Back Propagation with Balanced MSE Cost Function and Nearest Neighbor Editing for Handling Class Overlap and Class Imbalance*

R. Alejo[1], J.M. Sotoca[2] and V. García[2], and R.M. Valdovinos[3]

[1] Tecnológico de Estudios Superiores de Jocotitlán
Carretera Toluca-Atlacomulco KM. 44.8, Col. Ejido de San Juan y San Agustn, 50700 Jocotitlán (Mexico)
[2] Institute of New Imaging Technologies, Universitat Jaume I
Av. Sos Baynat s/n, 12071 Castelló de la Plana (Spain)
[3] Centro Universitario UAEM Valle de Chalco, Universidad Autónoma del Estado de México
Hermenegildo Galena No.3, Col. Ma. Isabel, 56615 Valle de Chalco (Mexico)

Abstract. The class imbalance problem has been considered a critical factor for designing and constructing the supervised classifiers. In the case of artificial neural networks, this complexity negatively affects the generalization process on under-represented classes. However, it has also been observed that the decrease in the performance attainable of standard learners is not directly caused by the class imbalance, but is also related with other difficulties, such as overlapping. In this work, a new empirical study for handling class overlap and class imbalance on multi-class problem is described. In order to solve this problem, we propose the joint use of editing techniques and a modified MSE cost function for MLP. This analysis was made on a remote sensing data . The experimental results demonstrate the consistency and validity of the combined strategy here proposed.

Keywords: Multi-class imbalance, Overlapping, backpropagation, cost function, editing techniques.

1 Introduction

Class imbalance constitutes one of the problems that has recently received most attention in research areas such as Machine Learning, Pattern Recognition and Data Mining. The class imbalance occurs when some classes heavily outhumber other classes. In the area of the artificial neural networks (NN) has been observed that the class imbalance problem causes important losses in the generalization capacity when the minority classes [1, 2] are learned, because these are often biased towards the majority class. This

* Partially supported by the Spanish Ministry of Education and Science under grants CSD2007–00018, TIN2009–14205–C04–04, and by Fundacio Caixa Castello–Bancaixa under grants P1–1B2009–04 and P1–1B2009–45, FE28/2009 (103.5/09/4195) of the Mexican PROMEP, 2703/2008U from the UAEM project and SEP-2003-C02-44225 from the Mexican CONACyT.

issue can be found in real–world applications from Government, Industry and Academic or Scientific Area [3–6].

Research on this topic can be roughly classified into three categories: assigning different classification error costs [7], resampling the original training set, either by oversampling the minority class and/or under-sampling the majority class until the classes are approximately equally represented [8, 9], and internally biasing the discrimination-based process so as to compensate for the class imbalance [10, 11].

Recently, several 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, such as the degree of overlapping between classes [12–14].

In this paper, we propose to combine two strategies for addressing the class overlap and the class imbalance for the classification of remote sensing data. The problem is of great relevance since very few approaches to deal with this challenge. In order to face such a problem, this work focus on the joint use of editing techniques and a modification in the mean square error (MSE) cost function for a multi–layer Percetron (MLP). This approach can be considered a two–stage method. Firstly, we remove noisy and borderline samples of the majority classes by application of editing techniques. Secondly, the edited data set is used for training a MLP with a modified MSE cost function, which overcomes the class imbalance problem.

2 Methodology

2.1 A Balanced MSE Cost Function for Backpropagation Algorithm

In the multilayer perceptron neural network (MLP) the training by Backpropagation algorithm is based on minimization of a cost function. One of the most popular used cost functions is the mean-square error (MSE) between the desired d_{zi} and the actual y_{zi} outputs for each class $i = 1, 2 \ldots J$,

$$E_i(U) = \frac{1}{N} \sum_{z=1}^{n_i} (d_{zi} - y_{zi})^2 \,, \tag{1}$$

where $N = \sum_i^J n_i$ is the total training samples and n_i is the size of class i.

For a two-class problem ($J = 2$) the mean square error function can be expressed as,

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

If $n_1 << n_2$ then $E_1(U) << E_2(U)$ and $\|\nabla E_1(U)\| << \|\nabla E_2(U)\|$, consequently $\nabla E(U) \approx \nabla E_2(U)$, which means that $-\nabla E(U)$.

To obtain a balanced MSE cost function, we introduce a parameter (γ) that balance the contributions of the MSE,

$$E(U) = \sum_{i=1}^{J} \gamma(i) E_i = \gamma(1) E_1(U) + \gamma(2) E_2(U) \tag{3}$$

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, the parameter γ is defined as

$$\gamma(i) = \|\nabla E_{max}(U)\| / \|\nabla E_i(U)\|, \tag{4}$$

where $\|\nabla E_{max}(U)\|$ corresponds to the largest majority class. When γ is included in the training process, the data probability distribution is altered [11]. However, this parameter (Eq. 4) reduces the 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.

