ISSN 2395-8618

Instance Selection to Improve Gamma Classifier

Jarvin A. Antón-Vargas, Yenny Villuendas-Rey, and Itzamá López-Yáñez

Abstract—**Pre-processing the dataset is an important stage in the Knowledge Discovery in Datasets (KDD) process. Filtering noise through instance selection is a necessary task. With this, the risk to use misclassified and non-representative instances to train supervised classifiers is reduced. This study aims at improving the performance of the Gamma associative classifier, by introducing a novel similarity function to guide instance selection. The experimental results, over 15 datasets, include several instance selection methods, and their influence in the performance of Gamma classifier is analyzed. The effectiveness of the proposed similarity function is tested, obtaining good results according to classifier accuracy and instance retention ratio.**

Index Terms—**Gamma classifier, instance selection, data pre-processing, similarity functions.**

I. Introduction

The training dataset plays a key role for supervised classification. Training data allows building classifiers able to estimate the label or class of a new unseeing instance. Several researchers have pointed out that if the dataset has high quality instances, the classifier can produce predictions that are more accurate [1]. However, in several real-world applications, it is not possible to obtain a training set without noisy and mislabeled instances. To overcome this problem, several algorithms for instance selection and construction have been proposed [1] [2].

The Gamma classifier [3] [4] is a recently proposed supervised classifier, and it has been applied successfully to several prediction tasks, such as air quality monitoring [5], pollutant prediction [6] and development effort prediction of software projects [7]. Despite of the excellent performance of the Gamma classifier, it is noted that it is affected by noisy or mislabeled instances.

Most of the instance selection algorithms are designed for the Nearest Neighbor (NN) classifier [8]. Little work has been done for selecting instance for other supervised classifiers, such as ALVOT [9] [10] and Neural Networks [11], and these proposals are no directly applicable to the Gamma classifier.

This paper proposes a novel similarity function based on the Gamma operator of the Gamma classifier, and use it for similarity comparisons in NN instance selection algorithms. The thorough experimental study carried out shows the significant performance gains of the proposed approach.

With the advance of digital technology, the technological advances of computers, the continued growth of the computerization of society, and the development of the web it has facilitated the easy accumulation of data (business data, websites, data warehouses, etc.) and information [12]. This phenomenon has referred by some authors as “drowning in information” [13], because working with them is tedious and involves a high computational cost [14]. As example the data collected by institutions such as the particle collider in Switzerland CERN or data obtained in sciences like astronomy and biology in studying the human genome and protein sequencing [15] [16]. Any of these fields of study can work with data in the order of petabytes. Researchers daily have to face this problem, mainly to the analysis of databases with a large dimensionality, we must understand large dimensionality as numerous features and a high number of instances.

It is common for researchers when they use values from real problems, have to deal in many cases with data represented by many characteristics and only a few of them are directly related to the objective of the problem. Redundancy may exist where several features can have a high correlation, which makes it not necessary to include them all in the final model. Find interdependence also applies, so that two or more features contains relevant information, and if is excluded any of them, it can make this information useless [17].

Another important problem arises when the training set is excessively large relative to the number of instances, making impracticable the supervised learning. By other hand if the classification is practicable, to cite one example; when it contains class imbalance problems, in most of the time the algorithms opt for the majority class and include in that category objects of minority classes.

It is normal that there are also instances that do not contain relevant information or are not significant for the classification of the problem in question. Once the preprocessing is performed through the selection of instances, the model could predict on the basis of a training set, adjusted to the most representative elements of the problem in question and in turn reducing the time execution, this are essential elements to arrive at a desirable outcome [18].

Manuscript received on December 14, 2015, accepted for publication on April 23, 2016, published on October 30, 2016.

J. A. Antón-Vargas is with the Computer Science Department of the University of Ciego de Ávila, Cuba. (e-mail: janton@ unica.cu).

Y. Villuendas-Rey (corresponding author) is with the CIDETEC, Instituto Politécnico Nacional, Mexico City, Mexico (e-mail: yvilluendasr@ipn.mx).

