Unsupervised one class identification by selecting and combining ranking functions

Antoine Cornuéjols[1] et Christine Martin[1]

[1]AgroParisTech, département MMIP et INRA UMR-518, 16, rue Claude Bernard , F-75231 Paris Cedex 5 (France)

20 avril 2014

Résumé

We study the problem of identifying a class of interest in an unsupervised data set. Assuming that a set $\mathcal{F}$ of score functions is available, of unknown performance for the task at hand, we propose a method in order to *select useful functions* from the set. Each of these functions induces a ranking over the data set.

We then show how to *combine the base rankings* thus obtained. Experimental results demonstrate that the combined performance is almost as good, or better, than the performance of the best, but unknown, score function in $\mathcal{F}$. In addition, we show, under some simplifying assumptions, how a proper combination of the base rankings allows one to end up with DNF formulas involving the selected score functions that converge to optimal precision and recall with respect to the target concept, if the capacity of $\mathcal{F}$ permits it. Such formulas, easily interpretable, are very desirable in the exploratory context of data mining.

Mots-clef : Unsupervised learning, Ensemble methods.

1 Introduction

Data exploration aimed at discovering interesting classes of patterns is an essential part of scientific discovery or, more mundanely, of data mining. For instance, in bioinformatics, many research works look for the identification of genes that respond to some conditions in the environment, or for finding proteins that could potentially interact with some given target drugs. In a different context, the IRS (Internal Revenue Service) would like to identify the most likely tax evaders. More generally, fraud detection is a growing application area. In each case, there is one class of interest that gathers objects the expert is looking for against the other data points.

In this exploratory setting, it is difficult to come up with informative functions good at distinguishing between the interesting data points versus the non interesting ones. While it might be easy to get candidate evaluation functions from experts or from libraries of functions commonly used in statistics or in Machine Learning, or even to generate such functions automatically, it is difficult in an unsupervised context to assess their merit. Therefore one is left guessing which one(s) of these functions to rely on. Additionally, for many application domains, and especially those where data is described by a large number of features, it is highly desirable that the class of interest be described in an interpretable way. This means that the class of interest should be expressed as much as possible using understandable features. For most experts, understanding and the capacity for reasoning imply descriptions that use combinations of predicates like disjunctive normal forms (DNF). This allows him/her to gain insight in what makes the class of interest apart and how this can be related to the current domain theory, possibly stimulating some revision of the theory.

In this work, we study the following problem. We suppose that there exists a set $\mathcal{S}$ of m data points from the input space $\mathcal{X}$ with no labels : $\mathcal{S} = \{\mathbf{x}_1, \ldots, \mathbf{x}_m\}$ that has been generated by an unknown mixture of distributions of which some components, belonging to $\mathbf{P}^+_{\mathcal{X}}$, correspond to the class of interest that we call $\mathcal{S}^+$, and the other components, $\mathbf{P}^-_{\mathcal{X}}$, correspond to the set of the remaining data points $\mathcal{S}^-$. The sets $\mathcal{S}^+$ and $\mathcal{S}^-$, such that $\mathcal{S}^+ \cup \mathcal{S}^- = \mathcal{S}$, are unknown and must be identified as well as possible.

In addition, we suppose that a set $\mathcal{F}$ of evaluation

functions (or score functions) is available, each function associating a score to a data point : $f_i : \mathcal{X} \to \mathbb{R}$. Nothing is assumed *a priori* about the usefulness of each function $f_i \in \mathcal{F}$, and in particular, one does not know if any given function is "aligned" with the target concept, that is if it tends to put the data points of the class of interest toward the top of the induced ranking over the data set $\mathcal{S}$.

We propose a method for identifying useful score functions in $\mathcal{F}$, if some exist, in this completely unsupervised setting. The basic idea is to look at the correlation between the rankings induced by the score functions over $\mathcal{S}$ and to select functions with a particular property. We explain how one can use the base rankings in order to get a combined ranking of the data points in $\mathcal{S}$ with good performances.

We end up by demonstrating, under some simplifying assumptions, how a proper combination of the base rankings allows one to end up with DNF formulas that converges to optimal precision and recall with respect to the target concept, if the capacity of $\mathcal{F}$ permits it.

2 The selection of useful evaluation functions

2.1 Principle of the method

In supervised learning with two classes ('+' and '−'), one looks for a decision function that provide a good separation between the training points of the two classes. It is usually possible to vary this decision function in the input space $\mathcal{X}$ by adjusting the value of some parameter(s). That way, the function may induce a ranking over the training set $\mathcal{S}$. If a null empirical risk is accessible, the ranking put all the positive data points before (or after) all the negative ones, and the ROC curve that can thus be computed has an AUC (Array Under the Curve) of 1.

If a combination of functions is employed, as in boosting, then one tries to use functions that, individually, induce good rankings of the training set (subject to the bias of the space of hypothesis functions $\mathcal{F}$ and to possible additional regularization constraints). The regions in the input space $\mathcal{X}$ where the positive examples lie therefore correspond to regions where several base functions agree on their ranking of the examples. I.e. in these regions, the selected score functions have put the examples towards the top of their ranking of the elements of $\mathcal{S}$.

We draw inspiration from that same idea in the context of unsupervised learning. As described in the previous section, we assume that a set $\mathcal{F}$ of evaluation functions exists, and we want to select the ones that are such that they place the positive data points at the top of their ranking. The only thing is that now the data points come without labels. We therefore have to find another lever.

The key assumption is that the data set at hand exhibits special regularities, otherwise it would not be of interest to any expert. Therefore, if one finds that there exists some match between the rankings of two evaluation functions over the data set $\mathcal{S}$ that usually do not exist over random data sets (of the same number of elements), one can suspect that the match is due to some specific regularity in the data set. This is the basis of the proposed approach.

We now have to settle on a correlation measure between evaluation functions, or rather between rankings.

