# PATTERN RECOGNITION METHOD FOR OFF-BOARD AUTOMOTIVE VEHICLE FAILURE ISOLATION

**Nasser Charkaoui** * **Bernard Dubuisson** *
**Christophe Ambroise** * **Armand Boatas** **

* *Université de Technologie de Compiègne, Laboratoire HEUDIASYC UMR CNRS 6599, BP 20529, F-60205 Compiègne Cedex*
** *PSA Peugeot Citroën, 2 route de Gizy, F-78943 Vélizy-Villacoublay*

Abstract: This paper introduces a pattern recognition method specially dedicated to the detection and isolation of after-sale failures on an automotive vehicle. This method is well suited to deal with binary data. It is based on the similarity between an input pattern and some reference patterns characteristic of each failure class. The initial space representation is divided into several subspaces in order to allow classification of patterns which belong simultaneously to several classes. Experiments with real data demonstrate the good performance and the adapted structure of this classification system. 

Keywords: Diagnosis, pattern recognition, classification, binary elements, decision theory, automobile industry, failure detection and isolation.

## 1. INTRODUCTION

In many industrial applications, it has become more and more important to monitor the behaviour of complex systems, using multiple measurements. Diagnosis techniques are usually classified into two main categories. On the one hand, the model based approach uses sensor signals considered as the inputs and outputs of a dynamic system. Process monitoring and diagnosis are then conducted using a physical process or identified model. On the other hand, the feature based or pattern recognition approach needs no physical process model. System knowledge is assumed to be contained in a learning set composed of measurement vectors and associated operating conditions (Denoeux *et al.*, 1997). Pattern recognition can use several methods: statistical approaches, neural networks, fuzzy logic, possibility theory, belief functions etc.

In classification approach, a set of classes $\Omega = (\omega_1, \ldots, \omega_m)$ is defined by a corresponding learning set. The data set is composed of so-called feature vectors or instances (McLachlan, 1992). A feature vector is written:

$$\underline{x} = (x^1, \ldots, x^d). \tag{1}$$

The objective of classification is to build a decision function $d(\underline{x})$ which is a mapping from a multidimensional feature space $\mathbf{X}^d$ to a decision space $\mathbf{C}^m = [0,1]^m$:

$$\begin{cases} d : \mathbf{X}^d \rightarrow \mathbf{C}^m \\ \underline{x} \rightarrow d(\underline{x}) \end{cases}. \tag{2}$$

The statistical pattern recognition approach uses two different kinds of methods: the first ones are parametric and the other ones are non-parametric (Dubuisson, 1990; Freitas, 2002; Little and Rubin, 2002). Parametric methods are used when one

can make an assumption about the data probability density (gaussian distribution, Bernoulli distribution...). Non-parametric methods are used when no assumption can be made; so-called general approximators are then used. Examples of non parametric methods are: Parzen-Windows, K-Nearest Neighbours ....

In this paper, a non-parametric pattern recognition method for binary data classification is proposed. In this approach, the decision of classifying a pattern into a class is made by computing dissimilarities with a limited number of prototypes (one prototype for each class), resulting in fast classification and low storage requirements. A class membership function is defined based on a dissimilarity function in order to build a decision system. Generally speaking, pattern recognition methods deal with the case where each pattern $\underline{x}$ of the data set belongs exclusively to one class. Contrarily to this approach, in some applications, one or more classes could be present at the same time. Therefore, in the scope of this paper, a pattern can belong simultaneously to several classes. It is unclear in the latter setting precisely how to formalize the goal of a learning algorithm, and, in general, the right formalization may well depend on the problem at hand. Classical methods, such as naive Bayes classifier, mixture models, are neither simply nor directly applied in this case. In order to solve this problem, the initial representation space has been divided into class dedicated subspaces. The dissimilarity method is then applied in each specific subspace.

The remainder of the paper is organized as follows. In section 2, our industrial application is introduced and data modelling is discussed. Afterwards, in section 3, a new non-parametric classification method for binary data is presented. Furthermore, a new technique for classification of patterns which could belong to several classes, based on the previous introduced method, is described in section 4. Experiments and results obtained on real data application are presented in section 5. Finally, results are discussed in section 6.

