A Unified Strategy to Deal with Different Natures of Reject

Harold Mouchère, Eric Anquetil

A Unified Strategy to Deal with Different Natures of Reject

Harold Mouchère[1,*]   Éric Anquetil[2]

IRISA / [1,2]INSA / [1]CNRS

Campus Universitaire de Beaulieu

Avenue du Général Leclerc,

35042 Rennes, France

Harold.Mouchere@irisa.fr

Abstract

The interest of reject for classifier optimization has been shown many times. The diversity of the applications requiring this concept makes us to distinguish two main natures of reject with distinct goals: the confusion reject and the distance reject. After the description of this two kinds of reject, we present a unified formalism to define them using reliability functions and reject thresholds. Then we present a generic algorithm dedicated to the automatic learning of these thresholds. Finally, we compare various possibilities of reject to achieve application goals.

1. Introduction

The reject becomes more and more needed for different classification contexts. We can distinguish two main reject natures: the confusion reject and the distance reject. For example some tasks need a very high accuracy and use the confusion reject option [4, 7] to minimize the risk of error using discriminant knowledge. In another context, the distance reject option using intrinsic knowledge can delimit the validity domain of a specialised recognizer by rejecting patterns belonging to classes which have not been learned [2]. These reject options could be used together or not to define cascading classifiers used to reduce the complexity of the recognition task [10].

In this work we present a general thought about the reject notion. Our aim is to better describe natures of reject and to design a generic and automatic learning process.

Most of works on reject do not focus on the nature of reject but on the error-reject trade-off optimization. One of the most used error-reject trade-off was given by Chow [4]. More recent works [7, 10] point out that this reject rule provides the optimal error-reject trade-off only if the *a posteriori* probabilities are exactly known. So they propose to use

class-related thresholds. Another approach [5] uses reliability functions which allow to define different reject options and to abstract the classifier type. In [6] confusion and distance reject are used on Fuzzy Rules classification without a formalized approach. In these different studies, the reject is learned after the classifier either manually in [6] or automatically: in [5] with techniques using histograms, in [7] with constraints optimization algorithm or in [10] with Particle Swarm Optimization.

These different works use different rejects but without any identification of the reject nature according to the needs of the application. In this study we focus on the interest to define different natures of reject using the notion of reliability functions and the power of multi-thresholds.

The remaining of this paper is structured as follows. In section 2 we show that different reject natures can be defined according to the needs. In section 3 we propose a formalization of the reject notion. Then the unified learning strategy is presented in section 4. Finally the results exposed in section 5 show the importance of taking into account the different natures of reject.

2. Different natures of reject

To define all the aspects of reject, the used classifier must have an enough explicit knowledge modeling. For example, classifiers which have an explicit modeling like Radial Basis Function Network (RBFN) [3], K-Nearest Neighbor or Fuzzy Inference System [6] allow to know if a pattern is near learned knowledge, in addition to class scores. On the contrary, some classifiers like Multi Layer Perceptron are *black boxes* and the only available information is class scores.

The latter presented principles can be applied to all these types of classifiers but their knowledge modeling limits the possible reject natures. Thus, to illustrate the different natures of reject we use here RBFN. Prototypes (RBF) of the

hidden layer are learned using Possibilistic C-Means [9] and define intrinsic properties of each class. The output layer gives class scores which are discriminant properties defining the decision boundaries.

These different natures of knowledge are used to define the reject nature. In the following we describe two main rejects: the confusion and the distance ones.

2.1. The confusion reject

The aim of the confusion reject is to improve the accuracy of the recognizer by rejecting pattern on which the classifier can strongly make a misclassification. These errors are near the decision boundaries because class scores of two (or more) are nearly equal.