2.2 Correlation measures

A measure of correlation between rankings estimates how much information about the rank in $\mathcal{S}$ of an example $\mathbf{x}$ by a given evaluation function f_i provides about the rank of the same example by another function f_j. Two measures are specially used : the *Spearman Rank-Order Correlation* and the *Kendall rank correlation coefficient*. In the context of Information Retrieval, the *Discounted cumulative gain* (DCG) and its normalized version (NDCG) are equally very much employed [**?**] (see [**?**] for a theoretical study). On advantage of the NDCG is to weight the correlation measure in function of the rank, that is to favor the objects that are ranked at the top of the rankings.

However, when one is considering only two classes of objects, with no hierarchy within each class, taking the rank into account is useless, and can even be misleading. This why we introduce another correlation function which is close the *Jaccard index*.

In the following, top_n^i will be used to denote the n examples of $\mathcal{S}$ that are top ranked by the evaluation function f_i. Similarly, $\cap_n^{i,j}$ will denote the intersection of the top_n elements by two evaluation functions f_i and $f_j : \cap_n^{i,j} = top_n^i \cap top_n^j$. Thus, if $top_5^i = \{a, b, c, d, e\}$ and $top_5^j = \{g, a, f, e, d\}$, then $\cap_5^{i,j} = \{a, d, e\}$.

We propose to measure the correlation between two evaluation functions on a set $\mathcal{S}$ by considering the values of $\cap_n^{i,j}$ when $1 \leq n \leq m$ if $\text{Card}(\mathcal{S}) = m$.

This measure is inspired by the hypergeometric law which gives the probabilistic law obeyed by the size of the intersection of two independent draws without replacement of n elements in a set of size m. The hy-

pergeometric law gives :

$$\mathbf{p}(|\cap_n^{i,j}| = k) = \frac{\binom{n}{k} \cdot \binom{m-n}{n-k}}{\binom{m}{n}}$$

For example, two independent draws of 500 elements among 6,000 have a maximal probability of sharing 42 elements. On can notice that $k/n = n/m$ (e.g. $42/500 \approx 500/6000$).

If the size of the intersection of two draws differs significantly from the most probable value given by the geometrical law, then it is unlikely that the draws are independent. In on extreme case, one draw is a copy of the other one, and then : $|\cap_n^{i,j}| = n, (\forall n \leq m)$. At the other end of the spectrum of possibilities, one draw avoids as much as possible to draw the same elements as the other one. For instance one ranking is the inverse of the other one. Then the size of the intersection is 0 up to $n = \lfloor m/2 \rfloor$, before rising as $\frac{2(n-(m/2))}{n}$ (see Figure 1). There exists therefore a whole spectrum of intersection laws between these two extreme cases.

It is essential to notice the special curve that one would obtain if two perfect base evaluation functions were selected, that is if they sorted the '+' elements of $\mathcal{S}$ at the top of their ranking, but were otherwise uncorrelated. This is depicted in Figure 1 on the right. The curve $|\cap_n^{i,j}|/n$ is thus intrinsically related to the detection of an AND function that could describes the '+' class.

In the proposed method, one tries to find out the interesting evaluation functions by measuring the difference in the correlation of their rankings of the elements in $\mathcal{S}$ as compared to the mean value of the correlations measured on random samples $\mathcal{S}_0$ of same size.

Figure 2 depicts a typical difference. Here the evaluation functions are ANOVA and RELIEF [**?**] and the data corresponds to 6,400 genes. The task was to find out if some genes were sensitive to low radioactivity levels. The upper curve $|\cap_n^{i,j}|$ shows the correlation over the data, while the lower curve with confidence intervals is obtained by computing the intersections $|\cap_n^{i,j}|$ over random samples $\mathcal{S}_0$ (here 100).

The difference in the measured correlations can be more or less accentuated depending on the difference in the classes '+' and '−'. There can be an overcorrelation peak that can be indicative of the number of positive objects in $\mathcal{S}$ (see Figure 3).

2.3 A theoretical analysis

In this section, we develop a simple model in order to allow us (in Section 5) to devise a strategy for discovering interpretable expressions of the hidden regularities in the data.

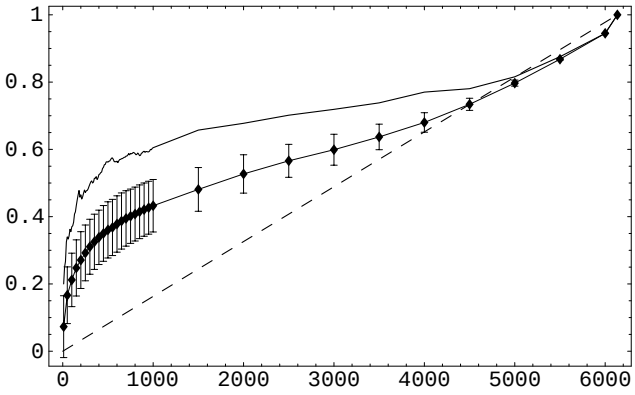

FIGURE 2 – Correlation curves measured on the data set (upper curve) and on random samples (lower curve with confidence intervals).

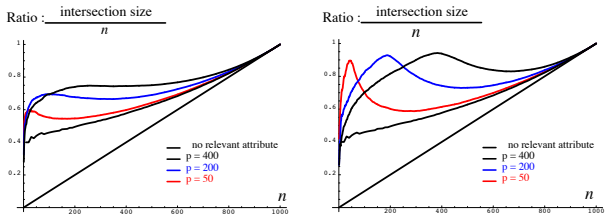

FIGURE 3 – Correlation curves between rankings of an artificial data set of 1,000 elements for various numbers of elements of class '+', here 50, 200 and 400. The peaks are accentuated on the right graph which corresponds to an easier problem.

We start by assuming that the evaluation functions are characterized by a positive (or negative) propensity to put the elements of class '+' at the top of their ranking. This propensity can be modeled by a ROC curve, of which one of the simplest form is given in Figure 4 [**?**]. When $1-\varepsilon_y > \varepsilon_x$ the function is positively aligned with an ideal function that would sort the '+' elements before the '−' ones, and the AUC is > 0.5.

