Reward-Punishment Editing for Mixed Data*

Raúl Rodríguez-Colín, J.A. Carrasco-Ochoa, and J.Fco. Martínez-Trinidad

National Institute for Astrophysics, Optics and Electronics,
Luis Enrique Erro No. 1 Sta. Ma. Tonantzintla, Puebla, México C. P. 72840
{raulrc, ariel, fmartine}@inaoep.mx

Abstract. The KNN rule has been widely used in many pattern recognition problems, but it is sensible to noisy data within the training set, therefore, several sample edition methods have been developed in order to solve this problem. A. Franco, D. Maltoni and L. Nanni proposed the Reward-Punishment Editing method in 2004 for editing numerical databases, but it has the problem that the selected prototypes could belong neither to the sample nor to the universe. In this work, we propose a modification based on selecting the prototypes from the training set. To do this selection, we propose the use of the Fuzzy C-means algorithm for mixed data and the KNN rule with similarity functions. Tests with different databases were made and the results were compared against the original Reward-Punishment Editing and the whole set (without any edition).

1 Introduction

The k-nearest neighbor rule (KNN) has been widely used in many pattern recognition problems. Given a set of n training objects, when a new object is going to be classified, the KNN rule identifies the k nearest neighbors in the training set and the new object is labeled with the most frequent class among the k nearest neighbors.

However, some of the data in the training set do not provide useful information to classify new objects; therefore, it is necessary to edit the sample in order to get a better training set which would contribute to obtain better classification rates. In the sample edition area, several methods have been developed [1-3].

The Reward-Punishment Editing method (R-P Editing) is based on two selection criteria: A local criterion rewards each pattern that contributes to classify its neighbors correctly (using the KNN rule), and punish the others; the second criterion rewards each pattern that is classified correctly (using the KNN rule) with a set of prototypes extracted from the training set. Based on these criteria, a weight is assigned to each pattern in the training set. If the weight is smaller than a predefined threshold, the pattern is eliminated from the training set.

In order to select prototypes, the Reward-Punishment Editing method uses the Fuzzy C-means algorithm. This does not guarantee that the selected prototypes belong to the sample, because the prototypes in the classical Fuzzy C-means are computed as the mean of the cluster. Therefore, we propose to use the Fuzzy C-means for mixed

* This work was financially supported by CONACyT (Mexico) through the project J38707-A.

M. Lazo and A. Sanfeliu (Eds.): CIARP 2005, LNCS 3773, pp. 481 – 488, 2005.

data, which guarantees that the selected prototypes belong to the sample. In addition, using the KNN rule with similarity functions allows working with object descriptions through qualitative and quantitative features.

This paper is organized as follows: in section 2 a description of the most similar neighbor method (KNN with similarity functions) is presented, in section 3 the Fuzzy C-means algorithm for mixed data is described, in section 4 the R-P Editing for mixed data algorithm is introduced, in section 5 the obtained results are reported. Finally, in section 6 some conclusions are given.

2 The Most Similar Neighbor

When we talk about the KNN rule with similarity functions, we are talking about the k-most similar neighbor (K-MSN). For that reason, we have to define a similarity comparison function for comparing feature values and establishing its similarity moreover, it is needed to define a similarity function for comparing objects in the data set.

Let us consider a set of n objects $\{O_1,O_2,\ldots,O_n\}$, each object in this set is described by a set $R = \{x_1,\ldots,x_m\}$ of features. Each feature x_i takes values in a set D_i, $x_i(O) \in D_i$, i=1,...,m. Thus, features could be qualitative or quantitative.

For each feature x_i, i=1,...,m, we define a comparison function C_i:$D_i \times D_i \rightarrow L_i$ with i=1,2,…,m. where L_i is a totally ordered set such that C_i gives us the similarity between two values of the feature x_i, for i=1,…,m.

Based on the C_i it is possible define a similarity function between objects.