I. López-Yáñez is with the CIDETEC, Instituto Politécnico Nacional, Mexico City, Mexico (e-mail: ilopezy@ipn.mx).

https://doi.org/10.17562/PB-54-9

II. Previous works

As is known in the supervised classification process, the training is an important phase. Such learning is guided by datasets containing training cases. It is usual in real life problems, that this training sets contain vain information for the classification process; understood by this superfluous cases, which may contain noise or may be redundant [18]. That is why removing these cases from the initial training set is needed.

Given a training set, the aim of the instance selection methods is to obtain a subset that does not contain superfluous instances, so that the accuracy of the classification obtained using the resulting subset of instances is not degraded. These methods can generate subsets incrementally, adding instances as the knowledge space is explored. Another alternative is to start from the initial set of instances and thus eliminating instances to find the optimal subset according to the algorithm used. Through the selection of instances, the training set is reduced, which could be useful in reducing the time during the learning process, particularly in based instance models where the classification of a new instance uses the entire training set.

Several models have been proposed in the literature to address this task, obtaining good results considering the main objective of this preprocessing technique. In the next sections, we will discuss different models, observing the diversity of approaches and the elements considered by the researchers in order to improve the supervised classification model.

A. *Instance selection for Nearest Neighbor classifier*

Most of the instance selection methods proposed are based on the NN classifier. CNN (Condensed Nearest Neighbor) [19] has been one of the first, this method follows the incremental model and its initial routine randomly include in the result set S, an instance belonging to each of the class problem. Then each instance of the original set is classified using as training set S; if the instance is classified incorrectly, then it is included in the set S pursuing the idea that if there is another instance like this then be classified correctly. One drawback of this model is that it could hold instances that constitute noise in the result set.

Since this method is derived a set of methods among which are the SNN (Selective Nearest Neighbor) [20], which generates the set S following the criterion that each member of the original set is closer to a member of the result set S than any other, this could be understood as each instance would be correctly classified by the NN classifier using as training set to S. Another variation within this group is the GCNN (Generalized Condensed Nearest Neighbor) [21], its operation is identical to CNN, and this just includes an absorption criterion according to a threshold. This means that for each instance, the absorption is calculated in terms of the nearest neighbors and closest enemies (those closest instances to a member of a class but belonging to another class).

Another method for selecting instances is the ENN (Nearest Neighbor Edited) [22] that focuses on discard the noisy instances present in the training set. This method discards those instances when the class is different from the majority class of their closest k neighbors (ENN generally used k = 3). An extension of the ENN is the RENN (Repeated ENN), this method applies ENN repeatedly until all instances present in the resulting set S belong to the same class as the class that belongs the majority of their k nearest neighbors. Another variant is the All-KNN [23], this method works iterating the routine k-NN algorithm k times, labeling the instances that are misclassified. Once the iterations are stopped, all labeled instances are discarded from the training set.

B. *Instance selection for Artificial Neural Networks*

It is well known that Artificial Neural Networks (ANNs) can produce robust performance when a large amount of data is available. However, the noisy data may produce inconsistent and unpredictable performance in ANN's classification. In addition, it may not be possible to train ANN or the training task cannot be effectively carried out without data reduction when a data set is too huge.

In the literature, we can find many researches trying to obtain the best training set for this powerful technique. In [24] the authors propose a new hybrid model of ANN and Genetic Algorithms (GAs) for instance selection. An evolutionary instance selection algorithm reduces the dimensionality of data and eliminate noisy and irrelevant instances. In addition, this study simultaneously searches the connection weights between layers in ANN through an evolutionary search. By the other hand the genetically evolved connection weights mitigate the well-known limitations of gradient descent algorithm.

In the same way to simplify the space dimension of input information and reduce the complexity of network structure, the information entropy reduction theory is brought in. Trying to aim at the main shortage of ANN (the converging speed is often slow and the network is easily involved in local optimum), is introduced the Particle Swarm Optimization (PSO) [25].

C. *Instance selection for other classifiers*

In the Logical Combinatorial approach to Pattern Recognition (LCPR), ALVOT is a model for supervised classification. This model was inspired in the works by Zhuravlev, and it is based on partial precedence. Let understand as partial precedence, the principle of calculating the similarity between objects using comparisons between their partial descriptions. A partial description is a combination of features. This is the way that many scientists such as physicians, and other natural scientists, establish comparisons among objects in real world problems [26].

