Using set functions for multiple classifiers combination

Fabien Rico, Antoine Rolland
Laboratoire ERIC - Université Lumière Lyon 2
5 avenue Pierre Mendes-France
F-69676 BRON Cedex - FRANCE
antoine.rolland@univ-lyon2.fr

Abstract. In machine learning, the multiple classifiers aggregation problems consist in using multiple classifiers to enhance the quality of a single classifier. Simple classifiers as mean or majority rules are already used, but the aggregation methods used in voting theory or multi-criteria decision making should increase the quality of the obtained results. Meanwhile, these methods should lead to better interpretable results for a human decision-maker. We present here the results of a first experiment based on the use of Choquet integral, decisive sets and rough sets based methods on four different datasets.

1 Introduction

A classification problem consists in affecting an individual to a predefined category (or class) from its description via some variables. A classifier or model is a mapping function giving a unique class to each individual. In supervised classification, this function is built from a set of examples thanks to learning method. A large amount of different supervised learning algorithms are used, as indicated for example in [1]. It is then common to have, for a given situation, several results given by several classifiers. These classifiers can be based on the use of very different methods, or can use the same method with variations on leaning set. Using these information to enhance the quality of the classification is the purpose of the multiple classifier aggregation problem. Several methods already exist to solve the multiple classifier aggregation problem (see [11] and [12] for a survey). We propose here new aggregation procedures inspired by some aggregation methods developed in the framework of multi-criteria decision making and social choice theory. In section 2, we present the multiple classifier aggregation problem and stand the needed notations. In section 3 we briefly present some aggregation procedures based on the use of set functions and their applications in our framework; then in section 4 we present the implementations and tests of these methods and propose an analysis of the results.

2 Multiple classifier aggregation

2.1 Multiple classifier

It is well known that there exists no perfect classifier neither universal classifier : each classifier makes mistakes, and each classification algorithm is really performing only on specific situations. So in order to reduce the errors number, it should be interesting to mix the results of several classifiers. Given a specific classifier, we can increase its performances by adding one or several other classifiers. These new classifiers should be as independent as possible from the first one to be able to 'correct" its errors. It is the case for example in the boosting method where classifiers are built to obtain a maximum diversity [15]. But the new classifiers should also be intrinsicaly performant, although they will degrade the general performance. On the other hand, if the first classifier is still good, many other good classifiers will be strongly related to the first one : it should be difficult to find another good classifier independent from the first one. Therefore, adding a new classifier will not give much more information to the decision maker. So two mains properties have to be considered for selecting and aggregating classifiers: the quality of each classifier and the diversity into the set of classifiers. Mean rules and majority rules are very dependent of the quality of the added classifier for one hand, and of the independence between classifiers on the other hand (see [14] for a theoretical study). We investigate in this paper some other aggregation procedures which should manage less quality and/or dependent classifiers.

2.2 Aggregation procedures

There already exist several aggregation procedures for the multi-classifier problem [11]. Two of them are considered as reference procedures, due to their use facility, and the fact that they are easily understandable :

- the majority rule : the allocated class for an individual is the class chosen by a majority of classifiers.
- the mean rule : the allocated class for an individual is obtained by a cutting level applied on the mean of the different labels given by each classifiers.

In this paper, we present new aggregation procedures which aim at enhancing the quality of these two procedures. The general idea is that a multiple classifiers aggregation procedure can be seen as a particular case of either a voting procedure, or a multi-criteria aggregation rule, as seen below :

- suppose that each classifier is a voter, who can vote *for* or *against* allocating x in class a. Then the aggregation of classifiers problem can be seen as a voting procedure.
- suppose that each classifier is giving a score related to the strength of its conviction that individual x should be affected to class 1. This score can be seen as a value taken by a criterion related to the considered classifier. Then the multiple classifiers aggregation can be seen as a multi-criteria aggregation problem.