Let Γ: $(D_1 \times \ldots \times D_m)^2 \rightarrow [0,1]$ be a similarity function. $\Gamma(O_j,O_k)$ gives the similarity between O_j and O_k, and satisfies:

$\Gamma(O_j,O_k) \in [0,1]$ for $1 \le j \le n$, $1 \le k \le n$;
$\Gamma(O_j,O_j) = 1$for $1 \le j \le n$;
$\Gamma(O_j,O_k) = \Gamma(O_k,O_j)$for $1 \le j \le n$, $1 \le k \le n$;
$\Gamma(O_i,O_j) > \Gamma(O_i,O_k)$means that O_i is more similar to O_j than to O_k

In this work, we used the following similarity functions.

$$\Gamma(O_i, O_j) = \frac{\left| \left\{ x \in R \wedge C\left(x(O_i), x(O_j)\right) = 1 \right\} \right|}{m} \tag{1}$$

Where the comparison functions used in this work are:
For qualitative data:

$$C(x(O_i), x(O_j)) = \begin{cases} 1 & \text{if } x(O_i) = x(O_j) \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

For quantitative data:

$$C(x(O_i), x(O_j)) = \begin{cases} 1 & \text{if } \left| x(O_i) - x(O_j) \right| < \varepsilon \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

The value for ε is introduced by the user based on the data set.

Based on the definitions described above, we can work on databases with numerical information, described in terms of qualitative and quantitative features.

The K-MSN is similar to the KNN rule, but the K-MSN identifies the k most similar neighbors of the new object in the training set, after that, the new object is labeled with the most frequent class among the k most similar neighbors.

3 Fuzzy C-means for Mixed Data

The use of Fuzzy C-means for mixed data allows working with object descriptions in terms of qualitative and quantitative features.

The objective is to obtain fuzzy clusters with the characteristic that the similarity among the objects that belong to the same cluster is high, and at the same time, the similarity among different clusters is low.

In order to obtain this type of clusters, given a group of objects to classify, the Fuzzy C-means for mixed data algorithm randomly selects c objects, which will be the initial representative objects (centers) of the clusters. With the representative objects, the algorithm classifies the rest of the objects in the dataset. After this classification, it calculates the new representative objects. This procedure is repeated until we obtain the same representative objects in two consecutive iterations.

The problem is reduced to optimize the next objective function:

$$J_m(\vartheta) = \sum_{k=1}^{n}\sum_{i=1}^{c} u_{ik}\left(1-\Gamma\left(O_k, O_i^*\right)\right) \tag{4}$$

Where ϑ is a representative object set, one for each cluster M_i, $\Gamma(O_k,O_i^*)$ is the similarity between the object O_k and the representative object O_i^* of M_i and u_{ik} is the membership degree of the object O_k to the cluster M_i. Thus the solution to this problem consists in minimizing $J_m(\vartheta)$. The next formulas are used to calculate u_{ik} and O_i^* respectively.

The degree of membership of the object O_k to M_i is computed via (5).

$$u_{ik} = \frac{1}{\sum_{j=1}^{c}\left[\frac{(1-\Gamma(O_k, O_i^*))}{(1-\Gamma(O_k, O_j^*))}\right]^2} \tag{5}$$

Finally, we use (5) to calculate the representative objects for the clusters M_i, $i=1,\ldots,c$.

$$O_i^* = \min_{q \in M_{i\alpha}} \left\{\sum_{k=1}^{n} u_{ik}\left(1-\Gamma\left(O_k, O_q\right)\right)\right\} \tag{6}$$

Where

$$M_{i_\alpha} = \left\{ O_k \;\middle|\; u_{M_i}(O_k) = \max_{j=1,\ldots,c} \left\{ u_{M_j}(O_k) \right\} \right\} \tag{7}$$

4 Reward-Punishment Editing for Mixed Data

The R-P Editing for mixed data is similar to the method proposed in [3] but using the k-most similar neighbor rule and the Fuzzy C-means for mixed data algorithm.

The algorithm is divided in two parts, in the first part, the method rewards and punishes patterns in the training set using the k-most similar neighbor (K-MSN) rule. Each pattern is rewarded if it contributes to the correct classification of another pattern in the training set, in this case the weight *WR* is increased, in the same way; a pattern is punished if it contributes to the wrong classification of another pattern, also in the training set, and the weight *WP* is increased. This part of the method is shown in figure 1.

In the second part, the method selects from the sample a prototype set using the Fuzzy C-means for mixed data and applies the K-MSN rule to classify the patterns in the sample, using the selected prototype set like a training set. These selected prototypes belong to the sample.

```
RPEMD(TS, CL)
    WR = WP = WPR = 0
    for each xi ∈ TS
        // Find the k-most similar neighbor of the pattern xi
        [L, c] = K-MSN(xi, TS, k)
        // Is the pattern correctly classified?
        if CL(i) = c then
            // Reward of the patterns that contributed to the correct classification
            for j = 1 to k
                if CL(L(j)) = c then
                    WR(L(j))= WR(L(j))+ 1
        else
            // Punishment of the patterns that contributed to the wrong classification
            for j = 1 to k
                if CL(L(j)) = c then
                    WP(L(j)) = WP(L(j))+ 1
```

Fig. 1. Part 1 of Reward-Punishment Editing for Mixed Data

Each pattern is rewarded if it is classified correctly using the selected prototypes set, that is, the weight *WPR* is increased. In the second part of this algorithm (fig 2) the variable *np_max* determines the number of elements, for each class, in the selected prototype set.