## 2. AFTER-SALES OFF-BOARD DIAGNOSIS OF AUTOMOTIVE VEHICLE FAILURES

In today automotive vehicles, the introduction of on-board electronic systems, the growth of electronic based functions have made the reparability very difficult for after-sales technicians (Gissigner and Le Fort-Piat, 2002). In order to efficiently isolate a vehicle failure origin, they need external information in addition to their own expertise. Each electronic control unit features on-board diagnostic functions. When a fault is detected, a fault code is generated. It is recorded with additional contextual information on an on-board flash ROM memory. Moreover, fault codes are often insufficient to isolate a faulty component and the recorded data are corrupted with false alarms. Besides, the data are highly multidimensional: it is very difficult for a technician to analyse it in an exhaustive way. Therefore, it is essential to develop diagnosis methods which allow automatic interpretation of the recorded data available on a vehicle.

This paper introduces a classification method based on fault codes in order to isolate faulty components. A data read-out is represented by a pattern vector $\underline{x}$ defined as: $\underline{x} = (x^1, \ldots, x^d)$ where $x^j$ is an index indicating if the fault code $j$ occurs or not:

$$\begin{cases} x^j = 1 \text{ if fault code } j \text{ is present} \\ \qquad\quad \text{at least once in the recording}. \\ x^j = 0 \text{ otherwise} \end{cases} \quad (3)$$

## 3. BINARY DATA CLASSIFIER

In this section, the binary data classifier basic elements are described. Firstly, hypotheses and data are presented. Secondly, method basic idea is discussed. Finally, prototypes definition, dissimilarity indexes and membership function are given.

### *3.1 Hypotheses and data*

For a given kind of vehicle which has a finite number $d$ of fault codes and a finite number $m$ of possible failures, corresponding to $m$ faulty components to be replaced. Each failure is modelled by a pattern class.

Let us indicate by: $\mathbf{X}$ the learning set on all known classes, $\mathbf{X_k}(k = 1, \ldots, m)$ the learning set corresponding to the vectors of class $\omega_k$, $n$ the cardinality of $\mathbf{X}$, $n_k$ the cardinality of $\mathbf{X_k}(k = 1, \ldots, m)$. The following relations are resulting:

$$\mathbf{X} = (\mathbf{X_1}, \ldots, \mathbf{X_m}) = (\underline{x}_1, \ldots, \underline{x}_n) \quad (4)$$

and

$$n = \sum_{k=1}^{m} n_k. \quad (5)$$

Let us indicate by $\mathbf{Z} = (\underline{z}_1, \ldots, \underline{z}_n)$ the $n$ label vectors corresponding to $(\underline{x}_1, \ldots, \underline{x}_n)$. A label vector $\underline{z}_k$ is defined as:

$$\underline{z}_k = (z_k^1, \ldots, z_k^m), \quad (6)$$

where $z_k^j \in [0, 1]$ is the membership degree of vector $\underline{x}_k$ to class $\omega_j$. The closer $z_k^j \in [0, 1]$ is to 1, the more $\underline{x}_k$ is similar with class $\omega_j$ patterns and the larger the chance is to belong to $\omega_j$. On the contrary, the closer $z_k^j \in [0, 1]$ is to 0, the more $\underline{x}_k$ is dissimilar with class $\omega_j$ patterns and the smaller the chance is to belong to $\omega_j$.

### 3.2 From Neighbours to Prototypes

The k-nearest neighbours (k-NN) classifier is a well known non-parametric classifier, generally applied when data are of quantitative type. It provides good performance for optimal values of $k$ (Cho *et al.*, 1991). In the k-NN rule, a new input pattern is assigned to the most frequently represented class among its k-nearest training samples. The computational complexity is known to be an important drawback of k-NN techniques (Denoeux, 1995; Denoeux, 2000; Zouhal and Denoeux, 1998; Zouhal and Denoeux, 1995). This problem can be partially solved by synthesizing the learning set by a limited number of representative patterns called *prototypes*. In this approach, pattern classification is made by computing a dissimilarity index with $q$ prototypes: $\mathbf{P} = (\mathbf{P^1}, \ldots, \mathbf{P^q})$. Each prototype $k$ is assumed to get a degree of membership $u_l^k$ to each class $\omega_l$. Full membership of a prototype $\mathbf{P^k}$ to a class $\omega_j$ can be considered as a special case where $u_l^k = 1$ for $l = j$ and $u_l^k = 0$ for $l \neq j$.

