
16. Yáñez-Márquez, C., Díaz, L.: Memorias Asociativas basadas en relaciones de orden y operaciones binarias. Computación y Sistemas, Vol. 6, pp. 300–311 (2003)
17. Ramirez, A., Lopez, I., Villuendas, Y., Yanez, C.: Evolutive Improvement of Parameters in an Associative Classifier. Latin America Transactions, IEEE (Revista IEEE America Latina), Vol. 13, pp. 1550–1555 (2015)
18. Asuncion, A., Newman, D.: UCI machine learning repository (2007)
19. Demsar, J.: Statistical comparison of classifiers over multiple datasets. The Journal of Machine Learning Research, Vol. 7, pp. 1–30 (2006)
20. Caballero, Y., Bello, R., Salgado, Y., García, M.M.: A method to edit training set based on rough sets. International Journal of Computational Intelligence Research, Vol. 3, pp. 219–229 (2007)