The field of multi-criteria aggregation procedure or voting procedure has been well studied in the past decades, and several approaches and methods are available solve these aggregation problems in social choice theory or multi-criteria decision aiding theory (see [2] for a review). We will focus here on a few a them, based on the same basic tool which is the use of set functions to represent the importance of each coalition of voters (resp. criteria).

2.3 Notations

We first establish the needed notations to have a formal representation of our framework. We define formally a classifier aggregation problem as a problem which consists in aggregating the information given by m classifiers on a individual ω in order to sort him into a pre-defined class. We note here Ω the set of n individuals $\{\omega_1, \ldots, \omega_n\}$ to be classified. Each individual ω_i is described by a set of q predictor variables $X^j \in \mathcal{R}$, $j = 1, \ldots, q$ and a class of membership $Y \in \{0,1\}$. By convention for the i^{th} individual ω_i we denote Y_i its class and $X_i = (X_i^1, \ldots, X_i^q) \in \mathcal{R}^q$ its representation.

A single classifier ϕ is a mapping :

$$\begin{array}{cccc} \phi : & \mathcal{R}^q & \rightarrow & [0,1] \\ & X & \mapsto & \phi(X) = p \end{array}$$

The result $\phi(X)$ is said to be a label according to X by the classifier ϕ. It can be seen as a score (probability, possibility...) for individual ω represented by X to belong to class 1.

According to this classifier, the chosen class should be obtained by a cutting level α :

$$c(X) = 1 \text{ if } \phi(X) > \alpha$$

Let $\mathcal{P} = \{\phi_1, \ldots, \phi_m\}$ be a set of m classifiers. An individual X can then be described by a vector of labels given by each classifiers $p_x = (p_1, \ldots, p_j, \ldots, p_m)$, or by a vector of chosen class affected by each classifier $c_x = (c_1, \ldots, c_j, \ldots, c_m)$.

The multiple classifier aggregation problem consists in aggregating the m outputs of the m classifiers to get a unique chosen class C.

An multi-classifier aggregation function Φ is a mapping :

$$\begin{array}{cccc} \Phi : & [0,1]^m & \rightarrow & \{0,1\} \\ & \{\phi_i(X)\} & \mapsto & \Phi(X) = C \end{array}$$

As $c_j(X) \in \{0,1\}$ $\forall \phi \in \mathcal{P}$ also, the multi-classifier aggregation function Φ can also takes a vector c_x as argument.

2.4 General framework

We focus here on aggregation problems with a few number of different classifiers (typically less than 10 classifiers). The input of the aggregation procedure is a vector $p_x = (p_1, \ldots, p_m) \in [0,1]^m$ of labels or a vector $c_x = (c_1, \ldots, c_m) \in \{0,1\}^m$ of classes. The result is a unique chosen class C_X.

A classifier gives for each individual a class which can be wrong or right, as soon as the real class Y of the individual is known. Let us recall that four situations can happen with a classifier. The following table stands the different sets cardinals for each possibility :

	real class	
obtained class	a	b
a	n_{aa}	n_{ab}
b	n_{ba}	n_{bb}

The quality of a classifier can be measured by several indicts.

- success ratio, denoted su.

$$su = \frac{n_{aa} + n_{bb}}{n}$$

The success ratio is the ratio of the number of well-affected individuals divided by the total number of individuals. It measures the ability of the classifier to well classify the individuals, whatever their class should be.

- precision ratio for the class a, denoted pr_a.

$$pr_a = \frac{n_{aa}}{n_{aa} + n_{ab}}$$

The precision ratio is the number of well-affected individuals of class a on the total number of individuals affected by the procedure to the class a. It measures the ability of the classifier to well reject the individuals which are not supposed to belong to the class a.

- callback ratio for the class a, denoted cr_a :

$$cr_a = \frac{n_{aa}}{(n_{aa} + n_{ba})}$$

The callback ratio is the ratio of the number of well-affected individuals of class a divided the total number of individuals of class a. It measures the ability of the classifier to well detect the individuals of class a : it is an asymmetric ratio, which is rather used in the field of statistic tests, or disease detection.

