Título artículo / Títol article:	A hybrid method to face class overlap and class imbalance on neural networks and multi-class scenarios
Autores / Autors	Alejo, R. ; Valdovinos, R. M. ; García Jiménez, Vicente ; Pacheco-Sanchez, J. H.
Revista:	Pattern Recognition Letters, 2013, Marzo, Vol. 34, nº 4
Versión / Versió:	Pre-print
Cita bibliográfica / Cita bibliogràfica (ISO 690):	ALEJO, Roberto, et al. A hybrid method to face class overlap and class imbalance on neural networks and multi-class scenarios. *Pattern Recognition Letters*, 2013, vol. 34, no 4, p. 380-388.
url Repositori UJI:	http://hdl.handle.net/10234/91710

A hybrid method to face with class overlap and class imbalance on multi-class scenarios

*R. Alejo[a], R.M. Valdovinos[b], V. García[c], J.H., Pacheco-Sanchez[d]

[a]*Tecnológico de Estudios Superiores de Jocotitlán*
Carretera Toluca-Atlacomulco KM. 44.8, Col. Ejido de San Juan y San Agustín, 50700 Jocotitlán (Mexico)
[b]*Centro Universitario Valle de Chalco, Universidad Autónoma del Estado de México*
Hermenegildo Galena No.3, Col. Ma. Isabel, 56615 Valle de Chalco (Mexico)
[c]*Institute of New Imaging Technologies, Universitat Jaume I*
Av. Sos Baynat s/n, 12071 Castelló de la Plana (Spain)
[d]*Instituto Tecnológico de Toluca, Av. Tecnolgico s/n Ex-Rancho La Virgen, 52140, Metepec, (Mexico)*

Abstract

Class imbalance and class overlap are two of the major problems in data mining and machine learning. Several studies have shown that these data complexities may affect the performance or behavior of artificial neural networks. Strategies proposed to face with both challenges have been separately applied. In this work, we introduce a hybrid method for handling both class imbalance and class overlap simultaneously in multi–class learning problems. Experimental results on three remote sensing data show that the combined approach is a promising method.

Keywords: Multi-class imbalance, overlapping, back-propagation, cost function, editing techniques.

*Corresponding author Tel.: +52(712)1231313; Fax:+52(712)1210113; E-mail address:ralejoll@hotmail.com.

1. Introduction

In supervised classification learning, the intrinsic difficulties in the data may significantly affect generalization performance of standard classifier algorithms. An important issue that has been identified into the 10 challenging problems is when the datasets suffer from skewed class distributions, that is, the number of samples of one class out numbers the other classes (class imbalance) [1]. Existing research indicates that class imbalance problem causes seriously negative effects on the classification performance [2], since the classifier algorithms are often biased towards the majority classes [3]. This phenomenon appears with high frequency in many real–world applications where it is often costly misclassified examples of the minority class. Typical examples are remote sensing [4], medical diagnosis [5], biological data analysis [6], fraud detection [7] and credit assessment [8].

Most of the research addressing the imbalance problem can be grouped into three categories: (*i*) Assigning distinct costs to the classification errors for positive and negative samples [9, 2], (*ii*) Resampling the original training data set, either by over-sampling the minority class and/or under-sampling the majority class until the classes are approximately equally represented [10, 11, 12], and (*iii*)

Internally biasing the discrimination-based process so as to compensate for the class imbalance [13, 4, 14].

It is generally accepted that imbalance is the main responsible for a significant degradation of the performance on individual classes. However, recent works have pointed out that there does not exist a direct correlation between class imbalance and the loss of performance. These studies suggest that the class imbalance is not a problem by itself, but the degradation of performance is also related to other factors, i.e., the degree of class overlapping [15, 16, 17]

The class overlapping occurs in those zones where the decision boundary regions intersect. The overlapped samples have a high probability of being misclassified for any classifier. Hence, several Instance Selection (IS) methods has been developed to address this challenging task [18]. The IS approaches that seek to remove points that are noisy or do not agree with their neighbours are called Edition algorithms. The most popular editing methods are based on the nearest neighbour rule.

Class overlap and class imbalance has been widely studied in the literature and treated separately. Rarely, however, both at the same time. There are also very few approaches facing with this complexities in multi-class scenarios. In this paper, we introduce a novel hybrid algorithm to face with class imbalance and class overlapping simultaneously on multi-class problems.

This method is based on using a Gabriel graphs editing technique to remove noisy and border–line negative samples to reduce the class overlapping, and then a modified the back-propagation algorithm to face with imbalanced classes. Our main contributions in this paper are: *a*) to propose a new cost function (based in the mean square error) to deal with the class imbalance problem, *b*) we adapted the Gabriel graphs editing (GGE) to become it effective to reduce the class overlap in the neural network context and *c*) to combine the point *a* and *b* to generate an effective strategy dealing with the class overlap and class imbalance.

The rest of this paper is organized as follows. Related works are briefly reviewed in Section 2. In Section 3 we introduce the modified back-propagation algorithm for tackling the class imbalance problem. The editing algorithm is described in Section 4. In section 5 we present a hybrid method dealing with the class overlap and class imbalance. Section 6 and 7 we show the experimental set up and results respectively. Finally, section 8 is the conclusion.

2. Related Works

Back-propagation is now the most widely used tool in the field of artificial neural networks (NN). However, despite the general success of back-propagation, several major deficiencies are still needed to be solved. The major disadvantage of back-propagation is the slow rate of convergence of net output error; this is especially a major difficulty in “imbalanced” classification problems [19, 3], i.e., where the training set contains many more samples of some “dominants” classes (majority classes) than the other “subordinates” classes (minority classes).

In the back-propagation algorithm, the class imbalance poses severe problems in training stage as the learning process becomes biased towards the majority classes, ignoring the minority classes and leaving them poorly trained at the end of the training stage. The learning process also becomes slower an it take a longer time to converge to expect solution [19].