In the simple analysis reported here, we suppose that we consider two evaluation functions f_i and f_j of the same strength (defined by ε_x and ε_y), that is they share a common ROC curve. The theoretical study with functions exhibiting different ROC curves does not change qualitatively the results.

Let us compute the size of the intersection of the top$_n$ elements : $|\cap_n^{i,j}|$. Let x be the number of false positive elements. Therefore, x varies on the FP axis. Let m^+ be the number of positive elements in $\mathcal{S}$ and

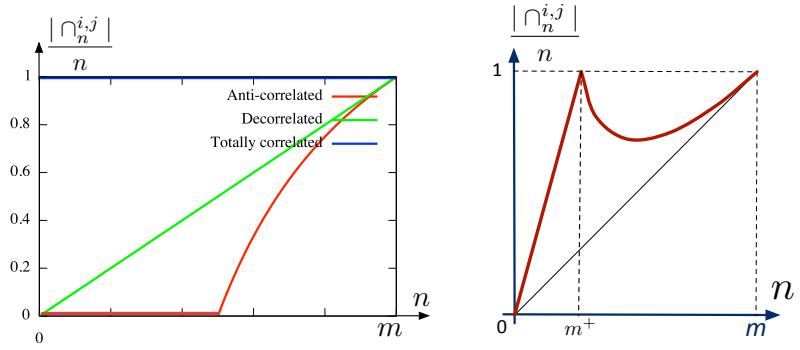

FIGURE 1 – The curve $|\cap_n^{i,j}|/n$ function of n. Two independent draws should approximately result in the diagonal law. (Left) Two maximally correlated draws give $|\cap_n^{i,j}|/n = 1 \ (\forall n)$. Two draws maximally inversely correlated give the red curve at the bottom. All possible behaviors fall between these two extreme curves. (Right) The characteristic curve for two rankings from uncorrelated but perfectly informed evaluation functions.

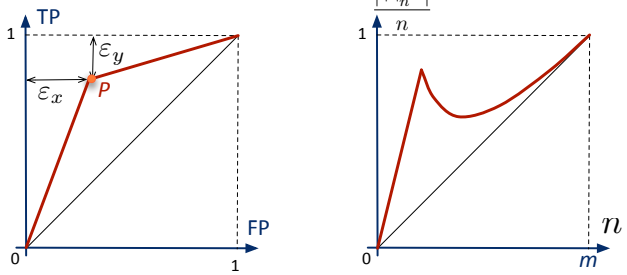

FIGURE 4 – (Left) The simple model of the ROC curve used in the theoretical analyses. (Right) The resulting curve of the most probable correlation size $\frac{|\cap_n^{i,j}|}{n}$.

m^- be the number of negative elements. Then, we have two phases to consider.

1. 1st phase : $x \leq \varepsilon_x$. One finds :

$$\begin{cases} n & = x\,m^- + \frac{1-\varepsilon_y}{\varepsilon_x}\,x\,m^+ \\ |\cap_n^{i,j}| & = x^2\,m^- + \left(\frac{1-\varepsilon_y}{\varepsilon_x}\right)^2 x^2\,m^+ \end{cases} \quad (1)$$

giving, for the first part of the curve, the equation :

$$\begin{aligned} \frac{|\cap_n^{i,j}|}{n} &= \frac{x^2\,m^- + \left(\frac{1-\varepsilon_y}{\varepsilon_x}\right)^2 x^2\,m^+}{x\,m^- + \frac{1-\varepsilon_y}{\varepsilon_x}\,x\,m^+} \\ &= x\,\frac{m^- + \left(\frac{1-\varepsilon_y}{\varepsilon_x}\right)^2 m^+}{m^- + \frac{1-\varepsilon_y}{\varepsilon_x}\,m^+} \end{aligned} \quad (2)$$

For the special value $x = \varepsilon_x$ (point P), we get :

$$\begin{cases} n & = \varepsilon_x\,m^- + (1-\varepsilon_y)\,m^+ \\ |\cap_n^{i,j}| & = \varepsilon_x^2\,m^- + (1-\varepsilon_y)^2\,m^+ \end{cases} \quad (3)$$

corresponding to the value on the y-axis :

$$\frac{|\cap_n^{i,j}|}{n} = \frac{\varepsilon_x^2\,m^- + (1-\varepsilon_y)^2\,m^+}{\varepsilon_x\,m^- + (1-\varepsilon_y)\,m^+} \quad (4)$$

2. 2nd phase : $\varepsilon_x < x$.

$$\begin{cases} n & = x\,m^- + \left[(1-\varepsilon_y) + \frac{\varepsilon_y}{1-\varepsilon_x}(x-\varepsilon_x)\right] m^+ \\ |\cap_n^{i,j}| & = x^2\,m^- + \left[(1-\varepsilon_y) + \frac{\varepsilon_y}{1-\varepsilon_x}(x-\varepsilon_x)\right]^2 m^+ \end{cases} \quad (5)$$

giving, for the second part of the curve, the equation :

$$\frac{|\cap_n^{i,j}|}{n} = \frac{x^2\,m^- + \left[(1-\varepsilon_y) + \frac{\varepsilon_y}{1-\varepsilon_x}(x-\varepsilon_x)\right]^2 m^+}{x\,m^- + \left[(1-\varepsilon_y) + \frac{\varepsilon_y}{1-\varepsilon_x}(x-\varepsilon_x)\right] m^+} \quad (6)$$

These equations give the most probable value for $\frac{|\cap_n^{i,j}|}{n}$, as shown on the right hand side of Figure 4. While computed from an idealized model, this curve is in good accordance with empirical observations.