We must define a reject zone on each side of the decision boundaries. If a shape is within one of these zones, it means that the reliability of the decision is low so the shape must be rejected. These reject zones are defined by comparing the two better class scores. If they are too close, the shape is rejected. As it uses just the class scores, this confusion reject is possible for most of classifier kinds. The figure 1 shows the decision boundaries and the related confusion reject zones in an example of a RBFN used for the classification of three classes in two dimensions with one prototype per class.
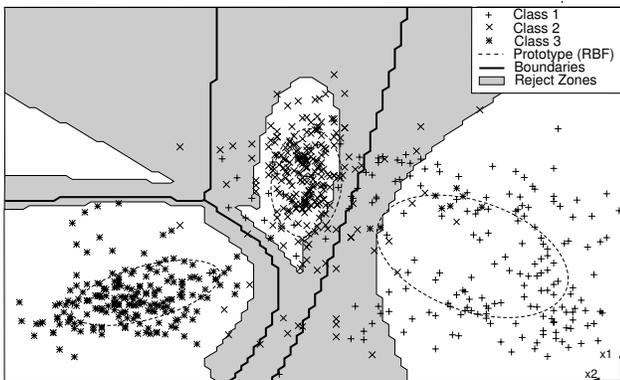

Figure 1. Confusion reject zones.

To evaluate this reject we use the compromise *Accuracy/Performance*. During the classification of a data set, the N_{Tot} examples are split in three cases: N_{Corr} correctly classified, N_{Err} misclassified and N_{Rej} rejected examples. Then we can define the performance and the accuracy as:

$$performance = N_{Corr}/N_{Tot} , \qquad (1)$$
$$accuracy = N_{Corr}/(N_{Corr} + N_{Err}) . \qquad (2)$$

The more the reject is hight, the more performance is low. The accuracy increases if rejected examples are errors. The both values must be maximized.

2.2. The distance reject

The distance reject allows to delimit the knowledge of the used classifier. In this way, it can reject patterns which do not belong to learned classes. Hence, if a shape is too far from the knowledge it must be rejected. Thus it can be used for outlier detection and rejection.

This reject is easily available for prototype based classifiers, *i.e.* activations of the prototypes allow to determine if a pattern is far from knowledge. The figure 2 shows the distance reject zones used to reject the class 4 which was not learned by the previous RBFN.
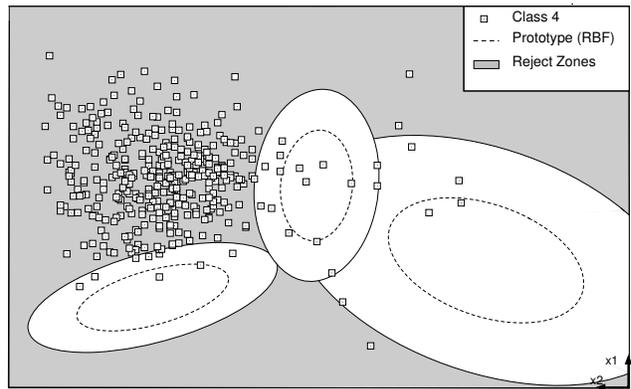

Figure 2. Distance reject zones.

To evaluate the distance reject, we use the compromise *False Reject/False Acceptance*. The False Reject Rate (FRR) is computed on a database of examples to accept and the False Acceptance Rate (FAR) on one of counterexamples to reject. During the classification with distance reject, some examples are rejected and some counterexamples are accepted. The more the reject increases, the more FAR decreases and the more FRR increases: the aim is to minimize both of them.

3. Formalization of the reject

To formalize the reject we use the notion of reliability functions [5]. A reliability function ψ in $\Re$ depends on the used classifier and of the nature of the wanted reject option. This function allows to determine the reliability you must have in the result of the classifier. The more a pattern must be rejected, the less is the reliability function. Contrary to [5] we permit to define a set of N reliability functions $\{\psi_i\}$ which allow more precision in the reject.

Thus the reject is defined with a set of N thresholds $\{\sigma_i\}$ each one associated to a reliability function ψ_i. Then to have a reject, all functions must be lower than their respective threshold:

$$\forall i = 1..N, \psi_i \leq \sigma_i. \qquad (3)$$

The use of several thresholds, is justified by [7] on class related thresholds. But we have generalized this notion to reliability functions. Furthermore, it allows to express different reject for each considering knowledge. Some of them need more reject than others.

To define different natures of reject we must firstly define the corresponding reliability functions. We present here some possible ones for the confusion and the distance reject used with a RBFN.

The distance reject delimits the knowledge of the classifier. In a RBFN, the main intrinsic knowledge is represented by the prototypes of the hidden layer. Thus we define as many functions as the number of prototypes using their activations μ_i:

$$\psi_i^{Dist} = \mu_i. \tag{4}$$

This function decreases when pattern is far away from the prototype. In [5] they use functions ψ_i^{DistO} for this reject considering the N_c output class scores s_i:

$$\psi_i^{DistO} = s_i. \tag{5}$$

When this function is used with RBFN, it keeps the notion of distance reject. Indeed, if a example is far away from all prototypes, all activations are low and all class scores are also low. But it is not the case with other classifiers, like MLP in [5].