3 Set functions approaches

As mentioned in section 2.2, the multi-classifier aggregation problem has strong formal links with the preference aggregation problem in social choice theory or multi-criteria decision making. Considering each classifier as a voter, we wonder if there exist some coalitions (sets of classifiers) such that if all the classifiers of a coalition agree on class a for individual ω then the aggregation result of $\Phi(X)$ is class a. We would like to represent the existence of such coalitions through set functions, roughly giving to each subset of $\mathcal{P}$ a weight corresponding to its power as a coalition. We present in this paper three methods based on a decisive sets concept.

3.1 Capacity and Choquet Integral

3.1.1 Definition

One of the limits of the use of the weighted mean as an aggregation function is that it is unable to take into account synergy possibly happening between criteria to aggregate. A Choquet integral (see [5], [13] for a complete presentation) can then be seen as a non-additive generalization of the weighted mean. It is based on the use of a non-additive set function named capacity :

Definition 1. *Let N be a set of objects and $\mu = card(N)$. A capacity $v : 2^N \rightarrow \mathbb{R}^+$ is a set function such that $v(\emptyset) = 0$, and $A \subseteq B \subseteq N$ implies that $v(A) \leq v(B)$. A capacity is said to be normalized iff $v(N) = 1$.*

Formally, a Choquet integral is a function $\mathcal{C}$ from $[0,1]^\mu$ into $[0,1]$ such that, $\forall x = (x_1, \ldots, x_\mu) \in [0,1]^\mu$:

$$\mathcal{C}(x) = \sum_{i=1}^{\mu} x_{\sigma(i)}(v(A_{\sigma(i)}) - v(A_{\sigma(i-1)}))$$

where

- σ is a permutation on $\{1, \ldots, \mu\}$ such that $\sigma(1) \leq \sigma(2) \leq \ldots \leq \sigma(\mu)$
- v is a capacity on the set $\{1, \ldots, \mu\}$.
- $A_{\sigma(i)} = \{\sigma(i), \sigma(i+1), \ldots, \sigma(\mu)\}$

The Choquet integral has been very used in the fields of decision under uncertainty and multi-criteria decision aiding along the past decade, as mentioned in [7].

Choquet integral based aggregation rule	
Input	a set of individuals App
	$p_X = (\phi_1(X), \ldots, \phi_m(X) \; \forall \omega \in App$
	or $c_X = (c_1(X), \ldots, c_m(X)) \; \forall \omega \in App$
	$Y_\omega \; \forall \omega \in App$
Output	a capacity v on the set $\{1, \ldots, m\}$
Aggregation	$C(X) = 1 \iff \alpha < \Phi(X)$
	$\Phi(X) = \sum_{i=1}^{m} \phi_{\sigma(i)}(X)(v(A_{\sigma(i)}) - v(A_{\sigma(i-1)}))$

Table 1. Summary of Choquet integral model

3.1.2 *Analogy with the multi-classifier aggregation problem*

Each classification function ϕ_i is giving a label in $[0, 1]$ to the individual ω. Formally, each classifier can then be seen as a criterion and the function Φ as an aggregation function on these criteria. If a capacity function is defined on the set of classifiers, we can then use a Choquet integral as an aggregation function to obtain a global score for individual ω described by predictor variables X. We obtain, with the above notations,

$$\Phi'(X) = \sum_{i=1}^{m} \phi_{\sigma(i)}(X)(v(A_{\sigma(i)}) - v(A_{\sigma(i-1)}))$$

where

- σ is a permutation on $\{1, \ldots, m\}$ such that $\sigma(1) \leq \sigma(2) \leq \ldots \leq \sigma(m)$
- v is a capacity on the set $\{1, \ldots, m\}$.
- $A_{\sigma(i)} = \{\sigma(i), \sigma(i+1), \ldots, \sigma(m)\}$

The chosen class should then be obtained from Φ' by a cutting level α.