2.2 Editing techniques

The editing techniques have been proposed to remove noise prototypes and possible overlap among classes from the training set. The aim is improve the classifier accuracy by producing smooth decision boundaries. One the most popular editing schemes is based on the well-know k-NN rule, which is mainly used for classification. However, this rule only take into account the distances to a number of close neighbors. Alternative concepts of neighborhood have been proposed to consider the neighbors of a sample in terms of proximity and spatial distribution (Surrounding Neighborhood).

The editing techniques was used to remove noisy samples of the majority classes but keeping all the positive examples. This task allows to improve the learning mechanics of the MLP. In next paragraphs we describe briefly basic concepts about editing algorithms.

Wilson Editing Wilson [15] developed the Edited Nearest Neighbor (ENN) algorithm in which **S** starts out the same as Traning Set (TS), and then each instance in S is removed if it does not agree with the majority of its k nearest neighbors (with k=3, typically). This eliminates noisy instances as well as close border cases producing smoother decision boundaries. Algorithmically, the ENN scheme can be expressed as follows:

1. Let $\mathbf{S} = \mathbf{X}$.
2. For each $\mathbf{x}_i$ in $\mathbf{X}$ do:
   - Discard $\mathbf{x}_i$ from $\mathbf{S}$ if it is misclassified using the k-NN rule with prototypes in $\mathbf{X} - \{\mathbf{x}_i\}$.

Editing Via Surrounding Approaches The Nearest Centroid Neighborhood (NCN) [16] refers to a concept in which neighborhood is defined taking into account the proximity of prototypes to a given input sample and maintaining their symmetrical distribution around it. The k-Nearest Centroid Neighborhood rule(k-NCN) [17] has been proved to overcome the traditional k-NN classifier in many practical situations. The

NCN Editing (NCNE) approach corresponds to slight modification of the original work of Wilson and basically consists of using the error estimated by the k-NCN classification rule.

Proximity graph editing scheme is based on the concepts of Gabriel Graph (GG) and Relative Neighborhood Graph (RNG) [18]. The method applies Wilson's editing algorithm [15] using proximity graphs (GG or RNG) for each sample instead of the Euclidean distance.

The Gabriel Graph Editing (GGE) and Relative Neighborhood Graph Editing (RNGE) can be summarized as follows: after computing the graph neighborhood of every sample in the original training set, discard those samples that are misclassified by their graph neighbors (instead of their k nearest neighbors).

These editing techniques provide some advantages in comparation to conventional methods. GGE and RNGE get some kind of information about prototypes close enough but homogeneously distributed around a given sample, which can be specially interesting to detect outliers close to the inter-class or decision boundaries. A more detailed description of GGE and RNGE can be found in [19].

2.3 Random under-sampling

Random under-sampling aims at balancing the data set through the random removal of negative examples. Despite its simplicity, it has empirically been shown to be one of the most effective resampling methods.

In this work, the random under-sampling is used to compare with the editing techniques, also, it was not hired to balance the training set.

3 Experimental Set–Up

In this part, a comparative was carried out among the strategies previously described to validate the methodology exposed in Section 2. The database used corresponds to remote-sensing task which is basically a multi-classification problem.

Experiments were conducted as follows:

Data set: A large data set, the Cayo data (4 bands, 11 classes) which corresponds to a spectral image with reference to a particular region in the Gulf of Mexico was employed in the experiments. The data set was transformed into five-class problems (MCayo) by joining the samples of several classes. The fourth column in Table 1 indicates the original classes that have been joined to shape the new classes. For instance, the samples of clases 1, 3, 6, 7, and 10 were combined to form the class C–01 and the original classes 2, 4 and 5 were left as C–02, C–04 and C–05, respectively.

Partitions: A stratified 10–fold cross–validation was employed.

Under–sampling strategies: random under–sampling (RUS), nearest centroid neighborhood editing (NCNE), Wilson's editing (ENN), relative neighborhood graph editing (RNGE) and Grabiel graph editing (GGE) were employed. All these techniques were applied over the majoriy classes. In the case of ENN and NCNE, the value of k has been set to 15 and 13, respectively.

Classifiers: We use a MLP with and without balanced cost function (Cost-MLP). Each one was trained with the back–propagation algorithm in batch mode. The following parameter settings were used: a learning rate η = 0.1 and one hidden layer with seven neurons.

Performance metrics: Overall accuracy, accuracy by class and the geometric mean of accuracies measured seperately on each class were used. These measures can be easily derived from a $m \times m$ matrix confusion as that given in Table 2. Thus overall accuracy is computed as $Accuracy = \sum_{i=1}^{m} n_{ii}/N$, where N is the total number of samples, *Accuracy by class* $= n_{ii}/n_{i+}$ and the geometric mean as *g-mean* $= (\prod_{i=1}^{m} n_{ii}/n_{i+})^{\frac{1}{m}}$.