### 3.3 Prototypes definition

In this paper, only one prototype is defined for each class but the method may be generalized to several prototypes per class. Class $\omega_k$ is composed of a binary kernel vector $\underline{a}_k$ and a deviation vector $\underline{\varepsilon}_k$. Therefore, the prototypes set is defined by:

$$\mathbf{P} = (\mathbf{P^1}, \ldots, \mathbf{P^m}) = (\{\underline{a}_1, \underline{\varepsilon}_1\}, \ldots, \{\underline{a}_m, \underline{\varepsilon}_m\}), \tag{7}$$

with $\underline{a}_j \in \{0,1\}^d$ and $\underline{\varepsilon}_j \in ]0, \frac{1}{2}[^d$.

Each prototype $\mathbf{P^k}$ is supposed to present a full membership to class $\omega_k$.

The binary kernel vector of class $\omega_k$, $\underline{a}_k = (a_k^1, \ldots, a_k^d)$ is defined as:

$$\begin{cases} a_k^j = 1 \text{ if } (\frac{\text{card}(x_i^j = 1, \underline{x}_i \in \mathbf{X_k})}{n_k}) \geq 0.5 \\ a_k^j = 0 \text{ if not} \end{cases}. \tag{8}$$

The deviation vector of class $\omega_k$, $\underline{\varepsilon}_k = (\varepsilon_k^1, ..., \varepsilon_k^d)$ is defined as follows:

$$\begin{cases} \varepsilon_k^j = \frac{\text{card}(x_i^j = 0, \underline{x}_i \in \mathbf{X_k})}{n_k} \text{ if } a_k^j = 1 \\ \varepsilon_k^j = \frac{\text{card}(x_i^j = 1, \underline{x}_i \in \mathbf{X_k})}{n_k} \text{ if } a_k^j = 0 \end{cases}, \tag{9}$$

where $\text{card}(x_i^j = 1, \underline{x}_i \in \mathbf{X_k})$ is the number of times the variable $x^j$ takes the value 1 within the $\mathbf{X_k}$ subset data.

$a_k^j$ is the majority element of variable $x^j$ within the data of class $\omega_k$. It is the median of $x^j$ in $\mathbf{X_k}$ subset data. $\varepsilon_k^j$ is the variation of $x^j$, within the $\mathbf{X_k}$ subset data, around $a_k^j$.

The closer $\varepsilon_k^j (j = 1, \ldots, m)$ is to zero, the better the class $\omega_k$ is represented by the prototype $\mathbf{P^k}$. The closer $\varepsilon_k^j (j = 1, \ldots, m)$ is to $\frac{1}{2}$, the more dispersed the data $\mathbf{X_k}$ is. This means that class $\omega_k$ is not well represented by prototype $\mathbf{P^k}$. In this case, definition of other classes $\omega_k$ related prototypes is necessary in order to improve the precision.

### 3.4 Dissimilarity indexes

The aim of a dissimilarity index is to measure the difference between two given vectors $\underline{y}_1$ and $\underline{y}_2$ (Diewert, 2002). A dissimilarity index on a set $\Omega$ is a function $ds$ from $\Omega \times \Omega$ to $\mathbb{R}^+$ which satisfies the following conditions:

$$\begin{array}{lll} \forall \underline{y}_1 & ds(\underline{y}_1, \underline{y}_1) = 0 & \text{(identity)} \\ \forall \underline{y}_1, \underline{y}_2 & ds(\underline{y}_1, \underline{y}_2) = ds(\underline{y}_2, \underline{y}_1) & \text{(symmetry)} \\ \forall \underline{y}_1 \neq \underline{y}_2 & ds(\underline{y}_1, \underline{y}_2) > 0 & \text{(positivity)} \end{array}. \tag{10}$$

Two indexes have been defined in this paper in order to measure the dissimilarity of an input pattern $\underline{x}$ with the prototypes $(\mathbf{P^1}, \ldots, \mathbf{P^m})$.

