A Distributed Graph Based Approach for Rough Classifications Considering Dominance Relations Between Overlapping Classes

Khalil Laghmari, Mohammed Ramdani, Christophe Marsala

▶ **To cite this version:**

Khalil Laghmari, Mohammed Ramdani, Christophe Marsala. A Distributed Graph Based Approach for Rough Classifications Considering Dominance Relations Between Overlapping Classes. 10th International Conference on Intelligent Systems: Theories and Applications (SITA 2015), Oct 2015, Rabat, Morocco. pp.1-6, 10.1109/SITA.2015.7358388 . hal-01303006

HAL Id: hal-01303006

https://hal.sorbonne-universite.fr/hal-01303006

Submitted on 15 Apr 2016

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

A Distributed Graph Based Approach for Rough Classifications Considering Dominance Relations Between Overlapping Classes

Laghmari.K*, Ramdani.M*, Marsala.C†
* Université Hassan II de Casablanca, FSTM, Laboratoire Informatique de Mohammedia,
BP 146 Mohammedia 20650 Maroc
laghmari.khalil@gmail.com, ramdani@fstm.ac.ma
† Sorbonne Universités, UPMC Univ Paris 06, CNRS, LIP6 UMR 7606,
4 place Jussieu 75005 Paris, France.
christophe.marsala@lip6.fr

***Abstract*—Several data from real world applications involves overlapping classes. Data is allowed to belong to multiple classes with different membership degrees. In this paper, we explore a different concept characterizing social networks, documents, and most of biological and chemical datasets: data could have multiple classes, but dominant classes are better noticed than dominated classes. For example, a document could discuss economy and politics, but it would be more focused on politics. A molecule could have multiple odors, but experts could notice some odors better than others. We are interested in this type of data, where a dominance relation exists between classes. Experts could easily make mistakes because dominated classes are hardly noticed. Data incoherence is a serious problem but not the only one. There is too much irrelevant and redundant attributes. Unfortunately this increases the computational time of generating classifiers. Our first challenge is to find an adapted model to overlapping classes considering dominance relations. The second challenge is to find the most relevant attributes. Finally the third challenge is to ensure that the approach gives results in an acceptable time. We address those challenges by taking advantage of the rough set theory, which is suited for incoherent data and allows multiple classes and attributes selection. The proposed approach works in a parallel and decentralized way to reduce the computational time. We tested it on real chemical data and the collected results are very promising.**

I. Introduction

Classification is a supervised machine learning method where data labels are already known. The process uses a training dataset as input, generates a classifier and then predicts a class label for each unlabeled data. A test dataset is used to compute the classifier accuracy and to validate the predictive model. Sometimes data could have more than one label, this problem is known as overlapping classification. In this paper we first discuss the state of the art for overlapping classification in Section II, then we present our proposed approach in Section III, and finally we discuss our experiment results in Section IV.

II. State of art

In the following, we first discuss mathematical tools applied to overlapping classifications, then we discuss some algorithms designed for dimensionality reduction. Finally we present some tools designed for distributed data classification as a solution to reduce the computational time.

A. The overlapping problem

Building a good classifier is the objective of the classification process. Traditional classifiers work well with simple data with only one possible class, but they produce bad results for complex data with multiple possible classes. The fuzzy sets theory developed by [1] is a popular and a powerful tool for solving the classes overlapping problem. By means of a membership function, data is allowed to be assigned to all classes with different membership degrees that range between *0* and *1*. In recent work [2], a new neuro fuzzy-classification technique is proposed. Experiments proved that this approach produces better results than Radial Basis Function Neural Network [3] or Adaptive Neuro-fuzzy Inference System [4] algorithms. However, even with medium data sizes n, if the size k of possible classes is large, then setting and transposing a $n \times k$ matrix would be very time consuming. The rough sets theory (RST) developed by [5] is a good choice when the objective is reducing the computational time. The RST-based approaches are usually easy to distribute. Each class is defined by means of upper and lower approximations. Depending on the classification certainty, a data is allowed to belong either to exactly the lower approximation of a single class, or to the upper approximation of multiple classes. The RST covers many application domains, especially biology and medicine: In [6] a RST-method is proposed to generate classification rules for Breast cancer data, and more recently in [7] a RST-method has been applied to generate classification rules for Hepatitis C virus dataset. Mixture model approaches are an interesting alternative to fuzzy set and rough set theories. Each class is represented by a distribution. Each data is assumed to be generated by one or more distributions. The objective is to find the relevant parameter values to suit distributions as much as possible. The Expectation-Minimization (EM) [8] algorithm is often used to determine those values. Recently, in [9] a