Many researches has been done in addressing the class imbalance problem [2]. In the NN field, the modified learning algorithm has been proposed for dealing with this problem. In reference [19] a modified back-propagation is proposed, this consists of calculating a direction in weight-space which decreases the error for each class (majority and minority class) in the same magnitude, in order to accelerate the learning rate for two-class imbalance problems. In the reference [4, 20, 3, 14], the error function was modified by introducing different costs associated with making errors in different classes. Basically, when the sum of square errors is calculated, each term is multiplied by a class dependent (regularization) factor. This compensates class imbalance [4, 20, 14] and accelerates the convergence of the NN [3]. However, the main drawback of these approaches is the use of free parameters, because these parameters control the updating amount of weights whether training samples are in the minority or majority classes.

The most popular strategies to deal with the class imbalance problem have been at the data level. These methods for balancing the classes are the most investigated because they are independent of the underlying classifier and can be easily implemented for any problem. The data level methods resampling the original dataset, either by over-sampling the minority class or by under-sampling the majority class, until the classes are approximately equally represented. Both strategies can be applied in any learning system since they act as a preprocessing phase, thus allowing the system to receive the training instances as if they belonged to a well-balanced dataset. By using this strategy, any bias of the learning system towards the majority class due to the skewed class priors will hopefully be eliminated.

The simplest method to increase the size of the minority class corresponds to random over-sampling, that is, a non-heuristic method that balances the class distribution through the random replication of positive examples. Nevertheless, since this method replicates existing examples in the minority class, overfitting is more likely to occur. Chawla et al.[10] proposed an over-sampling technique that generates new synthetic minority samples by interpolating between several pre-existing positive examples that lie close together. This method, called SMOTE (Synthetic Minority Over-sampling TEchnique), allows to the classifier to build larger decision regions that contain nearby samples from the minority class.

On the other hand, random under-sampling [21] aims at balancing the dataset through the random removal of negative examples. Despite its simplicity, it has empirically been shown to be one of the most effective resampling methods. Unlike the random approach, many other proposals are based on a more intelligent selection of the negative examples to be eliminated.

Several works point out class imbalance as an obstacle when applying machine learning algorithms to real world domains. However, in some cases, learning algorithms perform well on several imbalanced domains [22]. Recent work shows that class imbalance is not always a problem [17, 16]. Japkowicz and Stephen [21] suggest that some classifiers are not sensitive to the class imbalance problem in cases where the classes are separable. In the same way some researchers [20, 23] affirm that the class imbalance is not an intrinsic problem if the distributions do not overlap.

The overlapping appears when the samples of the minority class share a region with the majority one, where all the samples are intertwined (this is an intrinsic problem of the data). García et al. [17] have shown that overlap can play an even larger role in determining classifier performance than the class imbalance problem. Lawrence et al. [20] suggest that when distribution is overlapped, it is desirable to pre-process or editing the data in a manner that results in reduced overlap. The similar idea was studied in [22]. That work shows that data cleaning strategies usually lead to a performance improvement for highly overlapped datasets. Tang and Gao [24] use the inverse k-nearest neighbor and k-nearest neighbor (K-NN) algorithms to eliminate potential noisy patterns, and extraction of boundary pattern. The goal of that work is to deal with the classification problem, which involves class overlapping. Nevertheless, the main drawback of these approaches is that parameter setting in k-NN impacts directly on the classification performance. Kretzschmar et al. [25] introduce variance-controlled NN (VCNNs), which were developed to handle class overlap. These VCNNs are feed forward models trained by minimizing an error function involving the class-specific variance (CSV) computed at their outputs. This minimization suppresses abrupt changes in the responses of the trained classifiers in regions of the input space occupied by overlapping classes. The main restriction is that VCNNs require the selection of additional free parameter (to adjust of influence of CSV) specified empirically by the user.

3. A Modified Back-Propagation (MBP)

The multilayer perceptron (MLP) neural network [26] usually comprises one input layer, one or more hidden layers, and one output layer Input nodes correspond to features, hidden layers are used for computations, and output nodes are related with the number of classes. A neuron is the elemental unit of each layer. It computes the weighted sum of its inputs, adds a bias term and drives the result thought a generally non-linear (commonly a sigmoid) activation function to produce a single output.

The most popular training algorithm for MLP is the back-propagation algorithm, which uses a set of training instances for the learning process. Given a feed-forward network, the weights are initialized to small random numbers. Each training instance sent through the network and the output from each unit is computed. The target output is compared with the estimated output of the network by calculating the error, which is fed-back through the network.

To adjust the weights, the back-propagation algorithm uses a gradient descent to minimize the squared error. At each unit in the network starting from the output unit and moving to the hidden units, its error value is used to adjust the weights of its connections as well as to reduce the error. This process is repeated for a fixed number of times, or until the error is small.

On other hand, in the back-propagation algorithm the class imbalance problem generates unequal contributions to the mean square error (MSE) in the training phase [19]. Clearly the major contribution to the MSE is produced by the majority class.

Let us consider a training dataset (TDS) with two classes ($J = 2$) such that $N = \sum_j^J n_j$ and n_j is the number of samples from class j. Suppose that the MSE by class can be expressed as

$$E_j(U) = \frac{1}{N} \sum_{n=1}^{n_j} \sum_{p=1}^{J} (t_p^n - z_p^n)^2 \,, \tag{1}$$

where t_p^n is the desired output and z_p^n is the actual output of the network for the sample n.Then the overall MSE can be expressed as

$$E(U) = \sum_{j=1}^{J} E_j(U) = E_1(U) + E_2(U) \,. \tag{2}$$

If $n_1 << n_2$ then $E_1(U) << E_2(U)$ and $\|\nabla E_1(U)\| << \|\nabla E_2(U)\|$, consequently $\nabla E(U) \approx \nabla E_2(U)$. So, $-\nabla E(U)$ it is not always the best direction to minimize the MSE in both classes. [19].