The first dissimilarity index $ds_1$ between $\underline{x}$ and the prototype $\mathbf{P^k}$ is defined as follows:

$$ds_1(\underline{x}, \mathbf{P^k}) = (R(\underline{x}, \mathbf{P^k}) - R(\mathbf{P^k}, \mathbf{P^k})), \tag{11}$$

where

$$R(\underline{x}, \mathbf{P^k}) = \sum_{j=1}^{d} [(|x^j - a_k^j|)\bar{\varepsilon}_k^j + (1 - |x^j - a_k^j|)\varepsilon_k^j], \tag{12}$$

$$R(\mathbf{P^k}, \mathbf{P^k}) = \sum_{j=1}^{d} \varepsilon_k^j \tag{13}$$

and

$$\bar{\varepsilon}_k^j = 1 - \varepsilon_k^j. \tag{14}$$

The second dissimilarity index $ds_2$ between $\underline{x}$ and the prototype $\mathbf{P^k}$ is defined as follows:

$$ds_2(\underline{x}, \mathbf{P^k}) = \sum_{j=1}^{d} [\ln(\frac{\bar{\varepsilon}_k^j}{\varepsilon_k^j})|x^j - a_k^j|]. \tag{15}$$

It can be easily checked that both indexes satisfy all the dissimilarity properties.

### 3.5 Membership function

A membership function is an estimation of the label vector $\underline{z}$ of a new pattern vector $\underline{x}$ . Using a dissimilarity index $ds$ ($ds = \{ds_1, ds_2\}$), a class $\omega_k$ membership function of pattern $\underline{x}$ with respect to pattern $\mathbf{P^k}$ is defined as follows:

$$\mu_k(\underline{x}) = \frac{1}{1 + \gamma_k ds(\underline{x}, \mathbf{P^k})}. \tag{16}$$

Figure 1 illustrates the variation of the membership function according to the dissimilarity values

around the prototype $\mathbf{P^k}$ for different values of $\gamma_k$. For a constant value of $\gamma_k$, the more an input pattern $\underline{x}$ is dissimilar with prototype $\mathbf{P^k}$, the smaller its membership value is (Bezdek, 1981). The optimal value of $\gamma_k$ must be determined according to the training data in order to minimize the classification error.

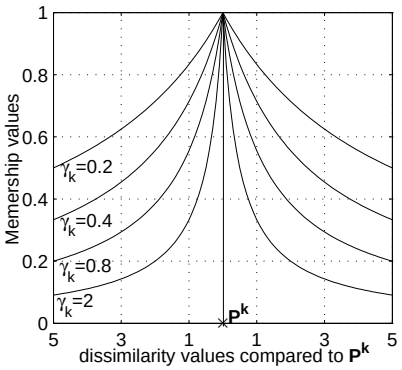


Fig. 1. Illustration of membership values according to dissimilarity values for several values of $\gamma_k$ ($\gamma_k = 0.2, 0.4, 0.8, 2$)

## 4. PATTERNS BELONGING TO MORE THAN ONE CLASS

Main (or most) pattern recognition methods deal with the case where data patterns belong to only one class. In the vehicle failure isolation application, it is possible to have several failures at the same time. Therefore, a pattern vector may belong simultaneously and with full certainty to several classes. This problem can be transformed into a classical one if the patterns which belong to several classes are considered as belonging to new classes. Let us assume that $h$ is the maximum number of failures which can occur simultaneously. Hence, the classes number of the system is of order $m^h$. The latter becomes too high and consequently not practical for large values of $m$.

In this paper, a new technique is introduced. It is based on the method presented in section 3. Only the singleton class patterns (i.e. patterns which only belong to one class) are presented to the classifier in the training phase. This method is based on 2 steps:

(1) Using the singleton class data, the whole representation space of dimension $d$ is divided into $m$ subspaces $\mathbf{EP_1}, \ldots, \mathbf{EP_m}$ (i.e. one subspace per class).
(2) The membership values of an input pattern $\underline{x}$ to classes $\omega_1, \ldots, \omega_m$ are calculated within their associated subspaces.