Gaussian mixture models is proposed to detect overlapped speeches, and the proposed approach was successfully tested on artificial and real datasets. A less flexible alternative could be hierarchical algorithms such as the pyramidal classification [10], the week and the k-week hierarchies [11]. The pyramidal classification allows overlaps only between adjacent classes. The week hierarchies assumes that the intersection of *3* classes is equal to the intersection of just *2* of them. The k-week hierarchies are a generalization of the week hierarchies. They assume that the intersection of *k+1* classes is equal to the intersection of just *k* of them. Those hierarchical classification algorithms reduce clearly the solutions search space. Unfortunately, without any preliminary knowledge about data we could not decide if k-week hierarchies could be adapted to our problem or not. We decided to use rough sets because no preliminary knowledge about data is required, and because the classification could easily be distributed to reduce the computational time.

B. The attributes selection problem

Rough set approaches are based on a discernibility relation. Each minimal attributes subset discerning data is called a reduct. Reducts computation could be done by generating a discernibility matrix and simplifying the extracted boolean function, or by using the concept of positive regions to add or delete attributes from candidate reducts. In [12] attributes selection is performed based on dependency, relevance, and significance criteria. In [13], normalized mutual information and attribute importance criteria are used. The more used attribute for the discernibility purpose is a simple criteria for the attributes selection operation. We have decided to use it because it's not a bad heuristic, and the positive region method could always be applied on the previously selected attributes instead of the set of all attributes.

C. The computational time problem

The RST discernibility function and positive region based methods are computationally expensive. Heuristics and new approaches are needed to obtain results in acceptable time. The MapReduce framework can be used to distribute the feature selection task [14], [15]. The MapReduce framework [16] was successfully tested on very large datasets from different domains [17], [18], [19]. The most known implementations of MapReduce are Hadoop and MongoDB. They are powerful tools but they were designed to handle unstructured data. We consider only structured data in this paper. Neo4j seems to be the right choice for our problem. It's a graph database server designed for big data. A graph database is easier to distribute than a relational database because there is no relational constraints to care about. A graph database has also the advantage of an easy generation of decision trees, and a simple modeling of dominance relations by means of edges. This motivates us to build a distributed graph based approach using the RST.

III. The proposed rough classification approach

In this section, we introduce a mathematical formulation of our problem and we review the necessary mathematical foundations of the RST, then we present our proposed approach.

A. Problem formulation and preliminary knowledge

Let $M = \{m_1, \ldots, m_z\}$ be the set of available machines, $A = \{a_1, \ldots, a_p\}$ be the set of attributes, $(q_j)_{1 \leq j \leq p}$ be the number of possible values for each attribute a_j, $(W_j = \{w_{j1}, w_{jq_j}\})_{1 \leq j \leq p}$ be the set of possible values for each attribute a_j, $C = \{c_1, \ldots, c_k\}$ be the set of classes, X be the set of data, $X_t = \{x_1, \ldots, x_n\} \subset X$ be the training set, and $X_v = X - X_t$ be the validation set. Each data x_i is described by values $(v_{i1}, \ldots, v_{ip}, c_{i1}, \ldots, c_{is})$, $v_{ij} \in W_j \ \forall j \in [1, p]$, $c_{ih} \in C \ \forall h \in [1, s]$. A data x_i could belong to one or more classes from C according to different ranks from $R = \{r_1, \ldots, r_s\}$. A class at a rank r_h dominates its successor at the rank r_{h+1}.The dominance relation gives us information about classes which never coexist, classes always together, strong (dominant) and weak (dominated) classes. Our objective is to find out simple classification rules, taking into consideration the dominance relation, and using the minimal subset of needed attributes.