When a new instance is classified, many partial comparisons with all the objects in the training set have to be calculated. This can be very time consuming, while the cardinality of the set increases. That is why an instance selection method for ALVOT was introduced in [27] [28] with good results. In both

ISSN 2395-8618

algorithms, the authors introduce a voting strategy to select the most relevant instances.

In addition, Genetic Algorithms have been used for instance selection in the context of ALVOT classifier [9]. The algorithm presented in [9] start generating randomly the initial population. The input parameters for the algorithm are the population size and iteration number. Then the population's individuals are sorted according to their fitness. The first and last individuals are crossed, the second is crossed with the penultimate and this process is repeated until finishing the population. They are crossed using a 1-point crossover operator in the middle of the individual. The fitness function is the ratio of well classified objects. The mutation operator is evaluated for each individual in the population changing randomly the values of an individual's gene. Then the fitness is evaluated for this new population. The original individuals together with those obtained by crossing and mutation are sorted in descending order according to their fitness and those with highest fitness are chosen (taking into account the population size). The new population is used in the next iteration of the algorithm.

To this classifier is possible apply others instances selections methods, such as the classical models based on NN rule. An analogue solution was reported by Decaestecker [29] and Konig et al. [30], in which the training set is edited for a Radial Based Function network, using a procedure originally designed for NN.

Because of the importance of data preprocessing for any classifier, it is interesting to note that for associative classifiers, such as Gamma, to the best of our knowledge, there are no analysis of the impact of instance selection in classifiers performance. Considering that this approach generates a memory of fundamental patterns, and associates instances with their respective classes, we hypothesize that this association process may provide better results if this memory is created from refined and representative instances of the problem to solve.

III. Gamma Classifier

The Gamma associative classifier belongs to the associative approach of Pattern Recognition, created in the National Polytechnic Institute of Mexico [31]. The Gamma classifier is based on two operators named Alpha and Beta, which are the foundation of the Alpha-Beta associative memories [32]. 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 Fig. 1.

In addition to the Alpha and Beta operator, the Gamma classifier also uses two other operators: the u_β operator and the generalized gamma similarity operator, γ_g. 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 $x \in A^n$ and $y \in A^m$ with $n, m \in \mathbb{Z}^+, n \leq m$, and also a non-negative integer θ, and returns a binary digit, as follows:

$$\gamma_g(x, y, \theta) = \begin{cases} 1 & if\ m - u_\beta[\alpha(x, y) mod2] \leq \theta \\ 0 & otherwise \end{cases} \tag{2}$$

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

The Gamma classifier is designed for numeric patterns, and assumes that each pattern belongs to a single class. However, as the generalized gamma similarity operator receives as input two binary vectors, the Gamma classifier codifies numeric instances using a modified Johnson-Möbius code [3]. In Figure 2 we show a simplified schema of the Gamma classifier.

$\alpha : A \times A \rightarrow B$		
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. Tabular definition of Alpha and Beta operators.

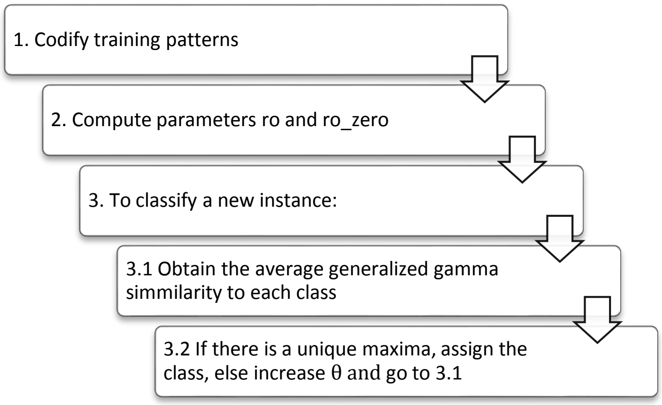
1. Codify training patterns
2. Compute parameters ro and ro_zero
3. To classify a new instance:
3.1 Obtain the average generalized gamma simmilarity to each class
3.2 If there is a unique maxima, assign the class, else increase θ and go to 3.1

Fig. 2. Simplified schema of the classification process with the Gamma classifier.