2.4 The algorithm

The selection of the useful base scoring functions is done according to algorithm1. First, the functions of $\mathcal{F}$ are ranked according to their degree of surcorrelation as measured by the difference of correlation on the data set $\mathcal{S}$ and the mean correlation computed on the random samples $\mathcal{S}_0$. The functions that have their surcorrelation with at least another function above a given threshold $\tau_{\text{min\_overcor}}$ are retained in the set $\mathcal{F}'$.

It is then highly desirable to keep only the functions that are as decorrelated as possible among themselves,

the idea being to get a *basis* of functions. Ideally, one would compare each function in $\mathcal{F}'$ to all others and retain only those that have a minimal surcorrelation with their counterparts, and then create a minimal set of such functions. This would necessitate a costly constraint satisfaction phase. The current implementation of the algorithm simplifies the problem by retaining the functions for which the *sum* of the surcorrelations with all other functions in $\mathcal{F}'$ is above a threshold $\tau_{\text{sum\_overcorr}}$.

Algorithm 1: Selection of "good enough" base scoring functions

Input: The data set $\mathcal{S}$
The set $\mathcal{F}$ of the base scoring functions
Output: A subset $\mathcal{F}'' \in \mathcal{F}$ of base functions

Generation of N random samples $\mathcal{S}_0$;

forall the *pairs of scoring functions* $(f_i, f_j)_{(i \neq j)} \in \mathcal{F}$ **do**
  compute the over-correlation of (f_i, f_j) on $\mathcal{S}$ compared to the mean correlation on the samples $\mathcal{S}_0$
end forall

Select the scoring functions $f_i \in \mathcal{F}$ with over-correlation $\geq \tau_{\text{min\_overcor}}$: producing $\mathcal{F}'$

Initialization : $\mathcal{F}'' = \emptyset$

forall the $f_i \in \mathcal{F}'$ **do**
  if $\sum_{j \neq i} overcorr(f_i, f_j) \geq \tau_{sum\_overcorr}$ **then**
    Put f_i in $\mathcal{F}''$
end forall

3 Experimental studies

These experiments address the question as to which extent the proposed method is able to select relevant evaluation functions in $\mathcal{F}$, that is functions which tend to put the '+' elements before the '−' elements in their ranking of $\mathcal{S}$. In other words, if we were to know the class of the elements in $\mathcal{S}$ and thus be able to compute ROC curves, would the selected functions have an AUC > 0.5 ?

In order to test for this, we have realized experiments with artificial data. The data where generated using two probability distributions over the input space $\mathbb{R}^d$ (here $d = 20$) : distribution $\mathbf{P}^+_{\mathcal{X}}$ for the '+' instances and distribution $\mathbf{P}^-_{\mathcal{X}}$ for the '+' instances. In the experiments reported here we have used two Gaussian distributions with means separated by a euclidian distance of 3. The difficulty of the task was controlled by adding noise of varying standard deviation σ to the data points ($\sigma = 1.5$, 2.5, 3.5 and 4.5).

The relative proportion of the class '+' to the total number of elements m was set to varying values : $40/320 = 1/8 \approx 12\%$, $80/320 = 1/4 = 25\%$, $120/320 = 3/8 \approx 37\%$ and $160/320 = 50\%$.

For the set $\mathcal{F}$ of evaluation functions, we used known methods such as ANOVA and Relief, and we build variations over these functions, for instance by varying the distance used (e.g. ℓ_0, ℓ_1, and up to ℓ_5) or the number of neighbor elements taken into account (specially in Relief) or by varying the coordinates of the data points in $\mathbb{R}^d$ that were taken into account by the functions. We also built functions of our own, relying on the computation of various "bizarre" statistics over the coordinates of the data points. In the reported experiments, we used 24 such functions, to which we added 20 "opposite" functions that returned just the opposite of one the previous 24 functions (these functions were thus supposed to be negatively aligned, to some extent, with the ideal sorting function). And finally, we added a random evaluation function to the set $\mathcal{F}$ in order to test if it would effectively be eliminated by our algorithm. We therefore considered 45 functions in $\mathcal{F}$.

In each experiment, a sample $\mathcal{S}$ is generated as explain above, and 100 random samples (of the same size) are used in order to estimate the a priori correlation between the evaluation functions.

Table 1 reports the minimal AUC (auc_m) and the maximum AUC (auc^M) for the functions in $\mathcal{F}$. Likewise, it reports the minimal AUC (auc_m), the maximal AUC (auc^M) and the mean AUC ($\overline{auc}$) for the functions selected by the method : in $\mathcal{F}''$. Finally, the last column gives the AUC obtained by combining the results of the evaluation functions selected in $\mathcal{F}''$ (see Section 4 for an explanation).

The first thing to notice is that, in all cases, the worse selected function has an AUC > 0.5, which means that the method is able to eliminate all evaluation functions negatively correlated or uncorrelated with the ideal function. On the other hand, it might also happen that the best evaluation function of $\mathcal{F}$ is not selected in $\mathcal{F}''$ (see for instance the line $\sigma = 4.5$ and $m^+/m = 80/320$). This happens when this unknown best function is not sufficiently overcorrelated with another function in $\mathcal{F}$.