The confusion reject determines if a shape is near the decision boundaries. So we use the difference between the better class C_1 and the second one C_2. Hence there are $N = N_c * (N_c - 1)$ functions:

$$\psi_{i,j}^{Conf} = \begin{cases} \frac{s_i - s_j}{s_i} & \text{if } s_i \neq 0, i = C_1, j = C_2, \\ 0 & \text{else.} \end{cases} \tag{6}$$

It must be noticed that the current example could only be accepted if ψ_{C_1,C_2}^{Conf} is greater than σ_{C_1,C_2}.

For these both reject natures, we can reduce to $N = 1$ the number of reliability functions like in [5], loosing precision:

$$\psi^{DistO1} = s_{C_1}, \tag{7}$$
$$\psi^{Conf1} = (s_{C_1} - s_{C_2})/s_{C_1}. \tag{8}$$

Now we have to learn automatically the thresholds σ_i without any dependency on the chosen reliability functions and on the kind of classifier.

4. Automatic thresholds learning

We present here a unified automatic algorithm which is a generic framework for thresholds learning for the different natures of reject. Different variants of this algorithm have been developed but in this paper we present just the most efficient one.

The threshold learning process uses an already learned classifier. It is based on the reliability functions using both an example database Be (to accept) and a counterexample database Bc (to reject). For the confusion reject the well recognized shapes are in Be and the misclassified ones in Bc. For the distance reject, Be contains the objects of classes learned by the classifier and Bc the classes not learned by it.

This algorithm has one parameter θ which is the maximum FRR allowed. It is compound of four steps:

1. At the initialization the reliability values ψ_i of examples and counterexamples are computed and the thresholds σ_i are set to reject all counterexamples (and so some examples);

2. Next steps are repeated while the FRR on Be is higher than the parameter θ;

3. The function `choose` decides the next threshold σ_r to be decreased;

4. The chosen threshold σ_r is decreased by the function `decrease` to accept more examples (and so some counterexamples).

In this study the function `choose` selects the threshold which minimizes the number of counterexamples accepted to accept one new example and the function `decrease` decreases the threshold to accept one more example.

5. Results

The aim is to show that the used reject nature must be chosen according to the application. Thus we study the different reliability functions for each compromise. To compare them the parameter θ varies from 0% FRR too high FRR. We work on the recognition of the 10 digits from the UNIPEN[8] database. With the distance reject, we reject the 26 letters from the Ironoff[11] databases. We use 21 features [1] to describe shapes and two RBF per class.

5.1. Distance reject

For the compromise *False reject / False acceptance* we use the curves of FRR versus FAR. So the aim is to be the nearest from the origin *i.e.* a classifier without any errors and rejects. Figure 3 compares the reliability functions associated to the distance reject *Dist* (eq. 4), *DistO* (eq. 5) and *DistO1* (eq. 7). We also compare them to the confusion reliability function *Conf* (eq. 6).

To compare these different approaches, we consider the FRR rate for a FAR of 15%. Firstly we can notice that the confusion reject is not an appropriate reject option for this compromise as *Conf* increases FRR of 28% w.r.t. *DistO*.
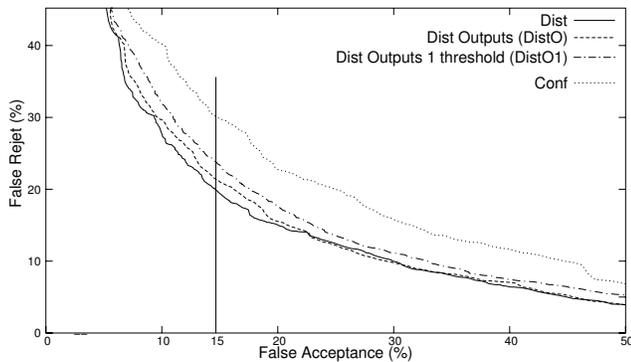
Figure 3. False reject / False acceptance.