### *4.1 Subspaces definition*

The presence of a failure class $\omega_k$ is only related with the appearance of a list of fault codes. In no case, this failure would be influenced by the absence or the presence of other fault codes. Consequently, a subspace $\mathbf{EP_k}$ is associated with one class. The subspace $\mathbf{EP_k}$ is composed of a features list $\mathbf{CP_k}$. The latter are determined as:

$$\mathbf{CP_k} = \{x^j, (j = 1, \ldots, d)/a_k^j = 1, \overline{\epsilon}_k^j > E\}, \quad (17)$$

where $E \in [\frac{1}{2}, 1]$ is a threshold determined according to the pattern data. Equation 17 says that features of subspace $\mathbf{EP_k}$ are those which almost always occur when class $\omega_k$ is present.

### *4.2 Computation of membership degrees in the subspaces*

Usually, prototypes are defined using the initial representation space, whereas $\mathbf{P^k}(k = 1, \ldots, m)$ are here defined inside each corresponding subspace $\mathbf{EP_k}$. In order to compute the membership degrees of an input pattern $\underline{x}$ for classes $\omega_1, \ldots, \omega_m$, 3 steps are required:

(1) $m$ new patterns $\underline{x}^1, \ldots, \underline{x}^m$ are built by projecting $\underline{x}$ onto the subspaces $\mathbf{EP_k}(k = 1, \ldots, m)$:
$$\underline{x}^k = \underline{x}(\mathbf{CP_k}). \quad (18)$$
(2) The dissimilarity index $ds (ds = ds_1, ds_2)$ is computed: $ds_k = ds(\underline{x}^k, \mathbf{P^k})$.
(3) The membership value with respect to class $\omega_k$ is computed: $\mu_k(\underline{x}) = \frac{1}{1+\gamma_k ds_k}$.

### *4.3 Influence of subspace dimensions and parameter $\gamma$ determination*

The subspace $\mathbf{EP_1}, \ldots, \mathbf{EP_m}$ dimensions are not identical because classes are not defined by the same number of features. However, membership function depends on the subspace dimensions since the dissimilarity index increases as the feature number increases. To make ideas become clear, let us give an example using the $ds_2$ dissimilarity index. Consider two classes $\omega_l$ and $\omega_k$ where $\omega_l$ dimension is 2 and $\omega_k$ dimension is 10. In addition, each deviation vector element $\epsilon_i^j$ equals to $\alpha(\forall j$ and $i = \{l, k\})$. Consider a pattern $\underline{x}$ whose $\omega_l$ and $\omega_k$ feature values are all set to 0. In this case, $ds_2(\underline{x}, \mathbf{P^l})$ is lower than $ds_2(\underline{x}, \mathbf{P^k})$. If $\gamma_l, \gamma_k$ are the same, $\mu_l(\underline{x})$ is greater than $\mu_k(\underline{x})$. Actually, both $\omega_l$ and $\omega_k$ classes are equally unlikely, hence their membership values must be exactly the same. In order to solve this problem, the $\gamma_k(k = 1, \ldots, m)$ values are defined in such a way that they normalize the influence of subspace dimensions. Assume $d_k$ is the $\mathbf{EP_k}$

subspace dimension and the biggest dimension is $d_{max} = \max(d_k, k = 1, \ldots, m)$, then $\gamma_k$ is defined by:

$$\gamma_k = \frac{d_{max}}{d_k}. \quad (19)$$

### *4.4 Decision system*

The membership degrees of an input pattern $\underline{x}$ are combined in a decision system in order to classify it into one or several classes. The entire classification algorithm is illustrated in the decision procedure code.