In addition to the results reported in Table 1, the experiments show that the number of selected functions in $\mathcal{F}''$ tends to decrease when the difficulty of the problem increases (increasing value of σ). Specifi-

σ	$\frac{m^+}{m}$	Before selection		After selection			
		auc_m	auc^M	auc_m	auc^M	$\overline{auc}$	AUC comb
1.5	$\frac{40}{320}$	0 ± 0	1 ± 0	$\mathbf{0.92} \pm 0.03$	1 ± 0	0.98 ± 0.01	$\mathbf{1} \pm 0$
	$\frac{80}{320}$	0 ± 0	1 ± 0	$\mathbf{0.87} \pm 0.06$	1 ± 0	0.97 ± 0.01	$\mathbf{1} \pm 0$
	$\frac{120}{320}$	0 ± 0	1 ± 0	$\mathbf{0.84} \pm 0.07$	1 ± 0	0.95 ± 0.01	$\mathbf{1} \pm 0$
2.5	$\frac{40}{320}$	0.02 ± 0.01	0.98 ± 0.01	$\mathbf{0.94} \pm 0.03$	0.98 ± 0.00	0.96 ± 0.02	$\mathbf{0.98} \pm 0.01$
	$\frac{80}{320}$	0.03 ± 0.01	0.98 ± 0.01	$\mathbf{0.85} \pm 0.05$	0.98 ± 0.01	0.91 ± 0.02	$\mathbf{0.97} \pm 0.01$
	$\frac{120}{320}$	0.03 ± 0.01	0.98 ± 0.01	$\mathbf{0.76} \pm 0.03$	0.98 ± 0.01	0.88 ± 0.02	$\mathbf{0.97} \pm 0.01$
	$\frac{160}{320}$	0.03 ± 0.01	0.98 ± 0.01	$\mathbf{0.73} \pm 0.04$	0.97 ± 0.01	0.85 ± 0.02	$\mathbf{0.95} \pm 0.01$
3.5	$\frac{40}{320}$	0.09 ± 0.02	0.91 ± 0.02	$\mathbf{0.75} \pm 0.06$	0.90 ± 0.03	0.83 ± 0.01	$\mathbf{0.90} \pm 0.03$
	$\frac{80}{320}$	0.09 ± 0.02	0.92 ± 0.02	$\mathbf{0.65} \pm 0.05$	0.92 ± 0.02	0.79 ± 0.02	$\mathbf{0.90} \pm 0.02$
	$\frac{120}{320}$	0.09 ± 0.02	0.91 ± 0.01	$\mathbf{0.64} \pm 0.04$	0.91 ± 0.01	0.77 ± 0.02	$\mathbf{0.89} \pm 0.02$
	$\frac{160}{320}$	0.10 ± 0.01	0.91 ± 0.02	$\mathbf{0.63} \pm 0.03$	0.91 ± 0.02	0.76 ± 0.02	$\mathbf{0.88} \pm 0.02$
4.5	$\frac{40}{320}$	0.13 ± 0.02	0.86 ± 0.02	$\mathbf{0.67} \pm 0.03$	0.86 ± 0.02	0.76 ± 0.02	$\mathbf{0.86} \pm 0.02$
	$\frac{80}{320}$	0.15 ± 0.02	0.85 ± 0.02	$\mathbf{0.65} \pm 0.03$	0.84 ± 0.03	0.75 ± 0.02	$\mathbf{0.84} \pm 0.03$
	$\frac{120}{320}$	0.15 ± 0.02	0.84 ± 0.02	$\mathbf{0.62} \pm 0.06$	0.84 ± 0.02	0.73 ± 0.03	$\mathbf{0.84} \pm 0.02$
	$\frac{160}{320}$	0.15 ± 0.01	0.85 ± 0.01	$\mathbf{0.61} \pm 0.03$	0.85 ± 0.01	0.72 ± 0.02	$\mathbf{0.83} \pm 0.03$

TABLE 1 – Experimental results in function of the noise parameter σ and the proportion of the class '+'.

cally, $|\mathcal{F}''| \approx 10$ for simple problems ($\sigma = 1.5$), whereas $|\mathcal{F}''| \approx 5.5$ for difficult problems ($\sigma = 4.5$). This is due to the fact that the noise in the data tend to decrease the difference of correlation as measured on $\mathcal{S}$ and on the random samples $\mathcal{S}_0$.

Finally, we made experiments where $\mathcal{F}$ contained only evaluation functions of AUC ≤ 0.5. Then, in approximately 60% of the experiments, the method select between 2 and 4 functions, which therefore tend to put negative examples before positive ones. Interestingly, it suffices that 3 or 4 functions of AUC > 0.5 be put in $\mathcal{F}$ to prevent this behavior to happen. This is because there is a dissymmetry between class '+' and '−', the last one being generally supposed to represent a majority of the data set $\mathcal{S}$.

4 A method for combining results

Each of the selected base function $f_i \in \mathcal{F}''$ outputs a ranking over the elements $\mathbf{x}$ of the set $\mathcal{S}$ ordered by decreasing value of $f_i(\mathbf{x})$. It is therefore possible to associate each element $\mathbf{x}$ with a vector of coordinates $(f_i(\mathbf{x}))_{f_i \in \mathcal{F}''}$ (the scores are normalized in $(0, 1)$).

In this redescription space $\Phi(\mathcal{X})$, the data points of class '+' tend to be aligned around the diagonal since they are points for which the selected base functions are correlated, and they are distant from the origin because they have high values of the evaluation functions (see Figure 5). Therefore, one method to sort the '+' points from the '−' ones is to project the points in $\Phi(\mathcal{X})$ over the principle diagonal and to use a threshold to decide the class of the examples. This method gives the same weight to all selected base function. In our experiments, we have weighted the functions according to their total surcorrelation with the other functions of $\mathcal{F}''$. We use an exponential function of this surcorrelation. This amounts to project the data points over a biased diagonal of $\Phi(\mathcal{X})$.

While this method gives good results on empirical evaluations (see Table 1 last column), it does not lead to easy to interpret regularities. The method developed in the next section aims at tackling this challenge.

5 Towards interpretable combinations of selected features

Assuming that there exists a class of m^+ objects of interest from a distribution $\mathbf{P}^+_{\mathcal{X}}$ and a class of m^- other objects in the data set $\mathcal{S}$ from a distribution $\mathbf{P}^-_{\mathcal{X}}$, is there any hope of identifying the objects of the class '+'? It all depends on the number and properties of the evaluation functions contained in $\mathcal{F}$.