3.1.3 *Using Choquet integral in multi-classifier aggregation framework*

The aim of the use of a Choquet integral in a multi-classifier aggregation problem is to exhibit interactions which can appear between classifiers. In order to do so, we will use identification procedures based on a least square approach as proposed in [6]. These procedures use a learning set of individuals as input. The label vector p_X for each individual given by all the classifiers is known, such as the real class of each individual, and the identification procedure is an optimization program that compute the parameters of the Choquet integral that better fit the learning set. We then use the calculated parameters to infer the category of new individuals.

We implemented two procedures:

- *Choquet ls* uses least-square based approach to infer the parameters of the whole set of capacity values.
- *Choquet 3-add* uses least-square approach also but is limited to a 3-additive capacity, i.e. a capacity with no interactions between sets of more than 3 criteria (see [4] for details on *k*-additivity). This limit has been chosen as a compromise, in order to facilitate the computation as it divides by two the number of parameters, but keeping a relevant amount of interaction between criteria.

The first experiments show that between 50 and 85% of the Mbius coefficients are almost null. For example, we can have $v(\{1\}) = 0$, $v(\{2\}) = 0$ and $v(\{1, 2\}) = 1$. It means in that case that if a alternative is classified in class 1 for both classifiers 1 and 2, then it should be classified in class 1 by the Choquet Integral operator. Note that it is not always easy to obtain such a simple semantic interpretation of the capacity parameters.

It is not always easy to obtain such a simple semantic interpretation of the capacity parameters and we have not study thoroughly the results. However, the first experiments show that between 50 and 85% of the Mbius coefficients are almost null. For example, we can have as typical parameters $v(\{1\}) = 0$, $v(\{2\}) = 0$ and $v(\{1, 2\}) = 1$. It means in that case that if a alternative is classified in class 1 for both classifiers 1 and 2, then it is classified in class 1 by the Choquet Integral operator. This may be compared to the decisive set method described below, noting that the Choquet integral method can take into account both positive and negative examples in learning.

3.2 Decisive sets

3.2.1 *Definition*

In social choice theory, voters $v_1, \ldots, v_n$ are supposed to be able to give a preference relation between two candidates (or individuals) x and y. The fact that voter v_1 prefers candidate x to candidate y is denoted by $x \succ_{v_1} y$. Following Fishburn [3], a voter v_i is said to be *decisive for the pair* (x, y) if the fact that $x \succ_{v_i} y$ implies that x is preferred to y in the aggregated order, denoted $x \succ y$. A voter who is decisive for all pair x, y is said to be totally decisive, or just *decisive*. Inspired by Weymark [17], we can also define a decisive set of voters $V = \{v_i, \ldots, v_j\}$ for the pair (x, y) if the fact that $x \succ_{v_i} y$ $\forall v_i \in V$ implies that $x \succ y$.

Decisive sets based aggregation rule	
Input	a set of individuals App
	$c_X = (c_1(X), \ldots, c_m(X)) \; \forall \omega \in App$
	$Y_\omega \; \forall \omega \in App$
Output	$\mathcal{D}$, a set of K decisive subsets
	$D_k \subseteq \mathcal{P}, \; k = 1, \ldots, K$ for the class a
Aggregation	$C(X) = a \iff \exists D \in \mathcal{D}$
	such that $\{i \in 1, \ldots, M \mid C_i(X) = a\} \subseteq D_k$

Table 2. Summary of Decisive sets model

3.2.2 *Analogy with the multi-classifier aggregation problem*

Analogously, we can settle the following definitions in our framework:

Definition 2. *A classifier $\phi_i \in \mathcal{P}$ is said to be* decisive for *X for the class a if $c_i(X) = a \Rightarrow C(X) = a$. If ϕ_i is decisive for all X, ϕ_i is said to be totally decisive, or simply* decisive.

