Jesús Ariel Carrasco-Ochoa
José Francisco Martínez-Trinidad
Josef Kittler (Eds.)

LNCS 6256

Advances in Pattern Recognition

Second Mexican Conference on
Pattern Recognition, MCPR 2010
Puebla, Mexico, September 2010, Proceedings

Springer

Lecture Notes in Computer Science 6256

Commenced Publication in 1973
Founding and Former Series Editors:
Gerhard Goos, Juris Hartmanis, and Jan van Leeuwen

Jesús Ariel Carrasco-Ochoa
José Francisco Martínez-Trinidad
Josef Kittler (Eds.)

Advances in Pattern Recognition

Second Mexican Conference on
Pattern Recognition, MCPR 2010
Puebla, Mexico, September 27-29, 2010
Proceedings

Springer

Volume Editors

Jesús Ariel Carrasco-Ochoa
José Francisco Martínez-Trinidad
National Institute of Astrophysics, Optics and Electronics (INAOE)
Computer Science Department, Luis Enrique Erro No. 1
72840 Sta. Maria Tonantzintla, Puebla, Mexico
E-mail: {ariel; fmartine}@inaoep.mx

Josef Kittler
University of Surrey
Centre for Vision, Speech and Signal Processing
School of Electronics and Physical Sciences
Guildford GU2 7XH, United Kingdom
E-mail: j.kittler@surrey.ac.uk

Library of Congress Control Number: Applied for

CR Subject Classification (1998): I.2, I.4, I.5, H.3, F.1, H.4

LNCS Sublibrary: SL 6 – Image Processing, Computer Vision, Pattern Recognition, and Graphics

ISSN 0302-9743
ISBN-10 3-642-15991-5 Springer Berlin Heidelberg New York
ISBN-13 978-3-642-15991-6 Springer Berlin Heidelberg New York

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India
Printed on acid-free paper 06/3180

Pattern Recognition and Data Mining

Cost-Sensitive Neural Networks and Editing Techniques for Imbalance Problems

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

[1] Institute of New Imaging Technologies
Dept. Llenguatges i Sistemes Informátics, Universitat Jaume I
Av. Sos Baynat s/n, 12071 Castelló de la Plana (Spain)
[2] 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 multi-class imbalance problem in supervised pattern recognition methods is receiving growing attention. Imbalanced datasets means that some classes are represented by a large number of samples while the others classes only contain a few. In real-world applications, imbalanced training sets may produce an important deterioration of the classifier performance when neural networks are applied in the classes less represented. In this paper we propose training cost-sentitive neural networks with editing techniques for handling the class imbalance problem on multi-class datasets. The aim is to remove majority samples while compensating the class imbalance during the training process. Experiments with real data sets demonstrate the effectiveness of the strategy here proposed.

Keywords: Multi-class imbalance; backpropagation; cost function; editing.

1 Introduction

Neural networks have become a popular tool in Pattern Recognition, Machine Learning and Data Mining [1]. Although there are several kinds of neural networks, most attention has been focused on the use of Multilayer Perceptron (MLP) [2], or feed-forward networks trained with a backpropagation learning algorithm for supervised classification.

However, it is well known that in MLP, the nature of the Training Data Sets (TDS) has a major impact on the ability of the network to generalize[2]. One of the problems in the complexity of the TDS that most affects the neural networks is the class imbalance [3].

A two-class data set is said to be imbalanced when one of the classes (the minority one) is heavily under-represented with regard to the other class (the majority one) [4]. This problem is encountered in a large number of domains, and in certain cases, it has been observed that class imbalance may cause a significant deterioration in the performance attainable by standard learners because these are often biased towards the majority class [5].

Many works have addressed the class imbalance problem [6,5]. The most popular strategies for dealing with this problem can be grouped in three categories. One is to assign different costs to the classification errors [3]. The second technique is to

J.A. Carrasco-Ochoa et al. (Eds.): MCPR 2010, LNCS 6256, pp. 180–188, 2010.

make a resampling of the original TDS, over-sampling the minority class and/or under-sampling the majority class until the classes are approximately equally represented [4]. The third technique consists in internally biasing the discrimination-based process and compensate the class imbalance [7,8]. However, these techniques do not consider other complexities that might result from TDS. In this regard, several studies suggest that other problems such as the overlap between classes should be taken into account in classification tasks [9,10].