A rough class is defined by means of upper and lower approximations. The set of all data which can be classified with certainty as belonging to a class c_h is called the lower approximation of c_h, denoted by c_{h*}. The set of all data which can be possibility classified as belonging to c_h is called the upper approximation of c_h denoted by ${c_h}^*$. The set ${c_h}'' = {c_h}^* - c_{h*}$ containing elements which cannot be classified neither as members of c_h nor as members of $\overline{c_h}$ is called the boundary region of c_h. The class c_h is crisp if ${c_h}'' = \emptyset$, otherwise c_h is rough. A rough class can be described by means of some coefficients:

- Quality coefficient of the lower approximation: $\alpha(c_h) = |c_{h*}|/|c_h|$ where $|.|$ denotes cardinality.
- Quality coefficient of the upper approximation: $\beta(c_h) = |c_h|/|{c_h}^*|$.
- Imprecision coefficient: $\gamma(c_h) = |c_{h*}|/|{c_h}^*| = \alpha(c_h) \times \beta(c_h)$.

Let $X^{h,l} = \{x_i \in X_t, c_{ih} = c_l\}$ be the set of data having the class label c_l at the rank r_h. In order to discern for a rank r_h, a class c_l from all other classes using a RST-approach, we could compute a symmetric $|X_t - X^{h,l}| \times |X^{h,l}|$ matrix D, called the discernibility matrix. Where $D_{i,j} = \{a_k \in A, v_{ik} \neq v_{jk}\}$ contains the set of all attributes that could be used to discern data $x_i \in X_t - X^{h,l}$ from data $x_j \in X^{h,l}$. Once the discernibility matrix is constructed, we could extract the called discernibility function $f = \bigwedge_{i<j} (\bigvee_{a_k \in m_{i,j}} (a_k))$, where $\wedge$ and $\vee$ denotes the conjunction and the disjunction operators respectively. Each possible simplification of f is called a reduct. The intersection of all reducts is called the core. A reduct represent a minimal attributes set to discern two datasets. It is used to generate rules based on attributes values. For a specific reduct, a rule for a rank r_h is in the form: if $(\bigwedge_{a_j \in reduct} (a_j = v_j))$ then c_l.

The set of all rules defines the classifier we are looking for. In our proposed approach we generate a decision tree, and since generating all reducts is a NP Hard problem, we generate a single subset of attributes, with good chances that it could be a reduct. We give the needed details in III-B.

B. The proposed approach

Our proposed approach includes 10 steps that are summarized in (Fig. 1). First, the Main Server (M.S) read the initial data (Fig. 1 step 1) and generates the training (X_t) and validation (X_v) sets (Fig. 1 step 2). Next, each DataBase Server (DB.S) receives a subset of the training set (Fig. 1 step 3) and do some preprocessing tasks, like generating new attributes and discretizing values (Fig. 1 step 4). Each DB.S generates a graph from the prepared data (Fig. 1 step 5). The graph database includes data and classes nodes, and two types of edges: Edges to link data with noticed classes, and edges to link dominant classes with dominated classes. Each DB.S generates discernibility edges between specific data pairs(Fig. 1 step 6). Classes discernibility edges (Fig. 1 step 7) and the decision tree (Fig. 1 step 8) are then generated for each rank $r_h \in R$. Finally The M.S generates the global decision tree (Fig. 1 step 9) and validates the model using the validation set (Fig. 1 step 10). Details are given for main steps below.

1) Training set and validation set selection (step 2): In order to build a good classifier without need to a cross validation step, the proposed approach try to generate a training set X_t containing all classes arrangements $(c_{i1}, ..., c_{is}), i \in [1, N]$ existing in the initial data set X. The main server count how many times each arrangement appears in the initial data. The parameter $\lambda = |X_v|/|X|$ defines how many data with a specific classes arrangement should go to the validation set X_v. We start the selection process from the most frequent arrangement, and we stop when X_v reaches the wanted size.