Definition 3. *a set of classifiers $P \subseteq \mathcal{P}$ is said to be* decisive for *X for the class a if $\forall \phi_i \in P$, $c_i(X) = a \Rightarrow C(X) = a$. If P is decisive for all X, P is said to be totally decisive, or simply* decisive.

3.2.3 *Using decisive sets in multi-classifier aggregation framework*

Practically, the aim of the identification process is to discover a set of decisive sets as small as possible for a given class a. In order to identify these decisive sets, we study a learning set of known individuals and we first catch all the existing decisive sets for each individual. Then we select the smallest (for the inclusion) decisive sets of classifiers that optimize the chosen ratio. We then use this set of decisive sets to infer the category of new individuals. The choice of $a = 0$ or $a = 1$ and the choice of the good ratio as an indicator of the fit quality have an importance on the detected decisive sets. We present below results obtained by considering successively $a = 0$ (method *Decisive sets 0*) or $a = 1$ (method *Decisive sets 1*) both focusing on the success ratio.

3.3 Rough sets dominance-based approximation

3.3.1 *Definition*

Another approach consists in using rough sets through the dominance-based rough set approach (see Greco, Matarazzo and Slowinski [9], [10]). In multi-criteria decision aiding, this approach uses decision rules to assign the alternatives to the different categories, with respect to some reference levels on each criterion. The axiomatic foundations of the rough set approach have been well studied by Greco, Mattarazo and Slowinski, including characterization of the sorting problem using a utility function or an outranking relation [8] or a Sugeno integral [16]. The dominance-based rough set approach for classification consists first in obtaining for each alternative the set of all the classes compatible with the dominance relation on the alternatives. It then produces a set of decision rules which characterize the allocation of each alternative to the possible classes. Decision rules present themselves as “if the value of the alternative on criteria i is at least ... and the value of the alternative on criteria j is at least ..., then the category of the alternative is at least ....”

dominance-based rough sets based aggregation rule	
Input	a set of individuals App
	$c_X = (c_1(X), \dots, c_m(X))\ \forall \omega \in App$
	$Y_\omega\ \forall \omega \in App$
Output	$\mathcal{D}$, a set of K decisive subsets
	$D_k \subseteq \mathcal{P},\ k = 1, \dots, K$ for the class a
Aggregation	$C(X) = a \iff \exists D \mathcal{D}$
	such that $\{i \in 1, \dots, M \mid C_i(X) = a\} \subseteq D_k$

Table 3. Summary of dominance-based rough sets model

3.3.2 *Analogy with the multi-classifier aggregation problem*

Each classification function ϕ_i is giving a score on $[0, 1]$ for the individual ω. Formally, each classifier can then be seen as a criterion and each individual as an alternative. Each alternative can then be classified only in one out of two classes. A dominance-based rough sets approach will then consist in sorting each individual into one out of three classes : individuals which are certainly in class a, individuals which are certainly not in class a, and ambiguous individuals, based on the dominance relation between individuals on values $\phi_i(X)$. We have then to produce a decision rules set to characterize the allocation of each individual to class 0 or 1. We can also directly use the classification vector c_X in the dominance-based rough sets approach. All the variables are then binary variables, and then decision rules can be interpreted as decisive sets of classifiers. We will then focus on this case.