In this work, we present some preliminary results to explore two issues related with the Multi-Class Imbalance Problem. Initially, we remove majority samples from the overlap region, producing a local balance of the classes. For this, the only requirement is that all samples of the minority classes must be saved in the TDS. As downsizing of the majority classes can throw away significant information, an editing scheme is applied. Note that a global balance in the class sizes is not achieved. Subsequently, the backpropagation algorithm is modified to avoid that the minority classes be ignored in the learning process, and to accelerate the convergence of the neural network.

2 Multilayer Perceptron

The multilayer perceptron (MLP) neural network [11] 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 layers 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 nonlinear (commonly a sigmoid) activation function to produce a single output.

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

To adjust the weights, the backpropagation 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.

2.1 The Backpropagation Algorithm and the Class Imbalance Problem

Empirical studies of the backpropagation algorithm [12] show that class imbalance problem generates unequal contributions to the mean square error (MSE) in the training phase. Clearly the major contribution to the MSE is produced by the majority class.

Let us consider a TDS with two classes ($m = 2$) such that $N = \sum_i^m n_i$ and n_i is the number of samples from class i. Suppose that the MSE by class can be expressed as

$$E_i(U) = \frac{1}{N} \sum_{n=1}^{n_i} \sum_{p=1}^{L} (d_p^n - y_p^n)^2 \,, \tag{1}$$

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

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

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

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

$$\begin{aligned} E(U) &= \textstyle\sum_{i=1}^{m} \gamma(i) E_i = \gamma(1)E_1(U) + \gamma(2)E_2(U) \\ &= \tfrac{1}{N} \textstyle\sum_{i=1}^{m} \gamma(i) \sum_{n=1}^{n_i} \sum_{p=1}^{L} (y_p^n - F_p^n)^2\,, \end{aligned} \tag{3}$$

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, the cost function 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 a cost function is included in the training process, the data probability distribution is altered [13]. However, this cost function (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 $\gamma(m)$ reduces its effect on the data distribution probability.

3 Edited Nearest Neighbor Rule

Wilson [14] developed the Edited Nearest Neighbor (ENN) algorithm in which the set of samples **S** starts out the same as TDS, and then each instance of the set **S** is removed if it does not agree with the majority of its k nearest neighbors (with k=3, typically). This method removes noisy instances as well as samples at the borderline, leaving 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 **S** if it is misclassified using the k-NN rule with prototypes in $\mathbf{X} - \{\mathbf{x}_i\}$.

In this work, the ENN is applied only in the majority classes. The aim is to reduce the complexity in the overlap region maintaining all the minority samples. This technique can be seen as focused under-sampling.

4 Methodology

The experiments were carried out on three images real data sets (Cayo, Feltwell and Satimage). A brief summary is given in the Table 1. 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 databases

Dataset	Size	Attr.	Class	Class distribution
Cayo	6019	4	11	838/293/624/322/133/369/324/722/789/833/772
Feltwell	10944	15	5	3531/2441/896/2295/1781
Satimage	6430	36	6	1508/1531/703/1356/625/707

The *Accuracy* and *g-mean* are used as performance measure to evaluate the classifier. It is common to obtain measure criteria from the confusion matrix where real classes are in columns, whereas predicted ones appear in rows (Table 2). The table built in this way is a general vision assignment, where diagonal elements count the correctly assigned samples and elements out of the diagonal count the wrongly classified ones.

From the confusion matrix, we can define

$$Accuracy = \sum_{i=1}^{m} n_{ii}/N\,, \tag{5}$$

where N is the total number of samples.

$$Accuracy\ by\ class = n_{ii}/n_{i+}. \tag{6}$$

Other measure used is the geometric mean (*g-mean*) defined as

$$g\text{-}mean = (\prod_{i=1}^{m} n_{ii}/n_{i+})^{\frac{1}{m}}\,. \tag{7}$$

All the MLP were trained with the backpropagation algorithm in batch mode. This process has been repeated ten times and the results correspond to the average. The learning rate (η) was set to 0.1 and only one hidden layer was used. The number of neurons for the hidden layer was established to 7, 6 and 12 for Cayo, Feltwell and Satimage datasets respectively.

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

Summarizing the strategy proposed in this work consists of the following:

1. To edit the TDS with the ENN technique, removing only majority samples in the overlap region and producing a local balance of the classes (sec. 3).
2. To modify the backpropagation algorithm applying a cost-function (Eq. 4) to avoid that the minority classes would be ignored in the learning process, and accelerating the convergence of the neural network.
3. To train the MLP with the modified algorithm over the TDS edited.