Secondly the reliability functions based on intrinsic knowledge are a little more efficient than the ones based on discriminant information as *Dist* decreases FRR of 8% w.r.t. *DistO*. This low difference is due to the particularity of the RBFN as explain in section 3. Finally it must be noted that the used of only one threshold is less efficient than multi-threshold reject as *DistO1* make 10% more FRR w.r.t. *DistO*.

5.2. Confusion reject

For the compromise *Accuracy / Performance* we draw the curve giving *(100% - Performance)* versus *(100% - Fiability)*. The aim is to be the nearest from the origin. The figure 4 compares three reliability functions: *Conf* (eq. 6), *Conf1* (eq. 8) and also the distance reliability function *Dist* (eq. 4).
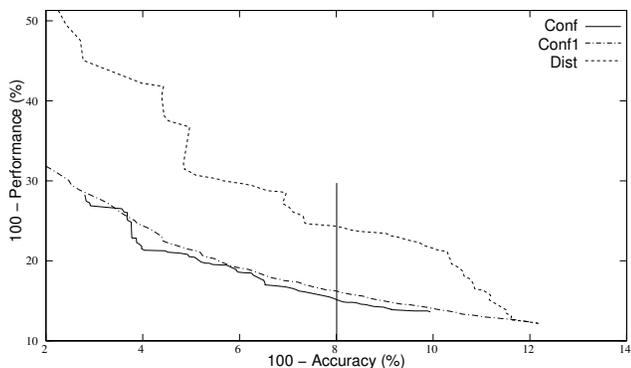

Figure 4. Accuracy / Performance.

To compare them, we consider an accuracy of 92%. We can see that the distance reject is not a good choice for this compromise as *Dist* makes 10% less performance than *Conf1*. Furthermore, the use of several thresholds is little better than using just one as *Conf* improves performance of 1% w.r.t. *Conf1*.

6. Conclusion

We show that nature of the reject option must correspond to the needs of the context. Thus we define two main reject natures. The confusion reject allows to increase the accuracy using discriminant information whereas the distance reject uses intrinsic information allowing to delimit knowledge of the recognizer to increase the robustness to false reject.

We define a formalism which allows to abstract the used nature of reject and the used classifier. With this formalism we define an algorithm to learn automatically all kind of reject.

The proposed formalism and algorithm are enough generic to allow new natures of reject and new approaches of learning. We will explore in future works these new ways, but also the combination of the confusion and distance reject.

References

[1] E. Anquetil and H. Bouchereau. Integration of an on-line handwriting recognition system in a smart phone device. In *Proc. of 16th ICPR*, volume 3, pages 192–195, 2002.

[2] E. Anquetil, B. Couasnon, and F. Dambreville. A symbol classifier able to reject wrong shapes for document recognition systems. In *GREC*, pages 195–202, 1999.

[3] C. Bishop. *Neural Network for Pattern Recognition*. Oxford University Press, 1995.

[4] C. Chow. On optimum recognition error and reject tradeoff. *IEEE Trans. on Inforamtion Theory*, 16:41–46, 1970.

[5] C. De Stefano, C. Sansone, and M. Vento. To reject or not to reject: That is the question - an answer in case of neural classifiers. *IEEE Trans. on Systems, Man and Cybernetics*, 30(1):84–94, 2000.

[6] C. Frélicot, M. Masson, and B. Dubuisson. Reject options in fuzzy pattern classification rules. In *Proc. of 3rd EUFIT*, 1995.

[7] G. Fumera, F. Roli, and G. Giacinto. Reject option with multiple thresholds. *Pattern Recognition*, 33(12):2099–2101, 2000.

[8] I. Guyon, L. Schomaker, R. Palmondon, M. Liberman, and S. Janet. Unipen project of on-line data exchange and recognizer benchmarks. In *Proc. of the 12th ICPR*, pages 29–33, 1994.

[9] R. Krishnapuram and J. Keller. A possibilistic approach to clustering. *IEEE Trans. on Fuzzy Systems*, 1(2):98–110, 1993.

[10] L. Oliveira, A. Britto Jr., and R. Sabourin. Improving cascading classifiers with particle swarm optimization. In *Proc. of 8th ICDAR*, pages 570–574, 2005.

[11] C. Viard-Gaudin, P. Lallican, S. Knerr, and P. Binter. The irest on/off dual handwritting database. In *Proc. of the 5th ICDAR*, pages 455–458, 1999.