Table 1. Number of training and testing samples in each class

New Classes	Training	Test	Original Classes	%
C-01	2689	299	1,3,6,7,10	49.64
C-02	264	29	2	4.87
C-03	2055	228	8,9,11	37.93
C-04	290	32	4	5.35
C-05	120	13	5	2.21

Both MLP as Cost-MLP classifiers were trained using each original and preprocessed training data set by different editing techniques.

Table 2. Confusion matrix for a multi-class problem

	Real Classes				
Predicted Classes	1	2	$\cdots$	m	total (n_{i+})
1	n_{11}	n_{12}	$\cdots$	n_{1m}	n_{1+}
2	n_{21}	n_{22}	$\cdots$	n_{2m}	n_{2+}
$\vdots$	$\vdots$	$\vdots$		$\vdots$	$\vdots$
m	n_{m1}	n_{m2}	$\cdots$	n_{mm}	n_{m+}
total (n_{+j})	n_{+1}	n_{+2}	$\cdots$	n_{+m}	N

4 Results and discussion

Several experiments with MCayo database were developed in the experimental process. 'Cost-MLP' denotes the balanced MSE cost function with MLP and 'TS edited' is the training set edited.

Table 3 shows the overall accuracy and the *g-mean* obtained with the approaches previously described. We can observe that the classification accuracy is high and the *g-mean* is low. So, the minority samples are missclassified while the samples of majority classes are well identified. When the original data set is classified with Cost-MLP both performance measures are improved.

On other hand, when the editing techniques are employed, the *g-mean* is improved than the original training data set (without preprocessing). The classification results obtained from the joint application of editing techniques and Cost-MLP outperform the *g-mean* with respect to apply the two techniques separately.

We observe that the RUS algorithm, although shows a slightly improvement, the editing techniques appears as the best strategies. Analyzing the percentage of reduction, higher values are obtained for the editing techniques that obtain better *g-mean*.

Table 3. Experimental results by editing the majority classes

MLP	Original	ENN	NCNE	RNGE	GGE	RUS
Accuracy	83.27(1.20)	85.50(1.43)	84.65(2.00)	84.96(1.29)	85.55(1.34)	84.95(1.46)
g-mean	00.00(0.00)	43.18(27.27)	65.19(12.15)	22.85(26.84)	47.06(25.94)	37.43(26.01)
Cost-MLP	Original	ENN	NCNE	RNGE	GGE	RUS
Accuracy	86.40(1.06)	83.37(2.32)	82.60(2.57)	84.64(1.84)	83.59(2.27)	86.25(1.21)
g-mean	69.80(3.14)	81.14(4.26)	82.26(3.73)	77.10(5.51)	82.05(4.42)	71.97(3.90)
% reduction	00.00(0.00)	25.00(0.46)	33.00(0.55)	19.00(0.19)	28.00(0.33)	33.00(0.08)

In Table 4 we can see the results of editing techniques and Cost-MLP for each class. The two first columns indicate the strategy applied and the number class. The third column we show the proportion of class elements in relation with the total samples ($ratio = n_i/N$, where n_i is the elements number of class i and N the total samples in the TS). The fourth column is the classification accuracy and the last one shows the classes with the level of confusion which is greater than 10% (the percentage of confusion appears in brackets).

In table 4, we can observe that the editing techniques reduce the confusion level among classes. For example, Class 2 is overlapped with the class 1. When it is applied the editing techniques, the confusion level is diminished. In this case, the NCNE obtains a better performance.

5 Conclusion

In this paper, we analyze how to deal the class overlap and class imbalance in a multi–classifcation problem. The goal was study the performance of these two techniques combined: editing techniques joint balanced MSE cost function with MLP.

The experiments show that the benefits associated to inclusion a balanced MSE cost function in the training process. However, this is not enough for reducing the overlapping among classes. For that, using the edition strategies, we can reduce the overlapping problem increasing the prediction of the minority classes. In this paper, the use of both