The training process of Gamma uses the modified Johnson Möbius code to codify the instances in sets of bits, this allows to obtain the values of ρ and ρ_0, necessaries parameters to the next phase. This values are determined finding the lowest of all e_m values which are the greatest of all values for each feature, as it shows in the next formulas:

$$\rho = \bigwedge_{i=1}^{p} e_m(j) \tag{3}$$

$$e_m(j) = \bigvee_{i=1}^{p} x_i^j \tag{4}$$

In the classification phase, the new instance is codified with the same Johnson-Möbius code, thus a θ parameter its initialized with zero value, then all class in the training dataset are grouped by class and its calculated the $\gamma_g(x, y, \theta)$ between the new instance and all instances for each kind of class. The final classification is assigned from the class with greatest similarity value calculated by:

$$C_i = \frac{\sum_{i=1}^{n} \gamma_g(x, y, \theta)}{n} \tag{5}$$

If the value of C_i is not unique, then θ increment its value relaxing the similarities between the instances and all the process is done again, until the θ gets a value equal to ρ. In the case that at the end of all iterations there's not a unique maximum for a class, then it is assigned the first class with maximum similarity.

A detailed characterization of the Gamma classifier can be found in [33]. Its use has been extended into datasets with different characteristics and with different objectives; mainly in classification tasks, for which it was designed. It also has been used in interpolation tasks and functions exploration, these last ones for which it was not designed. It was determined that good results can be expected when data induce a function. In other words, if each input element has a single output pattern or class, then the classifier is competitive and even superior to other algorithms. Otherwise, when an input pattern has various output patterns then the algorithm is in a situation not expected by the original algorithm.

The algorithm has a competitive performance also in the case of a known sequence is present, which output will be the known value that best matches with this sequence. However, it is clear to suppose that it may happen that this sequence is not exactly known, but it is near the border between two or more known sequences. Then, an output near the border between the known corresponding outputs is obtained.

IV. GAMMA BASED SIMILARITY

According to the classification strategy of the Gamma classifier, we propose a similarity function to compare pairs of instances, regarding the θ parameter. This allows us to detect noisy or mislabeled instances.

The proposed 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:

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

where p is the number 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.

Considering this novel similarity, we are able to apply several instance selection algorithms which were designed for the NN classifier, and test their performance in the filtering of noisy and mislabeled instances for the Gamma classifier.

As shown, in the instance selection methods described in previous sections, the similarity between instances is critical to the operation of any method that seeks to select or discard instances of a training set. In the case of Gamma associative classifier, a new similarity function it is proposed, based mainly on the criteria to take a decision over the values of the features of each instance. The original similarity, works in dependence on a θ, value that is dynamically updated, this dynamic value allows a relaxation of the classifier which is not a good criterion for selection instances model.

That is why the use of standard deviation for the θ value is proposed. The criteria taken into account for the adoption of this variant are the benefits that has the standard deviation over a set of values, in this case would be the values of each feature. First, keep in mind that the standard deviation is by far the measure generally used to analyze the variation in a group of values. It is a measure of absolute variation [34] that calculate the real amount of variation present in a dataset. This allows to know more about the dataset of interest. It is not enough to know the measures of central tendency, we also need to know the deviation present in the data with regard to the average of these values. Its calculation is determined by the following formula:

$$s = \sqrt{\frac{\sum_{i=1}^{n}(x_i - \tilde{x})^2}{n-1}} \tag{6}$$

where $\tilde{x}$ is the arithmetic mean.

This allow us to know for each feature its dispersion and get a better fit of the decision criteria that define if two values are similar or not.

V. EXPERIMENTAL RESULTS

To test the influence of instance selection algorithms in the performance of the Gamma classifier, we use some of the most representative instance selection algorithms reported in the literature, and apply them over well-known datasets from the Machine Learning repository of the University of California at Irvine [13]. Table 1 shows the characteristics of the selected datasets.