5 Results and Discussion

In order to evaluate the possibilities of the proposed approaches here exposed, several experiments with imbalance data sets were developed. In Tables 3, 4, 5 and 6 the main results are detailed. In these experiments, we denote "Cost-MLP" the cost function with MLP and "TDS edited" the imbalanced training set edited.

Table 3 shows the percentage of samples eliminated after applying the edition algorithm in the majority classes. In the case of Cayo database, the classes 1, 3, 8, 9, 10 and 11 were considered as majority classes. On the other hand, in Feltwell database, only the class 3 was identified like minority class. For Satimage database, the classes 1, 2 and 4 were considered as majority classes. The experiments used different values of k in the edition process choosing the most suitable for each database: Cayo $k = 15$, Feltwell $k = 9$ and Satimage $k = 5$.

In the case of majority classes, the number of samples eliminated were significant (see Table 3). This important reduction of the size tends to improve the classification accuracy in the minority classes. On the other hand, it is possible observe that in some majority classes, the number of samples eliminated was minimum: classes 1 and 8 for Cayo, classes 2, 4 and 5 for Feltwell and class 2 for Satimage.

The information presented in Tables 4, 5 and 6 was organized as follows. The first column of each table indicates the strategy applied, i.e., if the TDS were edited or not, or if we use the modified algorithm or the standard algorithm. The second column indicates the class to which the results correspond. In the third column (the ratio), 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 TDS). The fourth column is the classification accuracy and the last one shows the classes with the level of confusion is greater thant 10% (the percentage of confusion appears in brackets).

Table 3. Percentage of samples eliminated after editing the TDS

Class	1	2	3	4	5	6	7	8	9	10	11	Total reduction
Cayo	10.74	0.00	43.73	0.00	0.00	0.00	0.00	5.54	13.42	48.92	48.19	21.40%
Feltwell	15.26	10.93	0.00	11.2	13.14							11.89%
Satimage	47.5	12.69	0.00	60.56	0.00	0.00						27.31%

Table 4. Results of the classification phase with the MLP on the Cayo data base

	Class	Ratio	Accuracy	% confusion (> 10 %)
MLP + TDS	C-01	0.14	89.74	
	C-02	0.05	51.20	C-03 (48.63)
	C-03	0.10	95.69	
	C-04	0.05	70.99	C-03 (12.61) C-08 (11.43)
	C-05	0.02	19.92	C-01 (50.30) C-03 (19.39)
	C-06	0.06	56.44	C-07 (31.90)
	C-07	0.05	95.40	
	C-08	0.12	98.55	
	C-09	0.13	87.56	C-10 (12.44)
	C-10	0.14	77.03	C-11 (21.80)
	C-11	0.13	89.40	C-10 (10.14)
Cost-MLP + TDS	C-01	0.14	88.10	
	C-02	0.05	51.37	C-03 (48.63)
	C-03	0.10	93.42	
	C-04	0.05	93.54	
	C-05	0.02	73.79	C-01 (14.39) C-03 (11.52)
	C-06	0.06	60.43	C-07 (30.82)
	C-07	0.05	95.31	
	C-08	0.12	94.86	
	C-09	0.13	87.56	C-10 (12.44)
	C-10	0.14	76.36	C-11 (22.99)
	C-11	0.13	91.89	
MLP + TDS edited	C-01	0.14	88.15	
	C-02	0.05	51.99	C-03 (48.01)
	C-03	0.10	92.84	
	C-04	0.05	91.43	
	C-05	0.02	51.97	C-01 (25.23) C-03 (15.45)
	C-06	0.06	58.99	C-07 (31.90)
	C-07	0.05	96.29	
	C-08	0.12	97.60	
	C-09	0.13	87.56	C-10 (12.44)
	C-10	0.14	76.56	C-11 (15.06)
	C-11	0.13	75.41	C-09 (16.02)
Cost-MLP + TDS edited	C-01	0.14	86.87	
	C-02	0.05	70.72	C-03 (29.28)
	C-03	0.10	78.31	C-02 (14.79)
	C-04	0.05	94.22	
	C-05	0.02	86.67	
	C-06	0.06	60.95	C-07 (31.39)
	C-07	0.05	95.74	
	C-08	0.12	95.24	
	C-09	0.13	87.56	C-10 (12.44)
	C-10	0.14	73.94	C-11 (24.63)
	C-11	0.13	94.70	

In Table 4, we observe in Cayo dataset that the classes 2, 5 and 6 are affected seriously by the imbalance problem. We point out that when the imbalance is compensated with the cost function, the accuracy of the minority classes is increased (except for the class 2) especially in the case of class 5.