As a start, let us suppose that a pair of functions (f_i, f_j) exists in $\mathcal{F}$ such that each function is somewhat "aligned" with the ideal function that would separate

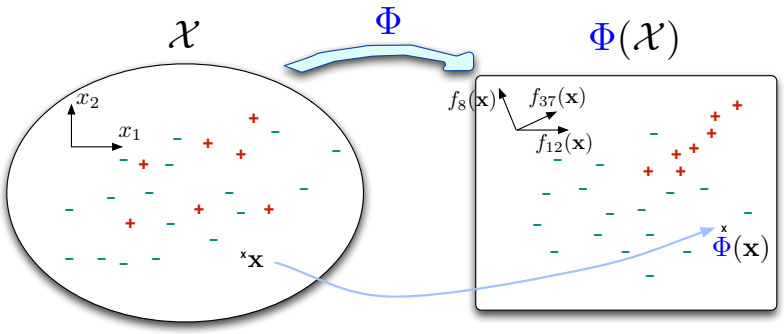

FIGURE 5 – Data points of $\mathcal{S}$ are projected from the input space $\mathcal{X}$ to a redescription space $\Phi(\mathcal{X})$.

the positive objects from the other. This translates in the fact that both f_i and f_j have a AUC > 0.5 as measured with respect to the unknown classes of objects. We will further assume, as in Section 2.3, that their ROC curve exhibit the simple profile of Figure 4 on the left.

One can compute the ROC curve obtained when considering the intersection $\frac{|\cap_n^{i,j}|}{n}$ of the top$_n$ of each function.

Using the equations of Section 2.3, one obtains :

For $x \leq \varepsilon_x$:

$$|\cap_n^{i,j}| = \underbrace{x^2 m^-}_{FP} + \underbrace{\left[\frac{1-\varepsilon_y}{\varepsilon_x}\right]^2 x^2 m^+}_{TP} \qquad (7)$$

and for $x > \varepsilon_x$:

$$|\cap_n^{i,j}| = \underbrace{x^2 m^-}_{FP} + \underbrace{\left[(1-\varepsilon_y) + \frac{\varepsilon_y}{1-\varepsilon_x}(x-\varepsilon_x)\right]^2 m^+}_{TP} \qquad (8)$$

As an illustration, Figure 6 shows the ROC curve of each score function and the ROC curve when using $|\cap_n^{i,j}|$.

What is interesting is that while the AUC of the function $\frac{|\cap_n^{i,j}|}{n}$ is not much larger than the AUC of each base function, its slope in the left part of the curve is much steeper. That means that the precision in this part seems very much improved. Does the theoretical analysis confirms this ? Let us see how the precision and recall evolve when one goes from a random selection of objects in $\mathcal{S}$ (stage 0), to using the base score function (stage 1), up to using the function $|\cap_n^{i,j}|$ (stage 2).

1. *Stage 0.* We suppose that a fraction η of the m objects are randomly selected in $\mathcal{S}$ and are assigned to the class '+'. We let : $m^- = \alpha\, m^+$, with $\alpha \geq 0$ and $\varepsilon_x = \beta\,(1-\varepsilon_y)$ with $0 \leq \beta < 1$ (note that $0 \leq \beta < 1$ entails an AUC > 0.5 while $\beta > 1$ entails an AUC < 0.5). Then, we get the precision (*prec.*) and recall :

$$\text{prec.} = \frac{\text{TP}}{\text{TP}+\text{FP}} = \frac{\eta\, m^+}{\eta(m^+ + m^-)} = \frac{1}{1+\alpha}$$
$$\text{recall} = \frac{\text{TP}}{\text{TP}+\text{FN}} = \frac{\eta\, m^+}{m^+} = \eta$$

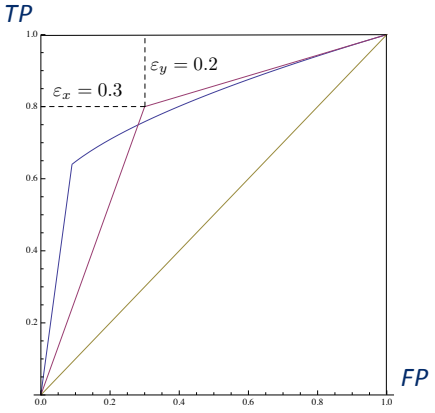

FIGURE 6 – In red (the top curve starting at $FP = 0.3$), the ROC curve of the two base score functions f_i and f_j. In blue (the top curve before $FP \approx 0.3$), the ROC curve of the function $\frac{|\cap_n^{i,j}|}{n}$ when n varies from 0 to m.

2. *Stage 1.* We look at the point on the ROC curve that maximizes precision and recall : $x = \varepsilon_x$ on

Figure 4.

$$\begin{aligned}\text{prec.} &= \frac{(1-\varepsilon_y)\,m^+}{(1-\varepsilon_y)\,m^+ + \varepsilon_x\,\alpha\,m^+} \\ &= \frac{1-\varepsilon_y}{1-\varepsilon_y+\alpha\,\beta\,(1-\varepsilon_y)} = \frac{1}{1+\alpha\,\beta} \\ \text{recall} &= \frac{(1-\varepsilon_y)\,m^+}{m^+} = 1-\varepsilon_y\end{aligned}$$

3. *Stage 2.* We now use the function $|\cap_n^{i,j}|$, again at the point with best precision and recall.

$$\begin{aligned}\text{prec.} &= \frac{(1-\varepsilon_y)^2\,m^+}{(1-\varepsilon_y)^2\,m^+ + {\varepsilon_x}^2\,\alpha\,m^+} \\ &= \frac{(1-\varepsilon_y)^2}{(1-\varepsilon_y)^2+\alpha\,\beta^2\,(1-\varepsilon_y)^2} = \frac{1}{1+\alpha\,\beta^2} \\ \text{recall} &= \frac{(1-\varepsilon_y)^2\,m^+}{m^+} = (1-\varepsilon_y)^2\end{aligned}$$

It is apparent that at each stage one looses on the recall, meaning that a smaller part of the class '+' gets recognized. At the same time, the precision increases, and this all the more that $\beta = \varepsilon_x/(1-\varepsilon_y)$ gets smaller, which corresponds to better base score functions.