ISSN 2395-8618

TABLE I
DATASETS USED IN THE NUMERICAL EXPERIMENTS

Datasets	Instances	Attributes	Classes
balance-scale	625	4	3
diabetes	768	8	2
ecoli	336	7	8
hayes-roth	160	4	3
heart-statlog	270	13	2
ionosphere	351	34	2
iris	150	4	3
liver-disorders	345	6	2
mfeat-morphological	2000	6	10
new-thyroid	215	5	3
page-blocks	5473	10	5
pendigits	10992	16	10
spambase	4601	57	2
vehicle	846	18	4
wine	154	13	3

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 [14], the Gabriel Graph Editing method (GGE) and Relative Neighborhood Graphs (RNGE) proposed by Toussaint [15] and the MSEditB method, proposed by García-Borroto et al. [16].

The ENN algorithm (Edited Nearest Neighbor) is the first error-based editing method reported [14]. It was proposed by Wilson in 1972 and it consist on the elimination of the objects misclassified by a 3-NN classifier. The ENN works by lots, because it flags the misclassified instances and then simultaneously deletes them all, which guaranteed order independence. The ENN has been extensively used in experimental comparisons, showing very good performance [1].

The GGE algorithm is based on the construction of a Gabriel graph. 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.

The GGE algorithm consists in deleting those instances connected to others of different class labels. It deletes borderline instances, and keep class representative ones.

Similar to GGE, the RNGE [15] uses a Relative Neighborhood graph to determine which instance delete. A Relative Neighborhood graph is a directed graph such that two instances $x \in U$ and $y \in U$ form an arc if and only if $\forall z \in U\ (d(x, y) < max\{d(x, z), d(y, z)\})$, where d is a dissimilarity function.

The MSEditB algorithm [16] uses a Maximum similarity graph to select the objects 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 algorithm deletes an instance if it has a majority of its predecessors and successors instances not of its class.

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 algorithms, it was used the classifier accuracy. The classifier accuracy is measure as the percent of correctly classified instances. Let be X the testing set, $l(x)$ the true label of instance $x \in X$, and $d(x)$ the decision made by the classifier. The classifier accuracy is defined as:

$$cc(X) = \frac{|\{x \in X : l(x) = d(x)\}|}{|X|} * 100 \qquad (7)$$

It was also computed the Instance Retention Ratio (IRR) for every algorithm, in order to determine the number of selected instances. The IRR is measured as the ratio of instances that are kept by the instance selection algorithm. Let be T the set of training instances, and $E \subseteq T$ the set of instances selected. The IRR is computed as:

$$IRR = \frac{|E|}{|T|} \qquad (8)$$

In Table 2, we show the accuracy of the Gamma classifier without selecting instances (Gamma) and the accuracy of the Gamma classifier trained using the instances selected by ENN, GGE, RNGE and MSEditB, respectively. Results corresponding to accuracy improvements of Gamma classifier are highlighted in bold.

TABLE II
CLASSIFIER ACCURACY AFTER SELECTING INSTANCES

Datasets	Gamma	ENN	GGE	RNGE	MSEditB
balance-scale	83.845	**74.071**	83.372	**90.719**	**90.241**
diabetes	59.516	**61.471**	58.083	**61.470**	**60.947**
ecoli	50.980	**52.433**	46.827	50.089	48.966
hayes-roth	74.375	71.250	65.625	68.750	**79.375**
heart-statlog	81.852	81.852	**82.593**	**82.222**	**82.963**
ionosphere	**74.373**	64.111	35.889	*	*
iris	88.667	**90.000**	**90.000**	**90.000**	**90.000**
liver-disorders	**57.697**	55.059	56.546	54.504	55.387
mfeat-morphological	46.000	43.500	**46.650**	41.850	41.650
new-thyroid	80.476	**81.861**	80.931	**81.407**	**82.316**
page-blocks	77.160	**77.873**	77.105	76.557	**77.471**
pendigits	64.456	**64.483**	64.465	**64.483**	**64.547**
spambase	71.310	70.441	**75.287**	73.918	70.202
vehicle	**59.592**	58.641	56.146	57.817	58.053
wine	72.451	71.863	70.294	**73.007**	72.451
Increases of classifier accuracy		7	4	**7**	8

* The RNGE and MSEditB algorithms select no instance.