**Decision procedure**

```
Begin
input pattern x
calculate μ(x) [μ(x) = (μ_1(x), ..., μ_m(x)]
A = (∃μ_k(x) > ρ)
if (A == true)
then [Rule 1]
    Classes = {ω_k/μ_k(x) > ρ}
    return Classes
else [Rule 2]
        x belongs to an unknown class
end if
End
```

Let us make some comments about the decision procedure rules. The membership values of an input $\underline{x}$ are calculated into subspaces $\mathbf{EP_1}, \ldots, \mathbf{EP_m}$. If some values are greater than a previously set threshold $\rho$ then $\underline{x}$ is affected to the corresponding classes (rule 1). If this condition is false, then $\underline{x}$ is rejected and considered as belonging to an unknown class (rule 2). One technique to determine $\rho$ threshold value consists in changing the value of $\rho$ between 0 and 1 with a fixed step (for example step equal to 0.05). For each threshold value the classifier performances are computed then the selected value is that corresponding to the highest performances. For $\rho = 0$, no instance will be rejected. For $\rho = 1$, all instances will be novelty rejected.

## 5. EXPERIMENTS

This section reports some experiments that demonstrate the efficiency of the classification method presented in this paper. Some tests were carried out on a vehicle. Eight data subsets have been obtained including 7 data subsets with only one failure per pattern $(\omega_1, \ldots, \omega_7)$ and one data subset with 2 failures per pattern: $\omega_1$ and $\omega_2$. Thus, there are 7 classes ($m = 7$). The vehicle can generate 1076 fault codes. Hence, the dimension of the initial representation space is $d = 1076$. Between 25 and 30 feature patterns are generated for each class. Some instances ($\cong 6\%$) which belong to no known class (outliers) are considered in order to evaluate if the classifier can detect instances from new classes. The data of each class are divided into 2 parts. One part (80 %) is used to learn and the other part (20 %) is used to test. It should be noted that in no case the subset with patterns which belong to two or more classes are used to determine the prototypes. Subspaces definition is done using the threshold value $E = 0.7$. Firstly, subset data with patterns which belong to 2 classes are not tested. In order to obtain a good estimation of the classification error rate and the novelty rejection rate, their values have been determined using 20 learnings. For every learning, a different training subset is used (consequently a different test subset). Figures 2 and 3 illustrate the obtained results while using the $ds_1$ dissimilarity function. Figures 4 and 5 illustrate the obtained results while using the $ds_2$ dissimilarity function. For both dissimilarity functions, the highest classifier performances are given for threshold decision value $\rho = 0.6$. The method gives good results with both $sd_1$ (mean classification error $\cong 7\%$) and $sd_2$ (mean classification error $\cong 5\%$) and results are very close. In addition, the outlier instances are correctly detected for both dissimilarity functions. Figures 2 to 5 show that classifier performances vary considerably according to the learning sample. This result is due to the fact that the data set is composed of few instances number. As a

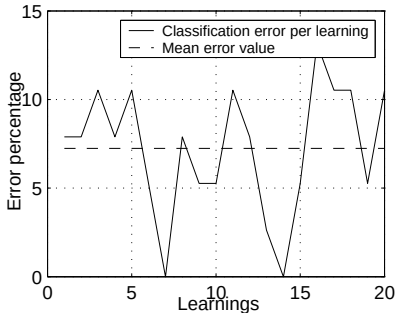


Fig. 2. Error classification estimation using the $ds_1$ dissimilarity function and the threshold values: $\rho = 0.6$.

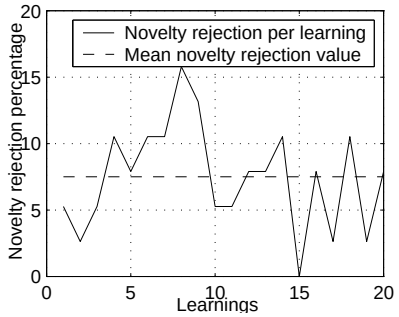


Fig. 3. Novelty rejection estimation using the $ds_1$ dissimilarity function and the threshold value: $\rho = 0.6$.

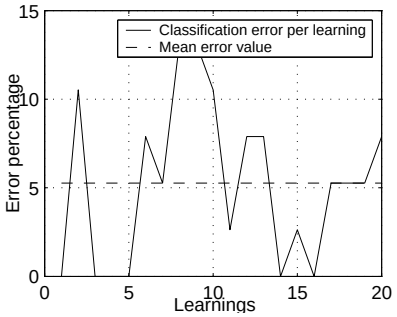


Fig. 4. Error classification estimation using the $ds_2$ dissimilarity function and the threshold values: $\rho = 0.6$.