It can indeed be shown that, assuming that the base score functions are independent (a priori uncorraletad), one can further consider higher order intersections of the top_n ranked objects by each base functions, getting for order k intersections :

$$\begin{aligned}\text{prec.} &= \frac{(1-\varepsilon_y)^k\,m^+}{(1-\varepsilon_y)^k\,m^+ + {\varepsilon_x}^k\,\alpha\,m^+} = \frac{1}{1+\alpha\,\beta^k} \\ \text{recall} &= \frac{(1-\varepsilon_y)^k\,m^+}{m^+} = (1-\varepsilon_y)^k\end{aligned}$$

For a precision at least *prec*, one should use intersections of order k, with :

$$k \geq \frac{\log\frac{1-prec}{\alpha\,prec}}{\log\beta}$$

For instance, if one wants a precision of at least 0.9, with twice as much negative objects than positive ones : $\alpha = 2$ and $\varepsilon_x = \beta\,(1-\varepsilon_y)$ (corresponding to an AUC $= \frac{1}{2}[1+(1-\beta)(1-\varepsilon_y)]$), with $\beta = 0.5$, one should consider intersections of order at least $k = 4$.

By taking different and independent such intersections of order k, denoted $|\cap_n^{(k)}|$, one can increase the recall. Taking the union of N of these subsets gives a set of size approximated by :

$$l = N|\cap_n^{(k)}| - \binom{N}{2}\frac{|\cap_n^{(k)}|^2}{m} + \binom{N}{3}\frac{|\cap_n^{(k)}|^3}{m^2} + \mathcal{O}(|\cap_n^{(k)}|^4)$$

To first order then, it suffices to take N subsets to multiply the recall by N.

As an illustration, suppose that we have a sufficient number of base score functions selected in $\mathcal{F}''$ by our selection algorithm, such that for these functions, $\varepsilon_y = 0.8$, and $\beta = 0.5$ which means that $\varepsilon_x = 0.4$ and the AUC $= \frac{1}{2}[1+(1-\beta)(1-\varepsilon_y)]) = \frac{1}{2}[1+(1-0.5)(1-0.2)]) = 0.7$. In addition, suppose $\alpha = 2$ (twice as much objects of the class '-' than objects of the class '+'), then, in order to get a precision of 0.9 and a recall of 0.9 also, we need to consider intersections of order 4 as seen above.

Since, $|\cap_n^{(k)}| \approx {\varepsilon_x}^k m^- + (1-\varepsilon_y)^k m^+ = {\varepsilon_x}^k \alpha\,m^+ + (1-\varepsilon_y)^k m^+ = m^+[(1+\frac{1}{1+2^{k-1}})(1-\varepsilon_y)^k$ and we want to cover 0.9 m^+ objects of the class '+', we need :

$$\frac{0.9\,m^+}{m^+[(1+\frac{1}{1+2^{k-1}})(1-\varepsilon_y)^k]} \tag{9}$$

intersections of order $k : \cap_n^{(k)}$. With the values considered, this gives approximately 2. In other words, in this example, a disjunction of two conjunctions, each one of them involving the intersection of the top_n of four score functions, will provide a subset of objects, with recall ≈ 0.9 and precision ≈ 0.9.

Let us recap the **lessons from this section**. In principle, assuming that the initial set of score functions is sufficiently well provided with functions of well-behaved ROC curves (characterized by the parameters ε_x and ε_y), it is possible to find disjunctions of intersections of order k such that a given level of recall and precision be met.

Each intersection $\cap_n^{(k)}$ involves the top_n ranked objects by the score functions considered. The value of n actually corresponds to a threshold which defines what is recognized as a '+' object by the score function and what is recognized as a '−' object. In this way, we can consider a score function together with a threshold as a predicate. For instance, we could have the $\text{ANOVA}_{(150)}$ predicate which retain as positive the objects that have a value above the value obtained by the 150th element of $\mathcal{S}$.

Figure 7 shows how, in the space of the rankings by two selected score functions (here the functions ANOVA and RELIEF_{2-2}), it is possible to isolate perfectly the positive data points from the negative ones, using the AND of the "predicates" $\text{ANOVA}_{(80)}$ and $\text{RELIEF}_{2-2-(80)}$. It is apparent that the ranks given by the two scoring function are uncorrelated within each class of objects, while they agree to distinguish between the two classes.

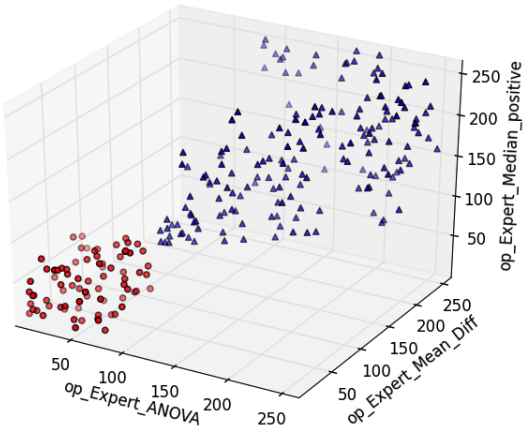

FIGURE 7 – The separation (here perfect) between the class '+' (red dots in the lower left part) and the class '−' (blue dots in the upper right part realized by the AND(ANOVA$_{(80)}$, Mean_Diff$_{(80)}$ Median_Diff$_{(80)}$) function.

The method proposed in this paper has thus the potential to detect relevant evaluation functions and to find combinations of them in order to reach any desired level of precision and recall if the set of functions $\mathcal{F}$ allows it. In addition, by turning the number of top ranked elements for each evaluation function into a threshold, it is possible to obtain predicate functions and thus to transform the combination into a DNF formula.

The method is therefore in principle able to "invent" new predicates and to produces expressions, DNF, that are conducive to easier interpretation.