The *WR*, *WP*, *WPR* values are used to determine if a pattern within the training set will be eliminated.

```
for np = 1 to np_max
    //Generation of np prototypes for each class
    PR = CREATEPROTOTYPES(TS, CL, np)
    for pk = 1 to np step 2
        for each xi ∈ TS
            //Classification of each pattern
            [L, c] = K-MSN(xi, PR, pk)
            if CL(i) = c then
                WPR(i) = WPR(i) + 1
NORMALIZE(WP, WR, WPR)
OPTIMIZE(TS, CL, A, B, Γ, et)
// Computation of the final weight and Editing
for each xi ∈ TS
    c = CL(i)
    WF(i) = αc·WR(i) + βc·(1-WP(i)) + γc·WPR(i)
    if WF(i) < et then
        TS = TS - { xi }
```

Fig. 2. Part 2 of Reward-Punishment Editing for Mixed Data

The procedure CREATEPROTOTYPES (*TS*, *CL*, *np*) generates a set of prototypes (*PR*) from the training set. For each class, the Fuzzy C-means for mixed data algorithm is used to determine *np* clusters; the *np* representative objects (these objects belong to the original training set) computed by the Fuzzy C-means for mixed data will be the prototype set. Those patterns that are classified correctly (using the K-MSN) with the selected prototype set are rewarded.

Based on *WR, WP* and *WPR* a final weight (*WF*) is computed, if this final weight is lower than a predefined threshold *et*, the pattern is eliminated from the training set. After that, the objects in the dataset are classified using the K-MSN rule with the edited training set.

5 Experimental Results

In this section, we present the results obtained with the Reward-Punishment Editing for mixed data and compare them against the original algorithm and the whole set (without any edition) results.

In our experiments, the training and test sets were randomly constructed. The average classification accuracy from 10 experiments using 10 fold cross validation were calculated. In each experiment k=3 (for K-MSN) and *np_max*=10 were used. Four datasets taken from [5] were used; the description of these databases is shown in Table 1.

Table 1. Databases used in the experiments

Database	Instances	Features	Classes
Iris	150	4	3
Wine	178	13	3
Credit	690	15	2
Bridges	105	11	6

In table 2, the amounts of quantitative and qualitative features are shown for each database.

Table 2. Quantitative and qualitative features for each used database

Datasets	Quantitative features	Qualitative features
Iris	4	0
Wine	13	0
Credit	6	9
Bridges	4	7

Tests with different thresholds were made. The best results were obtained with *et*=0.2 and *et*=0.3. The obtained results for each datasets are showed in figure 3 and figure 4.

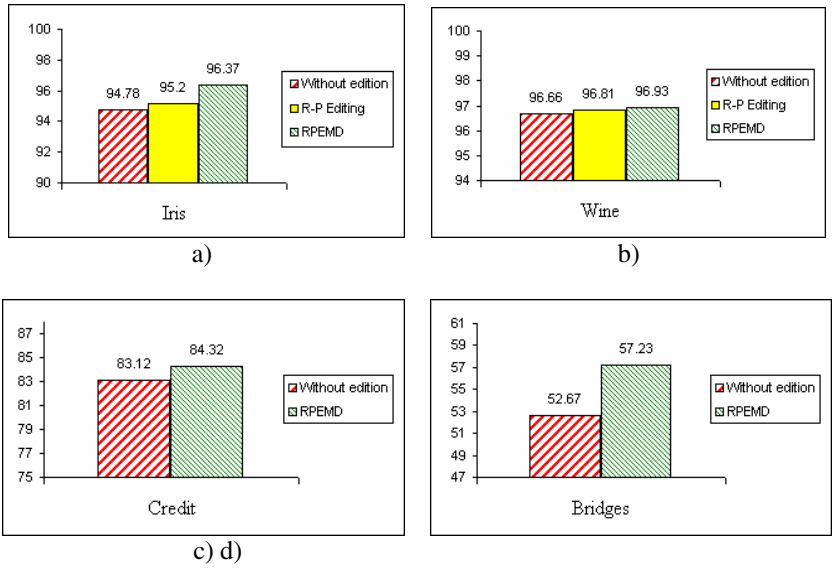

Fig. 3. Classification accuracy on a) Iris, b) Wine, c) Credit and d) Bridges using a threshold *et*=0.3 for editing the training set

Notice that the original R-P Editing could not be applied on Credit and Bridges datasets because they have qualitative features.