Considering that the class imbalance problem affects negatively in the back-propagation algorithm due to the disproportionate contributions in the MSE, it is possible to consider a cost function (γ) that balance the MSE as follows:

$$
\begin{aligned}
E(U) &= \sum_{j=1}^{J} \gamma(j) E_j = \gamma(1)E_1(U) + \gamma(2)E_2(U) \\
&= \frac{1}{N} \sum_{j=1}^{J} \gamma(j) \sum_{n=1}^{n_j} \sum_{p=1}^{J} (t_p^n - z_p^n)^2 ,
\end{aligned}
\tag{3}
$$

where $\gamma(1)\|\nabla E_1(U)\| \approx \gamma(2)\|\nabla E_2(U)\|$ avoiding that the minority class be ignored in the learning process. In this work, we propose a new cost function defined as:

$$
\gamma(j) = \frac{\|\nabla E_{max}(U)\|}{\|\nabla E_j(U)\|} \tag{4}
$$

where $\|\nabla E_{max}(U)\|$ corresponds to the largest majority class.

On the other hand, when a cost function is included in the training process, the data probability distribution is altered [20]. Nevertheless, the cost function $\gamma(j)$ (Eq. 4) reduces its impact in the data distribution probability because the cost function value is diminished gradually. In this way, the class imbalance problem is reduced in early iterations, and later $\gamma(j)$ reduces its effect on the data distribution probability.

4. Editing technique for handling class overlap

The editing techniques have been proposed to remove noisy samples as well as close border cases (overlapping), leaving smoother decision boundaries [27]. The aim is to improve the classifier accuracy. The most popular editing schemes are based on the well-know k Nearest Neighbour (k-NN) rule. However, this rule only takes into account the distances to a number of close neighbors. Alternative concepts of neighborhood have been proposed to consider geometrical relation between a sample and some of its neighbours [28].

The Gabriel Graph has recently been used for introducing a set of editing methods for the k-NN rule [29]. The Gabriel Graph Editing (GGE) consists of applying the general idea of Wilson's algorithm [30], but using the graph neighbours of each sample instead of the Euclidean or other norm-based distance neighbourhood. Two samples x and y are graph neighbours in a GG $= (V, E)$ if there exists an edge $(x, y) \epsilon E$ between them. Taking into account the definitions of GG, the graph neighbourhood of a given point requires that no other point lies inside the union of the zones of influence (i.e. hypersphere of influence) corresponding to all its graph neighbours.

The application of GGE has some additional properties as compared to the conventional methods: first, they consider the number of neighbours as a variable feature which depends upon every prototype. Secondly, since the graph neighbourhood of a sample always tends to be widely distributed around it, the information extracted from samples close to decision boundaries may be richer in the sense of the prototypes distribution [28].

The original GGE was proposed to improve the k-NN accuracy [29]. However, in this work the original GGE was adapted to do it effective in the back-propagation context. The aim was to remove noisy and overlapping samples of the majority classes, but keeping all the positive samples. This task allows improving the back-propagation learning over the minority classes. The proposed GGE can be summarized as follows:

- For each sample p, constructs the corresponding GG.
- Consider p in the Training Dataset (TDS), if all its graph neighbours are of its same class.
- Other issue, if p belongs to some majority class, then discard p from TDS.

5. Methodology for dealing with class imbalance and the class overlapping on multi-class problems

This section provides an overview of the method here proposed to deal with class imbalance and class overlapping simultaneously, which consists of combining an editing technique and a cost function. This strategy can be summarized as follows:

1. MBP: To deal with class imbalance problem.
   (a) To modify the back-propagation (MBP) algorithm applying a cost function (Eq. 4) in order to avoid that the minority classes would be ignored in the training process, and to accelerate the convergence of the neural network.
2. GGE: To deal with class overlapping problem.
   (a) To edit the TDS with the GGE technique (sec. 4), removing only majority samples in the overlap region and producing a local balance of the classes.
3. MBP + GGE (Proposed strategy).
   (a) To train the MLP with the modified back-propagation algorithm over the edited TDS.

6. Experimental Protocol

In this section we first provide details of the data sets chosen for the experimentation, the performance measures used to evaluate the classifiers and the resampling methods. Finally, a briefly description of the configuration parameters of the methods.

6.1. Database description

We used in our experiments five remote sensing datasets: Cayo, Feltwell Satimage, Segment and 92AV3C. Feltwell is related to an agriculture region near to Felt Ville, Feltwell (UK) [31], Cayo represents a particular region in the gulf of Mexico, and Satimage consists of the multi-spectral values of pixels in 3x3 neighborhoods in a satellite image. Segment contains instances drawn randomly from a dataset of 7 outdoor images [32]. 92AV3C dataset[2] corresponds to a hyperspectral image (145x145 pixels) taken over Northwestern Indianas Indian Pines by the AVIRIS sensor.

In order to covert Cayo in a highly imbalanced dataset some of their classes were merged as follows: join the classes 1,3,6,7 and 10 for integrating the class 1; join the classes 8, 9 and 11 for integrating the class 3, finally, the rest of classes (2,4 and 5) we obtain from original dataset. M92AV3C is a subset of 92AV3C, it contains six classes (2, 3, 4, 6,7 and 8) and 38 attributes. The attributes were

[2] `https://engineering.purdue.edu/biehl/MultiSpec/hyperspectral.html`

selected using a common features selection algorithm (Best-First Search [33]) implemented in WEKA[3]:

Feltwell,Satimage, Segment and 92AV3C were random under-sampling to generate severe class imbalance datasets. A brief summary of these multi-class imbalance datasets is shown in the Table 1. Note that are highly imbalanced datasets. For each database, a 10–fold cross–validation was applied. The datasets were divided into ten equal parts, using nine folds as training set and the remaining block as test set.