Table 4. Performance on each class with the Cost-MLP

	Class	Ratio	Accuracy	% confusion ($>$ 10 %)
Original	C-01	0.49	88.32	
	C-02	0.05	51.67	C-01 (48.26)
	C-03	0.38	93.69	
	C-04	0.06	61.76	C-01 (34.25)
	C-05	0.02	63.91	C-01 (34.14)
ENN	C-01	0.49	76.50	
	C-02	0.05	88.50	C-01 (11.16)
	C-03	0.38	93.37	
	C-04	0.06	74.61	C-01 (15.91)
	C-05	0.02	76.62	C-01 (21.73)
NCNE	C-01	0.49	74.87	
	C-02	0.05	91.40	
	C-03	0.38	92.53	
	C-04	0.06	76.91	C-01 (15.85)
	C-05	0.02	79.25	C-01 (19.17)
RNGE	C-01	0.49	81.53	
	C-02	0.05	75.39	C-01 (24.37)
	C-03	0.38	93.23	
	C-04	0.06	67.86	C-01 (24.61)
	C-05	0.02	72.93	C-01 (25.26)
GGE	C-01	0.49	76.59	
	C-02	0.05	91.60	
	C-03	0.38	93.34	
	C-04	0.06	74.88	C-01 (17.53)
	C-05	0.02	77.59	C-01 (20.30)
RUS	C-01	0.49	87.35	
	C-02	0.05	55.02	C-01 (44.98)
	C-03	0.38	93.57	
	C-04	0.06	62.68	C-01 (33.39)
	C-05	0.02	70.08	C-01 (28.27)

techniques is the best option to reduce the classification error and solve these kind of problems.

Future works will be addressed to investigate the potential of these editing methods applied in the hidden space of the neural network. This involves working in the space of the hidden layer and not in the feature space, such as commonly happens with the Wilson's editing and its variants.

References

1. Anand, R., Mehrotra, K., Mohan, C., Ranka, S.: An improved algorithm for neural network classification of imbalanced training sets. IEEE Transactions on Neural Networks **4** (1993) 962–969
2. Ou, G., Murphey, Y.L.: Multi-class pattern classification using neural networks. Pattern Recognition **40**(1) (2007) 4–18
3. Al-Haddad, L., Morris, C.W., Boddy, L.: Training radial basis function neural networks: effects of training set size and imbalanced training sets. Journal of Microbiological Methods **43**(1) (2000) 33 – 44
4. He, H., Garcia, E.: Learning from imbalanced data. IEEE Transactions on Knowledge and Data Engineering In Knowledge and Data Engineering **21**(9) (2009) 1263–1284

5. Huang, Y.M., Hung, C.M., Jiau, H.C.: Evaluation of neural networks and data mining methods on a credit assessment task for class imbalance problem. Nonlinear Analysis: Real World Applications **7**(4) (2006) 720–747
6. Mazurowski, M.A., Habas, P.A., Zurada, J.M., Lo, J.Y., Baker, J.A., Tourassi, G.D.: Training neural network classifiers for medical decision making: The effects of imbalanced datasets on classification performance. Neural Networks **21**(2-3) (2008) 427–436
7. Zhou, Z.H., Liu, X.Y.: Training cost-sensitive neural networks with methods addressing the class imbalance problem. IEEE Transactions on Knowledge and Data Engineering. **18** (2006) 63–77
8. Chawla, N.V., Bowyer, K.W., Hall, L.O., Kegelmeyer, W.P.: Smote: Synthetic minority oversampling technique. Journal of Artificial Intelligence Research **16** (2002) 321–357
9. García, S., Herrera, F.: Evolutionary undersampling for classification with imbalanced datasets: Proposals and taxonomy. Evolutionary Computation **17** (2009) 275–306
10. Anand, R., Mehrotra, K., Mohan, C., Ranka, S.: Efficient classification for multiclass problems using modular neural networks. Neural Networks, IEEE Transactions on **6**(1) (1995) 117–124
11. Bruzzone, L., Serpico, S.: Classification of imbalanced remote-sensing data by neural networks. Pattern Recognition Letters **18** (1997) 1323–1328
12. García, V., Mollineda, R.A., Sánchez, J.S.: On the k-nn performance in a challenging scenario of imbalance and overlapping. Pattern Analysis and Applications **11**(3) (September 2008) 269–280
13. Prati, R., Batista, G., Monard, M.: Class imbalances versus class overlapping: An analysis of a learning system behavior. In: MICAI. (2004) 312–321
14. Visa, S., Ralescu, A.: Learning imbalanced and overlapping classes using fuzzy sets. In: Workshop on Learning from Imbalanced Datasets(ICML0́3). (2003) 91–104
15. Wilson, D.: Asymptotic properties of nearest neighbor rules using edited data. IEEE Transactions on Systems, Man and Cybernetics **2**(4) (1972) 408–420
16. Chaudhuri, B.B.: A new definition of neighborhood of a point in multi-dimensional space. Pattern Recognition Letters **17**(1) (1996) 11–17
17. Sánchez, J.S., Pla, F., Ferri, F.J.: On the use of neighbourhood-based non-parametric classifiers. Pattern Recognition Letters **18**(11-13) (1997) 1179–1186
18. Jaromczyk, J., Toussaint, G.: Relative neighborhood graphs and their relatives. Proceedings of the IEEE **80**(9) (1992) 1502–1517
19. Sánchez, J.S., Pla, F., Ferri, F.J.: Prototype selection for the nearest neighbour rule through proximity graphs. Pattern Recognition Letters **18**(6) (1997) 507–513