3.3.3 *Using dominance-based rough set approach in multi-classifier aggregation framework*

Following the analogy developed in the decisive sets frameworks, we decide to aggregate the results c_X of the classifiers to obtain the final class for individual X. The inputs of the procedure are then only binary vectors $c_X = (c_1(X), \dots, c_m(X))$ with $c_i(X) \in \{0, 1\}$. The use of a dominance-based rough set approach in multi-classifier aggregation consists simply in finding a set of decision rules that better fits the learning set of individuals. Decision rules present themselves as “if $c_i(X) = a$ and ... and $c_j(X) = a$ then $c(X) = a$”. These rules can also be interpreted as decisive sets of classifiers : “if $c_i(X) = a$ and ... and $c_j(X) = a$ then $c(X) = a$” means that $\{\phi_i, \dots, \phi_j\}$ is a decisive set for class a. The used algorithm consists in building decisive sets from an empty set of classifiers, adding new classifiers in the set while the chosen ratio keeps on being optimized. The choice of $a = 0$ or $a = 1$ and the choice of the good ratio as an indicator of the fit quality have an importance on the detected decision rules. We present below results obtained by considering successively $a = 0$ (method *Rough sets 0*) or $a = 1$ (method *Rough sets 1*) both focusing on the success ratio.

4 Results

4.1 Data sets

We have compared those aggregation methods versus majority and mean rules for the following four datasets:

- UCI’s dataset `Letter`: recognition of letter “R” versus “B”.
- UCI’s dataset `Musk (v2)`: prediction if a molecule is (or not) a musk.
- Leo Breiman’s `Ringnorm` and `Threenorm`: recognition of two normal distribution with different mean and covariance.

Those datasets have medium size (detailed in table Tab:datasets) from 1500 to 6600 individuals), which gives sufficient individuals for the two learning steps (training simple classifiers and training aggregating methods). They have 2 classes and two of them are real examples (Letter and Musk) while the others (Threenorm and Ringnorm) are constructed data.

	nb indiv.	nb var.	prop of 1
Letter	1524	16	49.7%
Musk	6599	166	84%
Ringnorm	2128	20	50%
Threenorm	2128	20	50%

Table 4. List of the considered datasets.

4.2 Compared methods

We have compared the error, precision and call-back ratios through the three different methods for the four datasets. In order to do so, we split each dataset into a learning set L and a test set T. The learning set has been used to train the classifier and the test one to compare the computed class with the true one. More, our method used two levels of training, one for the simple classifiers to build m models and one for the aggregation model. So for our algorithm, the learning set L is itself split in two equal parts L_{train} and L_{agg}.

- L_{train} is used for training 7 simple well-known classifiers:
  - Breiman's random forest from the `randomForest` R library;
  - ada boost from the `ada` R library;
  - support vector machine using C classification and Gaussian kernel (`ksvm` function from `kernlab` R package);
  - linear Discriminant Analysis from `MASS` R package;
  - logistic regression using `glm` from `stats` R package;
  - single decision tree C4.5 using `J48` function provided by `RWeka` R package;
  - k nearest neighbours using `IBk` function from `RWeka` R package by default (k=1).

  Then we obtain $\begin{array}{rccl} p: & \Omega & \rightarrow & [0,1]^m \\ & x & \mapsto & p_x = (p_1, \ldots, p_m) \end{array}$
- The responses of the obtained classifiers are computed on L_{agg} and T, to obtain respectively the $p(L_{agg})$ and $p(T)$ results.
- The aggregation operator is trained using classifiers responses $p(L_{agg})$ and true classes $Y(L_{agg})$ in order to obtain the multi-classifier aggregation function Φ.
- The aggregated response for test set $\Phi(p(T))$ is computed and compared to the true class $Y(T)$ to compute the different ratios.

For mean and majority aggregation, the two levels learning is not necessary, so the classifier's training process is done one more time using the entire learning set L.

We also use Wilcoxon signed rank test to detect if the differences are significant or not. Our several learning sets and test sets are computed using 10 cross-validations. This means that the dataset is divided into 10 disjoint parts. We repeat the same test 10 times, each time, one part is used as test set and one is used for learning algorithm. The presented ratios are the means of the 10 corresponding results and the significance of the differences is computed thanks to Wilcoxon test.

4.3 Results

We present in tables 5 to 8 the results of our experiment on the different datasets. For each dataset, we present success ratio for each aggregation method, precision and callback ratio for class 1. For each aggregation method we indicate the significance degree (with $\alpha = 5\%$) compared first to the mean rule and second to the majority one.