Table 1: A brief summary of some basic characteristics of the datasets. The bold numbers represent the samples of minority classes.

dataset	Size	Attr.	Class	Class distribution
MCayo	6019	4	5	2941/**293**/2283/**322**/**133**
MFelt	10944	15	5	3531/2441/**91**/2295/**178**
MSat	6430	36	6	1508/1531/**104**/1356/**93**/**101**
MSeg	1470	19	7	330/**50**/330/330/**50**/**50**/330
M92AV3C	5062	38	6	**190**/**117**/1434/2468/747/**106**

[3]Available in: `http://www.cs.waikato.ac.nz/ml/weka/`

6.2. *Classifier performance and Significance Statistical Test*

The most traditional metric for measuring the performance of learning systems is the accuracy which can be defined as the degree of fit (matching) between the predictions and the true classes of data. However, the use of plain accuracy to evaluate the classifiers in imbalanced domains might produce misleading conclusions, since it is strongly biased to favour the majority classes [34, 14]. Shortcomings of this evaluator has motivated search for new measures. One the most widely-used techniques for the evaluation of binary classifiers in imbalanced domains is the Receiver Operating Characteristic curve (ROC), which is a tool for visualizing, organizing and selecting classifiers based on their trade-offs between true positive rates and false positive rates. Furthermore, a quantitative representation of a ROC curve is the area under it, which is known as AUC [35]. The AUC measure for multi-class problems can be defined as:

$$AUC = \frac{2}{\|J\|(\|J\| - 1)} \sum_{j_i, j_k \epsilon J} AUC_R(j_i, j_k) \tag{5}$$

where AUC$_R(j_i, j_k)$ is the AUC for each pair of classes j_i and j_k.

Kubat and Matwin [36] use the the geometric mean of accuracies measured separately on each class. For multi-class problems it can be computed as:

$$g - mean = (\prod_{i=1}^{J} acc_i)^{\frac{1}{J}} \ , \tag{6}$$

where acc_i is the accuracy on the class i and J the number of classes.

Statistical tests are used to evaluate whether the performance of a new method or learning algorithm on the same problem is significantly different. Into the framework of empirical analysis, the Student's paired t-test is the most widely used parametric statistical procedure. However, it is well-known that it is conceptually inappropriate and statistically unsafe to require certain assumptions like the data is normally distributed [37]. In this work, we adopt the non–parametric statistical Friedman test to perform a multiple comparison, which is equivalent of the repeated-measures ANOVA. This test used to check if all methods perform equal on the selected datasets can be rejected. The first step in calculating the test statistic is to rank the algorithms for each dataset separately; the best performing algorithm should have the rank of 1, the second best rank 2, etc. The Friedman test uses the average rankings to calculate the Friedman statistic, which can be computed as,

$$\chi_F^2 = \frac{12N}{K(K+1)}(\sum_j R_j^2 - \frac{K(K+1)^2}{4}) \tag{7}$$

where K denotes the number of methods, N the number of data sets, and R_j the average rank of method j on all datasets. Iman and Davenport [38] showed that χ_F^2 presents a conservative behaviour, so they proposed a better statistic distributed according to the $F-$distribution with $K-1$ and $(K-1)(N-1)$ degrees of freedom,

$$F_F = \frac{(N-1)\chi_F^2}{N(K-1)-\chi_F^2} \tag{8}$$

When the null-hypothesis is rejected, we can use post-hoc tests in order to find the particular pairwise comparisons that produce statistical significant differences. The Bonferroni-Dunn post-hoc test is applied to report any significant difference between individual methods here used. The test uses the average rank of each method and compare it to each other if these differ by at least the critical difference, which is given by

$$CD = q_\alpha \sqrt{\frac{K(K+1)}{6N}} \tag{9}$$

where the value q_α is based on the studentized range statistic divided by $\sqrt{2}$.

6.3. Resampling Methods

SMOTE, and random under sampling (RUS) are used in the empirical study, because are a popular approaches to deal with the class imbalance problem. However, it methods have internal parameters that enable the user to set up the resulting class distribution obtained after the application of these methods. In this paper, we decided to add or remove examples until a balanced distribution was reached. This decision was motivated for two reasons: a) by simplicity (to avoid use many free parameters) and b) by effectiveness. Results obtained with the other classifiers [39], have shown that when AUC is used as a performance measure, the best class distribution for learning tends to be near the balanced class distribution.

6.4. Neural network configuration

The MLP was trained with the standard back-propagation (SBP) and modified back-propagation (MBP) algorithm in batch mode. For each TDS, MLP was initialized ten times with different weights. The results here included correspond to the average of those achieved in the ten different initialization and of ten partitions. The learning rate (η) was set to 0.1 and only one hidden layer was used. The stop criterion was established at 25000 epoch or an MSE below to 0.001. The number of neurons for the hidden layer was obtained from the trial and error strategy. So, the number of neurons was 7, 6, 12, 10, 10 for MCayo, MFelt, and MSat, MSement and M datasets respectively.

7. Results and discussion

In order to asses the performance of the proposed method, we have carried out an experimental comparison with respect to well-known resampling approaches. In total, seven strategies were examined: (*i*) Standard Back-Propagation Algorithm (SBP), (*ii*) Modified Back-Propagation Algorithm (MBP), (*iii*) Standard Back-Propagation with Grabiel Graph Editing (SBP+GGE), (*iv*) Modified Back-Propagation with Grabiel Graph Editing (MBP+GGE), (*v*) SMOTE, (*vi*) SMOTE with Grabiel Graph Editing (SMOTE+GGE) and (*vii*) Random Under Sampling (RUS). The datsets that were preprocessed by the SMOTE, SMOTE+GGE and RUS strategies were applied to the SBP algorithm.

In this paper, we have omitted other neural networks approaches as the two-phase technique [4], threshold moving [2], or modified error function [14], because these methods contain many prior free parameters, so it is difficult to make a fair comparison.

With the aim of show the effectiveness of combining the MBP and the GGE techniques, in Fig. 1, the performance by class of the SBP, the MBP, the SBP+GGE and the proposed strategy (MBP+GGE), are presented separably (the bold boxes belong to minority classes). The results show that the minority classes performance is severally affected by the class imbalance. In Fig. 1a, 1e, 1i, and 1q are observed that the class imbalance problem cause that some minority classes are not enough learned. So these minority classes show 0% of accuracy. The effects the class imbalance problem is slow down the convergence of the SBP due to disproportionate contribution in the MSE in the training phase (see section 3). An immediate consequence of this, is the difficulty of achieving effective performance (in terms of classification) in a “reasonable” time. Especially in situations where there is an extreme class imbalance.

On other hand, the Fig. 1 shows that when the class imbalance is compensated (MBP) the minority classes performance is improve (Fig. 1b, 1f, 1j, 1n, and 1r). However, in hight overlapped TDS is not enough (Fig. 1j and 1r).