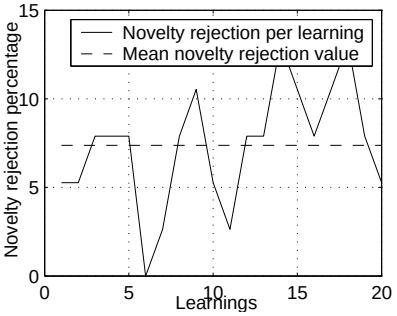


Fig. 5. Novelty rejection estimation using the $ds_2$ dissimilarity function and the threshold values: $\rho = 0.6$.

consequence, classifier performances depend a lot on the partition of the data set into learning and test sets. For every prototypes set obtained on the 20 learnings, 31 patterns belonging to classes $\omega_1$ and $\omega_2$ have been presented to the system. For the $sd_1$ and $sd_2$ dissimilarity functions, the classifier detects the correctly assigned classes with no error.

## 6. CONCLUSION

In this paper, a pattern classification method for binary data has been presented. This approach can be seen as a variant of a distance rule, well suited for the binary data classification problem. The classification method is based on the similarity of a new input pattern with a set of reference patterns. The main novelty introduced in this paper is that patterns can belong to several classes. In order to solve this problem, the initial feature space has been divided into several subspaces, one subspace per class. A subspace $k$ is composed of features that are always used when class $\omega_k$ is present. The membership values of an input pattern under consideration are combined in a decision procedure in order to assign the pattern to one or several classes or to consider the pattern as belonging to no known class (novelty rejection). The method has been applied to an automotive application. The results obtained on a real data set show that the method is efficient in both situations: when patterns belong to only one class and when patterns belong to several classes at the same time.

Surely, this method can be applied to different applications with simultaneously occurrence of several classes and using binary data.

## REFERENCES


Bezdek, J.C. (1981). *Pattern recognition with fuzzy objective function algorithms*. Plenum Press. New York, 233 Spring Street, N.Y. 10013.

Cho, T.H., R.W. Conners and P.A. Araman (1991). A comparison of rule-based, k-nearest neigbour and neural net classifiers for automated industrial inspection. *IEEE*.

Denoeux, T. (1995). A k-nearest neigbour classification rule based on Dempster-Shafer theory. *IEEE Transactions on Systems, Man and Cybernetics* **25**, 804–813.

Denoeux, T. (2000). A neural network classifier based on Dempster-Shafer theory. *IEEE Transactions on Systems, Man and Cybernetics-Part A: Systems and Humans* **30**(2), 131–150.

Denoeux, T., M. Masson and B. Dubuisson (1997). Advanced pattern recognition techniques for system monitoring and diagnosis: a survey. *Journal Européen des Systèmes Automatisés* **31**(9-10), 1509–1539.

Diewert, E. (2002). Similarity and dissimilarity indexes: an axiomatic approach. *Journal of Economic Literature Classification Numbers*.

Dubuisson, B. (1990). *Diagnostic et reconnaissance des formes*. Hermès. Paris, 34, rue Eugène Flachat, 75017.

Freitas, A.A. (2002). *Data mining and knowledge discovery with evolutionary algorithms*. Springer. Verlag Berlin Heidelberg, Germany.

Gissigner, G. and N. Le Fort-Piat (2002). *Contrôle-commande de la voiture*. Lavoisier. Paris, 11 rue Lavoisier, 75008.

Little, R.J.A. and D.B. Rubin (2002). *Statistical analysis with missing data, second edition*. Willey Series in Probability and Statistics.

McLachlan, G.J. (1992). *Discriminant analysis and statistical pattern recognition*. Wiley Series in Probability and Mathematical Statistics.

Zouhal, L.M. and T. Denoeux (1995). An adaptative k-nn rule based on Dempster-Shafer theory. *In Proceedings of the International Conference on Computer Analysis of Images and Patterns* pp. 310–317.

Zouhal, L.M. and T. Denoeux (1998). An evidence-theoretic k-nn rule with parameter optimisation. *IEEE Transactions on Systems, Man and Cybernetics C* **28**, 263–271.