- "+" denotes that the proposed aggregation method is significantly better than mean (respect. majority) rule,
- "-" denotes that it is significantly worse than mean (respect. majority) rule,
- "=" denotes that the difference is not significant.

For example in table 8, the success ratio of rough set oriented for 0 class is 87.1%, which is significantly better than majority rule but not than mean rule.

We can see that aggregation methods are often better than majority or mean rule, rarely worst (and never for success ratio). These results are promising as they are obtained with non optimized algorithm. For example we haven't study the effect of the size of L_{train} and L_{agg}, choosing same size for the both. This means that simpler classifiers (majority and mean rule) are trained on 2 times bigger sets. Our first intuition was that the orientation of decision or rough sets research should have an effect on precision and callback ration, but this is not obvious in our experiments. However, further studies in this direction certainly need to be lead.

Agg. method	Success ratio	Precision ratio	Call-back ratio
Mean	98.7	98.8	98.5
Majority	98.8	99.1	98.5
Decisive sets 1	98.6 =/=	99.0 =/=	98.2 =/=
Decisive sets 0	98.7 =/=	98.8 =/=	98.5 =/=
Rough sets 1	98.6 =/=	99.5 =/=	97.7 =/=
Rough sets 0	98.5 =/=	98.4 =/=	98.5 =/=
Choquet ls	98.8 =/=	99.2 =/=	98.4 =/=
Choquet 3-add	98.8 =/=	99.1 =/=	98.5 =/=

Table 5. Comparison of several methods for the Letter R/B data set

Agg. method	Success ratio	Precision ratio	Call-back ratio
Mean	97.6	97.7	99.5
Majority	97.7	97.8	99.6
Decisive sets 1	97.9 =/=	98.0 =/=	99.6 =/=
Decisive sets 0	98.1 +/+	98.6 +/+	99.2 -/-
Rough sets 1	98.2 +/+	98.5 +/+	99.3 -/=
Rough sets 0	98.2 +/+	98.4 +/+	99.5 =/=
Choquet ls	98.2 +/+	98.7 +/+	99.2 -/-
Choquet 3-add	98.2 +/+	98.7 +/+	99.1 +/+

Table 6. Comparison of several methods for the Musk data set

Agg. method	Success ratio	Precision ratio	Call-back ratio
Mean	95.9	94	98.3
Majority	94.2	92.5	96.2
Decisive sets 1	98.4 +/+	98.1 +/+	98.7 =/+
Decisive sets 0	97.2 =/=	98.3 +/+	96.0 =/=
Rough sets 1	98.5 +/+	98.2 +/+	99.0 =/+
Rough sets 0	92.8 =/=	90.1 =/=	99.4 +/+
Choquet ls	98.4 +/+	98.2 +/+	98.7 =/+
Choquet 3-add	98.4 +/+	98.2 +/+	98.7 =/+

Table 7. Comparison of several methods for the Ringnorm data set

5 Conclusion

In this paper, we obtained promising results which need further investigations. Among others, we propose two issues which are in our opinion relevant to be study:

- Does this approach can be applied to a larger number of classifiers ? This will be interesting to use it in ensemble methods framework, where several tens (or hundreds) of classifiers are aggregated. This leads to computation problems, because the complexity of some methods grows exponentially with the number of simple classifiers.

Agg. method	Success ratio	Precision ratio	Call-back ratio
Mean	85.8	86.1	85.4
Majority	86.	85.9	86.3
Decisive sets 1	86. =/=	83.7 =/-	89.9 +/+
Decisive sets 0	86.5 =/=	87.1 =/+	85.7 =/=
Rough sets 1	86.6 =/=	88.1 +/+	84.7 =/-
Rough sets 0	87.1 =/+	85.6 =/=	89.3 +/+
Choquet ls	87.5 +/+	87.4 =/+	87.7 =/=
Choquet 3-add	87.6 +/+	87.5 +/=	87.8 =/=

Table 8. Comparison of several methods for the Threenorm data set

- May these methods be used for selecting classifiers ? Indeed, rough set methods give generally a small number of rules. This may be seen as a simplification of the original set of classifiers. One drawback of aggregating different classifiers is that the process disintegrate the decision in multiple classifier, making it impossible to understand. So a human decision maker may need such a simplification.