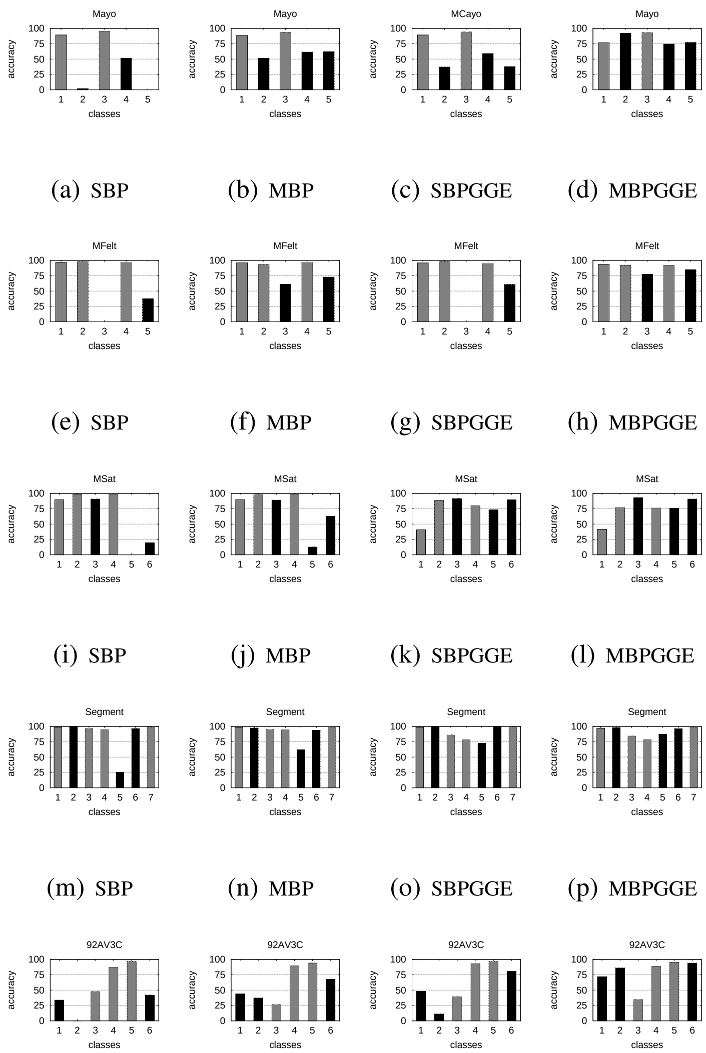
GGE technique is used to reduce the overlapping between classes. Fig. 1c, 1g, 1k, 1o, and 1s, present the results obtained to apply the GGE technique. Note that it archive improve the minority classes performance, specially in overlapped TDS (see class 5 in Fig. 1k and 1o). Nevertheless, the class imbalance problem continues to affect. For example, observe Mfelt, and M92AV3C datasets (Fig.1g and 1s respectively). A negative consequence of GGE technique is that when increase the minority classes accuracy, the majority classes performs is affected.

The four column of the Fig. 1 presents the combining the MBP and GGE (MBP+GGE). These results show a remarkable improvement in minority classes performance and exhibit a better performance that to apply individually the MBP and GGE techniques.

The modification of the training algorithm including a cost function (MBP) increases the recognition rate of less represented classes, accelerating the convergence of the network, and to apply GGE technique reduce the confusion of the minority classes in the overlap region. So the results presented in Fig. 1d, Fig. 1h, Fig. 1l, Fig. 1p and Fig. 1t, demonstrate the effectiveness of combining the MBP and GGE techniques.

Fig. 2 shows experimental results of compare the proposed method with respect to others well-known resampling approaches. The experimental results are presented in graphics where boxes represent the accuracy by class, and the bold boxes belong to minority classes. Fig. 2 exhibits that, the worst accuracy for the minority classes is shown by the RUS strategy (mainly over MFelt and MSat, see Fig. 2*h* and 2*l*). This means that when TDS is severely imbalanced removes samples to balance the class distribution, and it is not effective on back-propagation, because the RUS involves a loss of useful information that could be important for the training process. In the M92AV3C and MSeg datasets, the RUS shows a good minority classes performance, however, is not a tendency.

SMOTE was very successful in MCayo and MFelt, but in MSeg, Sat, and



(a) SBP (b) MBP (c) SBPGGE (d) MBPGGE

(e) SBP (f) MBP (g) SBPGGE (h) MBPGGE

(i) SBP (j) MBP (k) SBPGGE (l) MBPGGE

(m) SBP (n) MBP (o) SBPGGE (p) MBPGGE

(q) SBP (r) MBP (s) SBPGGE (t) MBPGGE

Figure 1: The comparison of methods deal with class imbalance problem and class overlap. The graphics shows accuracy by class. The bold boxes belong to minority classes. The acronyms SMO, it mean SMOTE.

M92AV3C datasets the minority classes performance is worst than the proposed method (see class 5 in Fig.2j and 2n, and class 1 in Fig. 2r). We believe that the explanation is that these datasets present high level of overlapping. For example MSat dataset shows high level of overlapping between the C-01 and C-05 classes, in other words, it is not enough to balance the TDS for improving the classifier performance over minority classes when the TDS overlaps. This is the reason of the low accuracy in class C-05 for RUS, MBP and SMOTE.

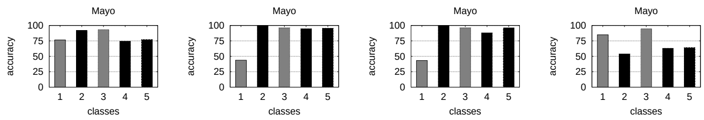
On other hand, MBP+GGE presents better results than the SMOTE for overlapping datasets (see MSeg MSat, and M92AV3C, Fig. 2 *j*, *n* and *r*), this is due to the data cleaning method (GGE) is more efficient in highly overlapped regions. The MBP+GGE method starts to be less effective as overlapping is reduced (for example see MCayo and MFelt in Fig. 2).