When the TDS is edited, the global accuracy and the performance of the minority classes are improved. Nevertheless, in the case of overlapped classes (see class 2 in Table 4) the results presented are practically the same.

When the classes imbalance is compensated and the network is trained with the TDS edited, the accuracy of the class 2 increases significantly. On the other hand, the combination of both strategies improves the rate of recognition on the minority classes. The classes 6 and 7 do not increase your performance due to these classes are overlapped each other.

Table 5. Feltwell: Classification with MLP

	Class	Ratio	Accuracy	% confusion (> 10 %)
MLP + TDS	C-01	0.35	99.07	
	C-02	0.24	81.97	C-03 (11.72)
	C-03	0.10	78.86	C-01 (10.70)
	C-04	0.15	83.91	C-01 (11.48)
	C-05	0.17	90.43	
Cost-MLP + TDS	C-01	0.35	98.58	
	C-02	0.24	80.85	C-03 (14.85)
	C-03	0.10	83.08	
	C-04	0.15	83.35	C-01 (10.74)
	C-05	0.17	88.92	C-01 (10.55)
MLP + TDS edited	C-01	0.35	97.63	
	C-02	0.24	73.62	C-03 (13.99) C-05 (11.17)
	C-03	0.10	81.48	C-04 (10.09)
	C-04	0.15	83.19	
	C-05	0.17	96.12	
Cost-MLP + TDS edited	C-01	0.35	97.45	
	C-02	0.24	69.70	C-03 (23.85)
	C-03	0.10	84.70	
	C-04	0.15	81.80	
	C-05	0.17	95.76	

Table 6. Satimage: Classification results with the MLP

	Class	Ratio	Accuracy	% confusion (> 10 %)
MLP + TDS	C-01	0.23	90.87	
	C-02	0.23	98.83	
	C-03	0.11	90.71	
	C-04	0.20	97.71	
	C-05	0.11	2.37	C-01 (61.04) C-04 (33.03)
	C-06	0.12	70.25	C-01 (15.74)
Cost-MLP + TDS	C-01	0.23	81.89	C-05 (13.83)
	C-02	0.23	97.51	
	C-03	0.11	90.54	
	C-04	0.20	91.61	
	C-05	0.11	65.73	C-01 (19.91) C-04 (13.22)
	C-06	0.12	76.71	C-01 (13.50)
MLP + TDS edited	C-01	0.23	75.21	C-05 (18.66)
	C-02	0.23	98.18	
	C-03	0.11	91.43	
	C-04	0.20	88.46	C-05 (10.18)
	C-05	0.11	60.95	C-01 (25.45) C-04 (11.04)
	C-06	0.12	77.05	
Cost-MLP + TDS edited	C-01	0.23	71.47	C-05 (23.09)
	C-02	0.23	96.83	
	C-03	0.11	93.39	
	C-04	0.20	83.27	C-05 (15.57)
	C-05	0.11	84.08	
	C-06	0.12	81.14	

The results of Feltwell database are included in Table 5. The use of the TDS edited improves the classifier effectiveness on the minority class 3. However, there is a tendency for reducing the network effectiveness on the majority classes, especially class 2.

Satimage database (see Table 6) shows a similar tendency to Cayo and Feltwell. When the TDS is edited, the classification on the minority classes is increased. The combination of both approaches increases the accuracy of minority classes and compensate the classes imbalance. This strategy qualitatively improves the accuracy of the minority classes. For example, in class 5 the accuracy is 65.73% with the original TDS,

Table 7. Global performance of the classifier

Cayo	MLP TDS	Cost-MLP TDS	MLP TDS edited	Cost-MLP TDS edited
Accuracy	83.58(0.77)	85.15(0.27)	82.75(1.90)	84.79(0.48)
g-mean	70.17(6.28)	80.96(0.44)	76.50(2.92)	83.30(0.78)

Feltwell	MLP TDS	Cost-MLP TDS	MLP TDS edited	Cost-MLP TDS edited
Accuracy	89.38(0.95)	89.01(0.51)	87.99(1.21)	87.04(0.71)
g-mean	86.60(1.64)	86.79(0.75)	85.97(1.47)	85.33(0.98)

Satimage	MLP TDS	Cost-MLP TDS	MLP TDS edited	Cost-MLP TDS edited
Accuracy	82.26(0.31)	86.07(0.34)	83.66(0.34)	84.59(0.38)
g-mean	47.31(5.72)	83.34(0.95)	80.90(1.17)	84.70(0.36)

whereas when the network is trained with the TDS edited and with the modified algorithm its value reaches 84.08%.