ISSN 2395-8618

In 12 of the tested datasets, the instance selection algorithms were able to improve the accuracy of the Gamma classifier. However, in datasets ionosphere, liver-disorders and vehicle no improvement was achieved. In particular, the ionosphere dataset shows a high degree of class overlapping, such that both RNGE and MSEditB algorithms do not kept any instance.

Despite this pathological behavior, the instance selection algorithms exhibit a very good performance, with several improvements in classifier accuracy.

In Table 3, we show the Instance Retention Rate (IRR) obtained by ENN, GGE, RNGE and MSEditB, respectively. Best results are highlighted in bold.

TABLE III
INSTANCE RETENTION RATIO OBTAINED BY THE ALGORITHMS

DATASETS	ENN	GGE	RNGE	MSEDITB
balance-scale	0.912	0.847	0.895	**0.777**
diabetes	0.847	**0.669**	0.770	**0.669**
ecoli	0.844	0.888	0.754	**0.675**
hayes-roth	0.866	0.647	0.278	**0.645**
heart-statlog	0.887	0.772	0.852	**0.763**
ionosphere	0.641	**0.359**	*	*
iris	0.955	0.973	0.943	**0.934**
liver-disorders	0.838	**0.537**	0.739	0.580
mfeat-morphological	0.826	0.936	0.715	**0.639**
new-thyroid	0.982	0.967	0.980	**0.957**
page-blocks	0.979	**0.958**	0.963	**0.958**
pendigits	0.997	0.997	0.995	**0.992**
spambase	0.952	**0.856**	0.870	0.902
vehicle	0.838	0.851	0.777	**0.675**
wine	0.986	**0.889**	0.988	0.951

* The RNGE and MSEditB algorithms select no instance.

Both GGE and MSEDitB were the algorithms with best results according to IRR. GGE obtained IRR varying from 0.35 to 0.95, and MSEditB from 0.63 to 0.992. ENN and RNGE obtained inferior results.

However, to determine the existence or not of significant differences in algorithm´s performance it was used the Wilcoxon test [17].

The Wilcoxon test is a non-parametric test recommended to statistically compare the performance of supervised classifiers. In the test, we set as null hypothesis that there is no difference in performance between the gamma classifier without instance selection (Gamma) and the gamma classifier after instance selection, and as alternative hypothesis that instance selection algorithms lead to better performance. We set a significant value of 0.05, for a 95% of confidence.

Tables 4 and 5 summarize the results of the Wilcoxon test, according to classifier accuracy and instance retention rate, respectively.

The Wilcoxon test obtains probability values greater than the significance level, and thus, we do not reject the null hypothesis. These results confirm that instance selection algorithms using the proposed similarity function are able to preserve classifier accuracy, using a small number of instances.

TABLE IV
WILCOXON'S TEST COMPARING CLASSIFIER ACCURACY

Gamma vs.	w-l-t	Z	Probability	Decision
ENN	6-8-1	-1.420	0.245	No reject
GGE	6-9-0	-1.420	0.156	No reject
RNGE	8-7-0	-0.454	0.650	No reject
MSEditB	8-6-1	-0.094	0.925	No reject

TABLE V
WILCOXON'S TEST COMPARING CLASSIFIER ACCURACY

Gamma vs.	w-l-t	Z	Probability	Decision
ENN	15-0-0	-3.408	0.001	Reject
GGE	15-0-0	-3.408	0.001	Reject
RNGE	15-0-0	-3.408	0.001	Reject
MSEditB	15-0-0	-3.408	0.001	Reject

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, RNGE and MSEditB with the proposed gamma based similarity function, was significantly lower than the original number of instances in the training set.

The experimental results carried out show that selecting instances by using a similarity function based on the Gamma operator maintains classifier accuracy, and also reduces the cardinality of the training sets, diminishing the computational cost of the Gamma classifier.

VI. CONCLUSION

We considered that instance selection process based on the proposed similarity function contributes to the improvement of the Gamma associative classifier by maintaining its performance with low computational complexity. As future work, we plan to experiment with the feature weight assignment process, in order to further improve the Gamma classifier.

REFERENCES

[1] S. García, J. Derrac, J. R. Cano and F. Herrera, "Prototype Selection for Nearest Neighbor Classification: Taxonomy and Empirical Study," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 34, pp. 417-435, 2012.