The accuracy showed by SMOTE+GGE was very similar at SMOTE, however, despite of that SMOTE+GGE include GGE technique this method was ineffective on overlapped datasets (see MSat, Fig. 2 *l*). The explanation is that, as SMOTE was firstly applied the overlap level was increased too, thus GGE was not able to remove the enough overlap for improving the accuracy of minority class. To prove this, we repeat the experiment: we first applied GGE over MSat, and then MSat was over-sampled using SMOTE. The results obtained were very successful and similar at achieved by MBP+GGE. The AUC= $0.756(0.050)$, *g-mean*= $0.713(0.071)$, C-05 accuracy = $0.91(0.02)$. This results show the effective of the GGE technique to reduce the class overlap and for improving the accuracy of the classifier over minority classes.

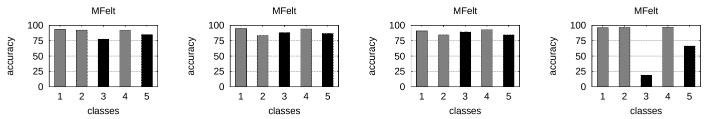
SMOTE and SMOTE+GGE strategies have made great improvement on the minority classes. However, they add information to the TDS by introducing new (non-replicate) minority classes samples, which involves a larger TDS and longer training times for the same number of training epochs. In addition, when the dataset present high overlapping the SMOTE can be not good choice, because can be increase the class overlapping. Meanwhile SMOTE+GGE is recommendable to apply first GGE and after the SMOTE, i.e., GGE+SMOTE.

Fig. 2 shows that the results obtained by MBP+GGE are very competitive with the results obtained by SMOTE and SMOTE+GGE. As well as MBP+GGE does not have internal parameters that the user needs to set up before to apply it and use of a TDS sight more reduced (much less training time). These are main advantages of MBP+GGE over SMOTE and SMOTE+GGE.

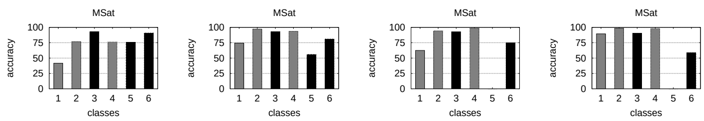
Table 2 summarize the experimental results in terms of AUC and $g-$mean on the five datasets when using six different strategies previously enumerated. For each method, the average ranking is shown. As can be seen in the table, the orig-



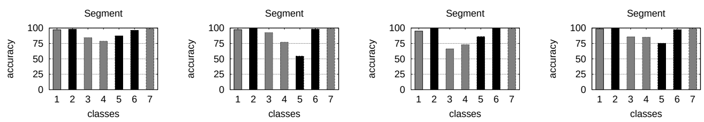
(a) MBPGGE (b) SBPSMO (c) SMOGGE (d) SBPRUS



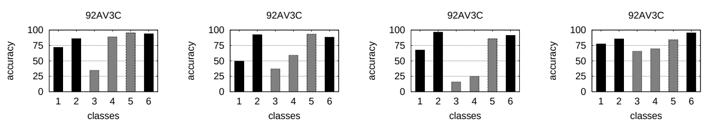
(e) MBPGGE (f) SBPSMO (g) SMOGGE (h) SBPRUS



(i) MBPGGE (j) SBPSMO (k) SMOGGE (l) SBPRUS



(m) MBPGGE (n) SBPSMO (o) SMOGGE (p) SBPRUS



(q) MBPGGE (r) SBPSMO (s) SMOGGE (t) SBPRUS

Figure 2: The comparison of methods deal with class imbalance problem and class overlap. The graphics shows accuracy by class. The bold boxes belong to minority classes. The acronyms SMO, it mean SMOTE.

inal (imbalanced) training set has the highest Friedman score (AR), which means that this strategy performs worse than other methods, whereas MBP+GGE is the best performing algorithm for AUC an $g-$mean. Note, that SMOTE performs equal to MBP+GGE when the results are evaluated with AUC.

Table 2: Performance on three datasets measured using AUC, $g-$mean and average rank (AR)

AUC							
Dataset	Imbalanced[1]	MBP	GGE	MBP+GGE	SMOTE[1]	SMOTE+GGE[1]	RUS[1]
MCayo	0.477 (0.020)	0.715 (0.034)	0.636(0.064)	0.828(0.040)	0.860 (0.040)	0.847 (0.024)	0.722 (0.035)
MFelt	0.658 (0.022)	0.839 (0.033)	0.700(0.017)	0.880(0.031)	0.895 (0.046)	0.884 (0.027)	0.749 (0.028)
MSat	0.663 (0.026)	0.752 (0.044)	0.774(0.049)	0.757 (0.041)	0.826 (0.038)	0.705 (0.038)	0.726 (0.031)
MSeg	0.871(0.032)	0.916(0.098)	0.905(0.030)	0.918(0.095)	0.880(0.053)	0.882(0.031)	0.914(0.027)
M92AV3C	0.512(0.061)	0.589(0.106)	0.615(0.039)	0.780(0.086)	0.690(0.136)	0.638(0.079)	0.796(0.054)
AR	7.0	4.2	4.6	2.4	2.4	3.8	3.6

$g-$mean							
Dataset	Imbalanced[1]	MBP	GEE	MBP+GGE	SMOTE[1]	SMOTE+GGE[1]	RUS[1]
MCayo	0.00 (0.00)	69.18 (4.18)	48.38(24.67)	81.99 (4.18)	82.24 (2.48)	80.63 (2.86)	70.22 (4.10)
MFelt	0.00 (0.00)	82.29 (4.10)	0.00(0.00)	87.54 (3.42)	89.05 (5.30)	88.14 (2.88)	53.05 (27.77)
MSat	0.00 (0.00)	49.36 (28.5)	73.89(6.71)	72.27 (5.38)	80.12 (5.28)	0.000 (0.00)	0.00 (0.00)
MSeg	66.60(31.81)	90.09(9.91)	89.21(3.82)	91.29(9.46)	78.83(24.62)	85.57(9.53)	90.37(3.46)
M92AV3C	0.00(0.00)	49.40(12.73)	32.17(26.39)	73.33(8.51)	54.79(26.28)	41.40(23.12)	77.77(6.39)
AR	6.7	4.0	4.9	2.2	2.4	4.2	3.6