However, this quite tempting possibility remains to be confirmed before a practical tool ensues. There are indeed several questions that need to be solved. First, and foremost, it is necessary to start with a rich enough set $\mathcal{F}$ of evaluation functions. Experts can often provide ideas for such functions. One can then construct variations around these "seed" functions. This is what we have done in our experiments, for instance by varying the distance, the attributes, and the number of neighbors in the RELIEF evaluation function. Nonetheless, this might still be insufficient to get enough interesting and uncorrelated functions. A second difficulty, linked to the previous one, is that the computational complexity of the method is of the order $O(|\mathcal{F}|^2$, that is a quadratic function of the number of functions considered. This is because the correlations between all pairs of functions need to be computed. A third difficulty lies in the fact that the formula we have given for the precision and recall, require that estimates of the values of α, that is the proportion of negative objects wrt. the positive ones be known. Likewise, one needs also to estimate the characteristics of the base score functions used, that is the parameters ε_x and ε_y. While, we have some ideas as to how to do this, they remain to be validated.

6 Related works

Ensemble methods have first been studied in the context of supervised learning (see [**?**, **?**] for comprehensive studies). It is indeed easy to estimate the performance of the base decision functions with labeled data, using a validation set or cross-validation for instance. Identifying relevant base functions in the context of unsupervised learning is quite a lot murkier. Each potential base function is (e.g. clustering method) is inevitably biased towards some type of regularity in the data, but how can someone measure its adequacy for a given problem ?

Existing works on collaborative clustering do not address this question in all its generality. In fact, they assume that the available base methods are somehow appropriate for the task at hand. The main concern is rather to reduce the *instability* of the methods, meaning the variations in the learned results that can be induced by variations in the data. For this, various authors [**?**, **?**, **?**] suggest to combine, often by a simple vote, the results of various unsupervised techniques. If indeed, generally, the instability is thus reduced, the final result is nonetheless not guaranteed to be good. This is why recent works have put forward the idea of selecting *a priori* the learned results depending on their *quality* and *diversity*. However, these very same criteria expresse themselves subjective biases, and the problem is not solved.

7 Conclusion and future works

In this paper, we addressed the question of developing an ensemble method in the context of 2-classes clustering. Ensemble methods start from an existing hypothesis space, or pool of base decision functions and must solve two problems : first, to identify relevant decision functions for the task at hand, and, second, to combine these functions in order to get a final decision function. In contrast to supervised learning, where labeled data makes it possible to assess the value of decision functions, in unsupervised learning these assessments are much more problematic. This is why existing approaches in collaborative unsupervised learning assume that the base methods are somehow appropriate.

The proposed method departs from this perspective. By measuring the difference of correlation of pairs of base evaluation functions on the data set $\mathcal{S}$ and on random ones, it offers a tool to select base functions that are sensitive to the hidden regularities of the data. The empirical evaluation of the method confirms, thus far, its worth at identifying relevant base functions. This is important in applications, since this allows the user to be less concerned with the exquisite design and tuning of good evaluation functions. It suffices to provide the method with a pool of reasonable (or not) evaluation functions and variants of these.

Our work also tackles the issue of the combination of base functions. In a first approach, we have used a rather traditional technique that weights the base functions depending on their degree of surcorrelation with other base functions. The performances thus empirically obtained are quite good, always at the level of the best base function in the initial set $\mathcal{F}$. We expect that using larger sets $\mathcal{F}$, this type of combination could yield even better results.

However, a theoretical analysis points to another very alluring idea. By looking at the formula for precision and recall when considering intersections of the top_n ranked elements by base functions, it appears that one could hope for two benefits at once. The first is that, provided the set $\mathcal{F}$ of base functions is "rich" enough, it is theoretically possible to reach as high a precision and recall as one desires. The second is that this involves combinations of base functions of the form of DNF, which naturally lead to easier interpretations by the experts. The whole approach is still tentative and has strong links with the notoriously difficult problem of predicate invention. We have underlined the hurdles that remain to be solved in order to get a fully operational method. But we hope this first foray in this direction will stimulate further works that will overcome the problems pointed out and maybe bring about a new set of interesting tools for ensemble methods in the unsupervised setting.

Acknowledgments. Part of this work has been supported by the French ANR project "Coclico" (2013-2016).

Références

[DAR09] Carlotta Domeniconi and Muna Al-Razgan. Weighted cluster ensembles : Methods and analysis. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 2(4) :17, 2009.

[DWH01] Evgenia Dimitriadou, Andreas Weingessel, and Kurt Hornik. Voting-merging : An ensemble method for clustering. In *Artificial Neural Networks—ICANN 2001*, pages 217–224. Springer, 2001.

[Fla12] Peter Flach. *Machine learning : the art and science of algorithms that make sense of data*. Cambridge University Press, 2012.

[JK02] Kalervo Järvelin and Jaana Kekäläinen. Cumulated gain-based evaluation of ir techniques. *ACM Transactions on Information Systems (TOIS)*, 20(4) :422–446, 2002.

[Kon94] Igor Kononenko. Estimating attributes : analysis and extensions of relief. In *Machine Learning : ECML-94*, pages 171–182. Springer, 1994.

[SAVdP08] Yvan Saeys, Thomas Abeel, and Yves Van de Peer. Robust feature selection using ensemble feature selection techniques. In *Machine learning and knowledge discovery in databases (ECML-PKDD-2008)*, pages 313–325. Springer, 2008.

[SF12] Robert E Schapire and Yoav Freund. *Boosting : Foundations and Algorithms*. MIT Press, 2012.

[WLL+13] Yining Wang, Wang Liwei, Yuanzhi Li, Di He, Wei Chen, and Tie-Yan Liu. A theoretical analysis of ndcg ranking measures. In *26th Annual Conference on Learning Theory*, 2013.

[Zho12] Zhi-Hua Zhou. *Ensemble methods : foundations and algorithms*. CRC Press, 2012.