On the other hand, analyzing the global values of accuracy and geometric mean (see Table 7), we can see that these measures obtain better results when the TDS is edited, even in cases where the imbalance is not compensated.

In Feltwell database, it is possible that the proposed strategy does not represent a significant improvement. Only when we apply cost-functions in training neural network, the results are similar to original dataset. The editing technique proposed obtains clearly worse results and it is not adequate for this database.

Summarizing we can say that the editing of TDS and the application of cost functions in the neural network training reduces the confusion between classes. However, when we give priority to minority classes, the majority classes are affected in the training process with a loss of accuracy in these classes.

6 Conclusion

In this work we propose a strategy based on combination of training cost-functions with editing technique in neural networks to deal with the class imbalance problem on multi-class datasets. This generates two effects: a) to compensate the class imbalance during the training process and b) 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 modification of the training algorithm including a cost function increases the recognition rate of less represented classes, accelerating the convergence of the network.

However, we have seen in some situations that the proposed editing technique has not been adequate. Thus, it is interesting the use of new strategies to reduce the confusion region taking into account both the imbalance and the representativeness of the data.

Acknowledgment

This work has been partially supported by the Spanish Ministry of Science and Education under project CSD2007-00018, UAEMCA-114 and SBI112 from the Mexican SEP, the 2703/2008U from the UAEM project and P1.1B2009-45 (Fundació Caixa-Castelló).

References

1. Jain, A., Mao, J., Mohiuddin, K.: Artificial neural networks: A tutorial. Computer 29(3), 31–44 (1996)
2. Foody, G.: The significance of border training patterns in classification by a feedforward neural network using back propagation learning. International Journal of Remote Sensing 20(18), 3549–3562 (1999)
3. 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, 63–77 (2006)
4. Japkowicz, N., Stephen, S.: The class imbalance problem: a systematic study. Intelligent Data Analysis 6, 429–449 (2002)
5. He, H., Garcia, E.: Learning from imbalanced data. IEEE Trans. on Knowl. and Data Eng. 21(9), 1263–1284 (2009)
6. Visa, S.: Issues in mining imbalanced data sets - a review paper. In: Artificial Intelligence and Cognitive Science Conference, pp. 67–73 (2005)
7. Anand, R., Mehrotra, K., Mohan, C., Ranka, S.: Efficient classification for multiclass problems using modular neural networks. IEEE Transactions on Neural Networks 6(1), 117–124 (1995)
8. Bruzzone, L., Serpico, S.: Classification of imbalanced remote-sensing data by neural networks. Pattern Recognition Letters 18, 1323–1328 (1997)
9. Visa, S., Ralescu, A.: Learning imbalanced and overlapping classes using fuzzy sets. In: Workshop on Learning from Imbalanced Datasets(ICML'03), pp. 91–104 (2003)
10. Prati, R., Batista, G., Monard, M.: Class imbalances versus class overlapping: An analysis of a learning system behavior. In: Monroy, R., Arroyo-Figueroa, G., Sucar, L.E., Sossa, H. (eds.) MICAI 2004. LNCS (LNAI), vol. 2972, pp. 312–321. Springer, Heidelberg (2004)
11. Bishop, C.M.: Neural Networks for Pattern Recognition, January 1996. Oxford University Press, USA (1996)
12. 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, 962–969 (1993)
13. Lawrence, S., Burns, I., Back, A., Tsoi, A., Giles, C.L.: Neural network classification and unequal prior class probabilities. In: Orr, G.B., Müller, K.-R. (eds.) NIPS-WS 1996. LNCS, vol. 1524, pp. 299–314. Springer, Heidelberg (1998)
14. Wilson, D.R., Martinez, T.R.: Reduction techniques for instance-based learning algorithms. Machine Learning 38(3), 257–286 (2000)