[2] I. Triguero, J. Derrac, S. Garcia and F. Herrera, "A taxonomy and experimental study on prototype generation for nearest neighbor classification," *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions,* vol. 42, pp. 86-100, 2012.

[3] I. López-Yáñez, "Clasificador automático de alto desempeño (MS dissertation)," 2007.

[4] I. López-Yáñez, L. Sheremetov and C. Yáñez-Márquez, "A novel associative model for time series data mining," *Pattern recognition Letters,* vol. 41, pp. 23-33, 2014.

[5] C. Yánez-Márquez, I. López-Yánez and G. Morales, "Analysis and prediction of air quality data with the gamma classifier," *Progress in Pattern Recognition, Image Analysis and Applications,* pp. 651-658, 2008.

[6] I. Lopez-Yanez, A. J. Argüelles-Cruz, O. Camacho-Nieto and C. Yanez-Marquez, "Pollutants time-series prediction using the Gamma classifier,"

International Journal of Computational Intelligence Systems, nº 4, pp. 680-711, 2012.

[7] C. López-Martın, I. López-Yánez and C. Yánez-Márquez, “Application of Gamma Classifier to Development Effort Prediction of Software Projects,” *Appl. Math,* vol. 6, nº 3, pp. 411-418, 2012.

[8] T. M. Hart and P. E. Cover, “Nearest Neighbor pattern classification,” *IEEE Transactions on Information Theory,* vol. 13, pp. 21-27, 1967.

[9] M. A. Medina-Pérez, M. García-Borroto, Y. Villuendas-Rey and J. Ruiz-Shulcloper, “Selecting objects for ALVOT,” *Progress in Pattern Recognition, Image Analysis and Applications,* pp. 606-613, 2006.

[10] M. A. Medina-Pérez, M. García-Borroto and J. Ruiz-Shulcloper, “Object selection based on subclass error correcting for ALVOT,” *Progress in Pattern Recognition, Image Analysis and Applications,* pp. 496-505, 2007.

[11] H. Ishibuchi, T. Nakashima and M. Nii, “Learning of neural networks with GA-based instance selection,” *IFSA World Congress and 20th NAFIPS International Conference,* vol. 4, pp. 2102-2107, 2001.

[12] H. M. HUAN-LIU, “On Issues of Instance Selection,” *Data Mining and Knowledge Discovery,* vol. 6, pp. 115-130, 2002.

[13] A. Szalay, “Drowning in data,” *Scientific American.,* 1999.

[14] B. Z. Jun-Yan, “Effective and Efficient Dimensionality Reduction for Large-Scale and Streaming Data Preprocessing,” *IEEE Transactions on Knowledge and Data Engineering,* vol. 18, 2006.

[15] L. Y. Liu, “Efficient feature selection via analysis of relevance and redundancy,” *Journal Machine Learning Research,* vol. 5, pp. 1205-1224, 2004.

[16] J. A. Antón-Vargas and C. Santiesteban-Toca, “Selección de algoritmos para la predicción de contactos interresiduales de proteínas,” *9no Congreso Internacional Biotecnología Vegetal,* 2013.

[17] I. Guyon and A. Elisseeff, “An introduction to variable and feature selection,” *JMLR Special Issue on Variable and Feature Selection,* pp. 1157-1182, 2003.

[18] J. Olvera-López, J. Carrasco-Ochoa, J. Martínez-Trinidad and J. Kittler, “A review of instance selection methods,” *Artif Intell Rev,* vol. 34, pp. 133-143, 2010.

[19] P. E. Hart, “The condensed nearest neighbor rule,” *IEEE Trans Inf Theory,* vol. 14, pp. 515-516, 1968.

[20] G. L. Ritter, H. Woodruff, S. R. Lowry and T. L. Isenhour, “An algorithm for a selective nearest neighbor decision rule,” *IEEE Trans Inf,* vol. 21, pp. 665-669, 1975.

[21] C. Chien-Hsing, K. B.H. and C. Fu, “The generalized condensed nearest neighbor rule as a data reduction method,” *in 18th international conference on pattern recognition,* pp. 556-559, 2006.