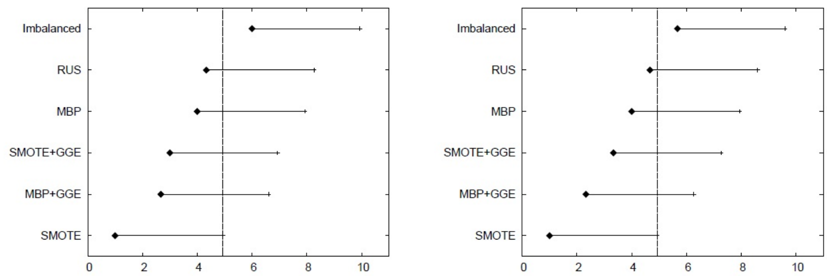
[1] Classification using SBP

The Iman and Davenport statistic computed using Equation 8 yields $F_F = 4.43$ and $F_F = 4.06$, for AUC and $g-$mean respectively. The critical value of the $F-$Distribution with 6 and 24 degrees of freedom for $\alpha = 0.05$ is 2.51. Given that the Iman and Davenport statistics are clearly greater than their associated critical value, the null-hypothesis that all methods perform equally can be rejected with a level of significance $\alpha = 0.05$. Then a post-hoc statistical analysis was used to detect significant differences for the control algorithm (method with the lowest ranking) in each measure.

Fig. 3 display a graphical representation of the results of Bonferroni-Dunn's post-hoc test, where for each method on the $y-$axis (ordered in ascending rank), the AR is plotted on the $x-$axis. For each AR we sum the critical difference obtained by the Bonferroni method, $CD = 3.60$ for $\alpha = 0.05$ in the two measures considered. The vertical dashed line segment represents the end of the best performing algorithm and the start of the next significantly method. MBP+GGE is the best algorithm, although according to Bonferroni-Dunn's test, only the difference to the Imbalanced approach is different[4].

The effects of the MBP+GGE can be better analysed by considering the num-

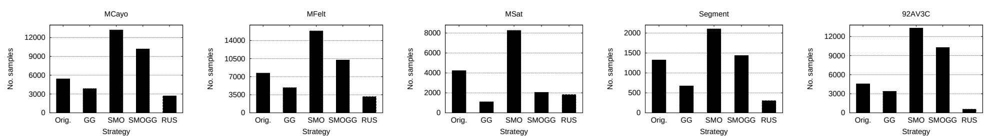
[4]Other powerful tests, such as Holm and Hochbergs ones would be necessary, for comparing the control algorithm with the rest of algorithms.



(a) AUC (b) $g-$mean

Figure 3: Bonferroni-Dunn's Critical Difference Diagram for AUC and $g-$mean

ber of samples that remain in the TDS after its application. Results in Figure 4 suggest a higher decrease in the size of the dataset when it is processed with the GGE, whereas using SMOTE increase twice of the original size. RUS reduce more the TDS size, however, not always present a good classifier performance. Reducing the dataset involve to reach a better neural network learning time and reduce storage requirements.



(a) MCayo (b) MFelt (c) MSat (d) MSeg (e) M92AV3C

Figure 4: Training size after resampling TDS with the techniques GGE, SMOTE, SMOTE+GGE and RUS. The acronyms Orig., GG and SMO, they mean Imbalance TDS, GGE and SMOTE respectively.

8. Conclusions

In this work, we propose an hybrid method (MBP+GGE) for dealing with class imbalance and the class overlapping on multi-class problems. The MBP+ GGE is based on combination of modified back-propagation (MBP) with a Gabriel graph editing technique (GGE). For modified back-propagation algorithm we proposed to include a new cost function (based on MSE) in the algorithm, and for doing effective the Gabriel graph editing we adapted it in the back-propagation context. MBP+GGE generates two effects: a) MBP: to compensate the class imbalance during the training process and b) GGE: to reduce the confusion of the minority classes in the overlap region. With the edition of the majority classes it is possible to reduce the confusion between the minority and majority classes.

The MBP+GGE strategy was compared with the conventional class imbalance techniques: RUS, SMOTE, MBP, GGE and SMOTE+GGE. Results show that SMOTE and SMOTE+GGE are very effective even with highly imbalanced datasets, but inadequate on overlapped datasets. MBP+GGE show a better performance on class overlap problems. The data cleaning step used in the MBP+GGE seems to be specially suitable in situations having a high degree of overlapping, moreover, GGE produces a small training dataset.

The SMOTE is needed to find the most appropriate re-sampling rate, i.e., to determine the number of samples when we introduce them in the minority classes before applying it. So the main advantages of MBP+GGE over SMOTE and SMOTE+GGE are: a) does not have internal parameters that the user needs to set up before applying it and b) use of a TDS sight more reduced (much less training time). As we see from the results, MBP+GGE is a very competitive strategy for dealing with class imbalance and the class overlapping on multi-class problems.

Further research is required to investigate the potential of the strategy proposed in this paper in "severe" multi-class imbalance and highly class overlapping problems. So, the exploration of the other editing strategies is necessary when approaching the graph based on editing scheme. Also, the study of new cost functions which help to speed up the neural network convergence in order to avoid altering the data probability distribution.

References

[1] Q. Yang, X. Wu, 10 challenging problems in data mining research, International Journal of Information Technology and Decision Making 5 (4) (2006) 597–604.

[2] Z.-H. Zhou, X.-Y. Liu, Training cost-sensitive neural networks with methods

addressing the class imbalance problem., IEEE Transactions on Knowledge and Data Engineering. 18 (2006) 63–77.

[3] S. Ramanan, T. Clarkson, J. Taylor, Adaptive algorithm for training pram neural networks on unbalanced data sets, Electronics Letters 34 (13) (1998) 1335–1336.

[4] L. Bruzzone, S. Serpico, Classification of imbalanced remote-sensing data by neural networks., Pattern Recognition Letters 18 (1997) 1323–1328.

[5] S. C. Mohammed Khalilia, M. Popescu, Predicting disease risks from highly imbalanced data using random forest, BMC Medical Informatics and Decision Making 11 (2011) 1–13.