In all the experiments, we can see that the classification rates obtained with Reward-Punishment Editing for Mixed Data (RPEMD) are better than the rates obtained with the original R-P Editing method and the whole set without any edition.

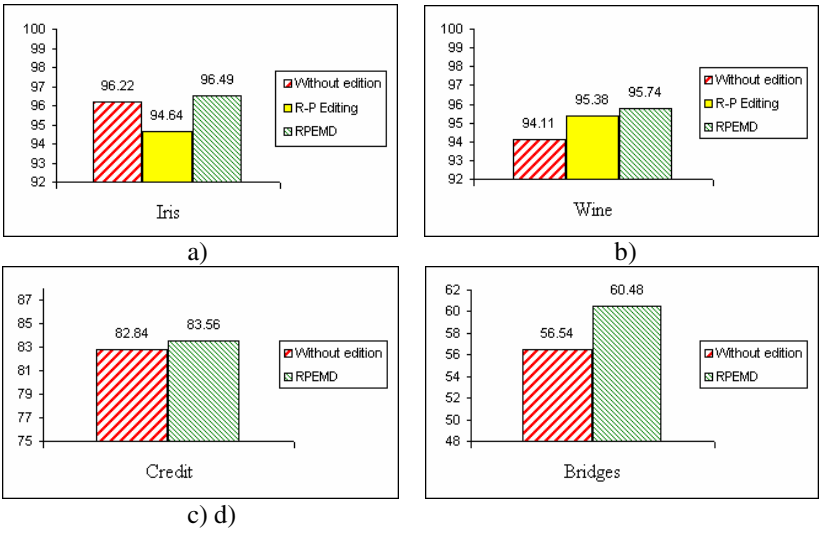

Fig. 4. Classification accuracy on a) Iris, b) Credit, c) Wine and d) Bridges using a threshold *et*=0.2 for editing the training set

Table 3. Size of the Training set after the edition, using *et*=0.2 as threshold of edition

Database	Without Edition	RP-Editing	RPEMD
Iris	100 %	95 %	94 %
Wine	100 %	96 %	93 %
Credit	100 %	-----	95 %
Bridges	100 %	-----	93 %

Table 4. Size of the Training set after the edition, using *et*=0.3 as threshold of edition

Database	Without Edition	RP-Editing	RPEMD
Iris	100 %	94 %	92 %
Wine	100 %	95 %	93 %
Credit	100 %	-----	95 %
Bridges	100 %	-----	92 %

6 Conclusion and Future Work

In supervised classification, the training set quality is very important because it is the basis of the training process. However, in practical cases, there could be irrelevant objects; therefore, it is necessary editing the training sample.

The use of Fuzzy C-means for mixed data and KNN rule with similarity functions in RPEMD allows us to work with object descriptions with mixed data, i.e. quantitative and qualitative features. These characteristics allow applying the new algorithm in many classification problems where the R-P Editing cannot be applied.

The obtained results show that the use of Fuzzy C-means for mixed data and the KNN rule with similarity functions in RPEMD allows getting better accuracy in the classification process.

As future work, we are going to extend the algorithm in order to use other classifiers.

References

1. Wilson, D. Randall and Tony R. Martínez: Reduction Techniques for Instance-Based Learning Algorithms. Machine Learning, Vol. 38. (2000) 257-286.
2. R. Paredes, T. Wagner: Weighting prototypes, a new approach. In the proceedings of International Conference on Pattern Recognition (ICPR), Vol II. (2000) 25-28.
3. A. Franco, D. Maltoni y L. Nanni: Reward- Punishment Editing. In the proceedings of International Conference on Pattern Recognition (ICPR). (2004) (In CD).
4. Irene O. Ayaquica-Martínez and J. Fco. Martínez-Trinidad: Fuzzy C-means algorithm to analyze mixed data. In the proceedings of the 6th Iberoamerican Symposium on Pattern Recognition. Florianópolis, Brazil. (2001) 27-33.
5. C.L. Blake, C.J. Merz: UCI Repository of machine learning databases. [http://www.ics.uci.edu/~mlearn/MLRepository.html] Irvine, CA: University of California, Department of Information and Computer Science. (1998).