[22] D. L. Wilson, “Asymptotic properties of nearest neighbor rules using edited data,” *IEEE Transactions on Systems, Man and Cybernetics,* vol. 1, pp. 408-421, 1972.

[23] I. Tomek, “An experiment with the edited nearest-neighbor rule,” *IEEE Trans Syst Man Cybern,* vol. 6, pp. 448-452, 1976.

[24] K. Kyoung-jae, “Artificial neural networks with evolutionary instance selection for financial forecasting,” *Expert Systems with Applications,* vol. 30, pp. 519-526, 2006.

[25] Z. Li-Ning, Z. Qi, L. Da-Chao and A. Jing, “Research on the Pre-warning Model of Enterprise Financial Crisis Based on the Information Entropy and PSO-ANN,” *International Conference on E-Business and E-Government (ICEE),* pp. 1567-1570, 2010.

[26] J. Ruiz-Shulcloper and M. Abidi, “Logical Combinatorial Pattern Recognition: A Review,” de *Recent Research Developments in Pattern Recognition. Transword Research Networks*, 2002, pp. 133-176.

[27] J. A. Carrasco-Ochoa and J. F. Martínez-Trinidad, “Editing and Training for ALVOT, an Evolutionary Approach,” *Lecture Notes in Computer Science,* vol. 2690, pp. 452-456, 2003.

[28] J. A. Carrasco-Ochoa and J. F. Martínez-Trinidad, “Combining Evolutionary Techniques to Improve ALVOT Efficiency,” *WSEAS Transactions on Systems,* vol. 2, pp. 1073-1078, 2003.

[29] C. Decaestecker, “NNP: A neural net classifier using prototype,” *International Conference on Neural Networks,* pp. 822-824, 1993.

[30] A. König, R. J. Rashhofer and M. Glesner, “A novel method for the design of radial-basisfunction networks and its implication for knowledge extraction,” *International Conference on Neural Networks,* pp. 1804-1809, 1994.

[31] I. López-Yáñez, “Clasificador automático de alto desempeño,” Mexico, D.F., 2007.

[32] L. C. Yáñez-Márquez and J. L. Díaz, “Memorias Asociativas basadas en relaciones de orden y operaciones binarias,” *Computación y Sistemas,* vol. 6, nº 4, pp. 300-311.

[33] I. López-Yáñez, “Teoría y aplicación del clasificador asociativo Gamma,” Mexico, D.F., 2011.

[34] I. R. Miller, J. E. Freund and R. Johnson, Probabilidad y Estadística para Ingenieros, La Habana: Félix Varela, 2005.

[35] G. T. Toussaint, “Proximity Graphs for Nearest Neighbor Decision Rules: Recent Progress,” de *34 Symposium on Computing and Statistics INTERFACE-2002*, Montreal, Canada, 2002.

[36] J. Demsar, “Statistical comparison of classifiers over multiple datasets,” *The Journal of Machine Learning Research,* vol. 7, pp. 1-30, 2006.

[37] Z. Pawlak, “Rough Sets,” *International Journal of Information & Computer Sciences,* vol. 11, pp. 341-356, 1982.

[38] Y. Caballero, R. Bello, Y. Salgado and M. M. García, “A method to edit training set based on rough sets,” *International Journal of Computational Intelligence Research,* vol. 3, pp. 219-229, 2007.

[39] J. L. Díaz and C. Yáñez-Márquez, “Memorias Asociativas basadas en relaciones de orden y operaciones binarias”.

[40] M. García-Borroto, Y. Villuendas-Rey, J. A. Carrasco-Ochoa and J. F. Martinez-Trinidad, “Using Maximum Similarity Graphs to edit nearest neighbor classifiers,” *Lecture Notes on Computer Science,* vol. 5856, pp. 489-496, 2009.

[41] M. A. Medina-Pérez, M. García-Borroto, Y. Villuendas-Rey and J. Ruiz-Shulcloper, “Selecting Objects for ALVOT,” *Progress in Pattern Recognition, Image Analysis and Applications,* pp. 606-613, 2006.

[42] A. Newman and D. Asuncion, “UCI machine learning repository,” 2007.