[6] L. Al-Haddad, C. W. Morris, L. Boddy, Training radial basis function neural networks: effects of training set size and imbalanced training sets., Journal of Microbiological Methods 43 (1) (2000) 33–44.

[7] T. Fawcett, F. Provost, Adaptive fraud detection, Data Min. Knowl. Discov. 1 (3) (1997) 291–316.

[8] I. Brown, C. Mues, An experimental comparison of classification algorithms

for imbalanced credit scoring data sets, Expert Systems with Applications 39 (3) (2012) 3446 – 3453.

[9] P. Domingos, Metacost: a general method for making classifiers cost-sensitive, in: Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining, KDD '99, 1999, pp. 155–164.

[10] N. V. Chawla, K. W. Bowyer, L. O. Hall, W. P. Kegelmeyer, Smote: Synthetic minority over-sampling technique, Journal of Artificial Intelligence Research 16 (2002) 321–357.

[11] A. Estabrooks, T. Jo, N. Japkowicz, A multiple resampling method for learning from imbalanced data sets, Computational Intelligence 20 (1) (2004) 1836.

[12] S. García, F. Herrera, Evolutionary undersampling for classification with imbalanced datasets: Proposals and taxonomy, Evolutionary Computation 17 (2009) 275–306.

[13] R. Anand, K. Mehrotra, C. Mohan, S. Ranka, Efficient classification for multiclass problems using modular neural networks, IEEE Transactions on Neural Networks 6 (1) (1995) 117–124.

[14] S.-H. Oh, Error back-propagation algorithm for classification of imbalanced data, Neurocomputing 74 (6) (2011) 1058–1061.

[15] G. E. A. P. A. Batista, R. C. Prati, M. C. Monard, Balancing strategies and class overlapping, in: IDA, 2005, pp. 24–35.

[16] M. Denil, T. P. Trappenberg, Overlap versus imbalance, in: Canadian Conference on AI, 2010, pp. 220–231.

[17] V. García, R. A. Mollineda, J. S. Sánchez, On the k-nn performance in a challenging scenario of imbalance and overlapping, Pattern Analysis and Applications 11 (3) (2008) 269–280.

[18] J. Olvera-López, J. Carrasco-Ochoa, J. Martínez-Trinidad, J. Kittler, A review of instance selection methods, Artificial Intelligence Review 34 (2010) 133–143.

[19] R. Anand, K. Mehrotra, C. Mohan, S. Ranka, An improved algorithm for neural network classification of imbalanced training sets., IEEE Transactions on Neural Networks 4 (1993) 962–969.

[20] S. Lawrence, I. Burns, A. Back, A. Tsoi, C. L. Giles, Neural network classification and unequal prior class probabilities, in: G. Orr, K.-R. Müller,

R. Caruana (Eds.), Neural Networks: Tricks of the Trade, Lecture Notes in Computer Science, Springer Verlag, 1998, pp. 299–314.

[21] N. Japkowicz, S. Stephen, The class imbalance problem: A systematic study, Intell. Data Anal. 6 (5) (2002) 429–449.

[22] R. Prati, G. Batista, M. Monard, Class imbalances versus class overlapping: An analysis of a learning system behavior, in: MICAI, 2004, pp. 312–321.

[23] S. Visa, A. Ralescu, Issues in mining imbalanced data sets - a review paper, in: Artificial Intelligence and Cognitive Science Conference, 2005, pp. 67–73.

[24] Y. Tang, J. Gao, Improved classification for problem involving overlapping patterns, IEICE Transactions 90-D (11) (2007) 1787–1795.

[25] R. Kretzschmar, N. B. Karayiannis, F. Eggimann, Feedforward neural network models for handling class overlap and class imbalance, Int. J. Neural Syst. 15 (5) (2005) 323–338.

[26] C. M. Bishop, Neural Networks for Pattern Recognition, 1st Edition, Oxford University Press, USA, 1996.

[27] D. R. Wilson, T. R. Martinez, Reduction techniques for instance-based learning algorithms, Machine Learning 38 (3) (2000) 257–286.

[28] B. V. Dasarathy, J. S. Sánchez, S. Townsend, Nearest neighbour editing and condensing tools-synergy exploitation, Pattern Anal. Appl. 3 (1) (2000) 19–30.

[29] J. S. Sánchez, F. Pla, F. J. Ferri, Prototype selection for the nearest neighbour rule through proximity graphs, Pattern Recognition Letters 18 (6) (1997) 507–513.

[30] D. Wilson, Asymptotic properties of nearest neighbor rules using edited data, IEEE Transactions on Systems, Man and Cybernetics 2 (4) (1972) 408–420.

[31] S. Serpico, F. Roli, P. Pellegretti, G. Vemazza, Structured neural networks for the classification of multisensor remote-sensing images., in: Int. Geosci. Remote Sensing Symp., 1993, pp. 907–909.

[32] D. N. A. Asuncion, UCI machine learning repository (2007). URL `http://www.ics.uci.edu/$\sim$mlearn/{MLR}epository.html`

[33] R. Kohavi, G. H. John, Wrappers for feature subset selection, Artif. Intell. 97 (1-2) (1997) 273–324.

[34] H. He, E. Garcia, Learning from imbalanced data, IEEE Transactions on Knowledge and Data Engineering In Knowledge and Data Engineering 21 (9) (2009) 1263–1284.

[35] T. Fawcett, An introduction to roc analysis, Pattern Recogn. Lett. 27 (2006) 861–874.

[36] M. Kubat, S. Matwin, Addressing the curse of imbalanced training sets: one-sided selection, in: Proc. 14th International Conference on Machine Learning, Morgan Kaufmann, 1997, pp. 179–186.

[37] J. Demšar, Statistical comparisons of classifiers over multiple data sets, Journal of Machine Learning Research 7 (1) (2006) 1–30.

[38] R. L. Iman, J. M. Davenport, Approximations of the critical region of the friedman statistic, Communications in Statistics - Theory and Methods 9 (6) (1980) 571–595.

[39] G. M. Weiss, F. J. Provost, Learning when training data are costly: The

effect of class distribution on tree induction, J. Artif. Intell. Res. (JAIR) 19 (2003) 315–354.