REFERENCES

[1] Christopher M. Bishop, *Pattern Recognition and Machine Learning*, Information Science and Statistics, Springer, 2006.

[2] *Concepts and Methods of Decision-Making*, eds., Denis Bouyssou, Didier Dubois, Marc Pirlot, and Henri Prade, Wiley-ISTE, 2009.

[3] P. C. Fishburn, *The Theory of Social Choice*, Princeton University Press, 1973.

[4] M. Grabisch, 'k-order additive discrete fuzzy measures and their representation', *Fuzzy Sets and Systems*, **92**, 167?189, (1997).

[5] M. Grabisch and M. Roubens, 'Application of the Choquet integral in multicriteria decision making', in *Fuzzy Measures and Integrals-Theory and Applications*, eds., M. Grabisch, T. Murofushi, and M. Sugeno, 348–374, Physica Verlag, (2000).

[6] Michel Grabisch, Ivan Kojadinovic, and Patrick Meyer, 'A review of methods for capacity identification in choquet integral based multi-attribute utility theory: Applications of the kappalab r package', *European Journal of Operational Research*, **186**(2), 766–785, (2008).

[7] Michel Grabisch and Christophe Labreuche, 'A decade of application of the choquet and sugeno integrals in multi-criteria decision aid', *4OR: A Quarterly Journal of Operations Research*, **6**, 1–44, (2008).

[8] S. Greco, B. Matarazzo, and R. Slowinski, 'Conjoint measurement and rough set approach for multicriteria sorting problems in presence of ordinal criteria', in *A-MCD-A, Aide Multicritère à la Décision/Multiple Criteria Decision Aid*, eds., A. Colorni, M. Paruccini, and B. Roy, 117–144, European Commission, Joint Research Centre, EUR 19808 EN, Ispra, (2001).

[9] S. Greco, B. Matarazzo, and R. Slowinski, 'Rough sets theory for multicriteria decision analysis', *European Journal of Operational Research*, **129**, 1–47, (2001).

[10] S. Greco, B. Matarazzo, and R. Slowinski, 'Rough sets methodology for sorting problems in presence of multiple attributes and criteria', *European Journal of Operational Research*, **138**, 247–259, (2002).

[11] L. I Kuncheva, *Combining Pattern Classifiers. Methods and Algorithms*, Wiley, 2004.

[12] Ludmila I. Kuncheva. Classifier ensembles: Facts, fiction, faults and future, 2008. (slides, plenary talk).

[13] J.-L. Marichal, 'An axiomatic approach of the discrete Choquet integral as a tool to aggregate interacting criteria', *IEEE Transactions on Fuzzy Systems*, **8**(6), 800–807, (December 2000).

[14] Dymitr Ruta and Bogdan Gabrys. A theoretical analysis of the limits of majority voting errors for multiple classifier systems, 2000.

[15] Robert E. Schapire, Yoav Freund, Peter Bartlett, and Wee Sun Lee. Boosting the margin: A new explanation for the effectiveness of voting methods, 1997.

[16] R. Slowinski, S. Greco, and B Matarazzo, 'Axiomatization of utility, outranking and decision-rule preference models for multiple-criteria classification problems under partial inconsistency with the dominance principle', *Control and Cybernetics*, **4**(31), 1005–1035, (2002).

[17] J. A. Weymark, 'Arrow's theorem with social quasi-orderings', *Public Choice*, (42), 235–246, (1984).