2) Generating Data discernibility edges (step 6): In order to quickly count how many times each attribute was used to discern data, we generate for each data pair $(x_i, x_{i'})_{i<i'}$ a vector $b_{ii'} = (b_{ii'1}, ..., b_{ii'p})$ of booleans instead of a set of selected attributes. The vector $b_{ii'}$ (Fig. 2 step 6) is defined as follows: $\forall j \in [1, p]$ $b_{ii'j} = 1$ if $v_{ij} \neq v_{i'j}$, and $b_{ii'j} = 0$ if not.

3) Generating class discernibility edges (step 7): Some classes could never be noticed at a rank r_h. Also some classes could be noticed only if another class is noticed before at rank r_{h-1}. Considering this, we generate classes discernibility edges (Fig. 2 step 7) between classes for a given r_h and a given class c_f at r_{h-1}. $c_l^{h,f} = X^{h,l} \cap X^{h-1,f}$ denotes the set of data having the class c_l at the rank r_h and the class c_f at the rank r_{h-1}. For each pair $(c_l^{h,f}, c_{l'}^{h,f})_{l<l'}$, we count how many times an attribute a_j, was used to discern data belonging to $c_l^{h,f}$ from data belonging to $c_{l'}^{h,f}$. The following vectors are then generated for each DB.S:

- $(\omega_{ll'j}^{h,f})_{1\leq j\leq p} = (\sum_{\substack{i=1 \\ x_i \in c_l^{h,f}}}^{n-1} \sum_{\substack{i'=i+1 \\ x_{i'} \in c_{l'}^{h,f}}}^{n} b_{ii'j})_{1\leq j\leq p}$
- $(\omega_j^{h,f})_{1\leq j\leq p} = (\sum_{1\leq l<l'\leq k} \omega_{ll'j}^{h,f})_{1\leq j\leq p}$
- $(\omega_j^h)_{1\leq j\leq p} = (\sum_{f=1}^{k} \omega_j^{h,f})_{1\leq j\leq p}$

4) Generating local decision tree (step 8): The M.S collects ω^h vectors from all z DB.S (Fig. 2 step9). $\omega^h(d)$ denotes the vector ω^h generated by the DB.S d. The M.S generates $(\Omega^h)_{1\leq h\leq s}$ vectors as follows: $(\Omega_j^h)_{1\leq j\leq p} = (\sum_{d=1}^{z} \omega_j^h(d))_{1\leq j\leq p}$. The most used attribute $\hat{a}_j^h$ to discern data at the rank r_h corresponds to the attribute with the highest value in the Ω^h vector: $\hat{a}_j^h = arg(\max_{1\leq j'\leq p}(\Omega_{j'}^h))$. Usually we need to select more than one attribute to discern data and classes. Let $\hat{A}^h$ be the set of selected attributes at r_h, and $\hat{J}^h = \{j \in [1, p], a_j \in \hat{A}^h\}$ be the set of corresponding indexes.

We then create a node $F^{h,l}$ (Fig. 2 step8) with parameters:

- r_h: current rank.
- c_f: noticed class at the rank r_{h-1}.
- $c^{h,f} = \bigcup_{1\leq l\leq k} c_l^{h,f}$: The set of data to discern.
- $L^{h,f} = \{l \in [1, k], c_l^{h,f} \neq \emptyset\}$: Indexes corresponding to possible classes at the rank r_h considering the class c_f at the rank r_{h-1}.

F nodes help us at the aggregation step and they allow us to evaluate some coefficients at the validation step. Referenced classes in $L^{h,f}$ should be discerned using the selected attribute $\hat{a}_j^h$. For each value $(w_{je})_{1\leq e\leq q_j}$ we create a node $F^{h,f,w_{je}}$ with the parameters described below:

- $c_l^{h,f,w_{je}} = \{x_i \in c_l^{h,f}, v_{ij} = w_{je}\}$: The set of data to discern at the node $F^{h,f,w_{je}}$.
- $L^{h,f,w_{je}} = \{l \in [1, k], c_l^{h,f,w_{je}} \neq \emptyset\}$: Indexes corresponding to the set of classes to discern at $F^{h,f,w_{je}}$.

Next, for each node $F^{h,f,w_{je}}$, we repeat our method from step 7 (Fig. 2 step 7) using $F^{h,f,w_{je}}$ parameters instead of $F^{h,f}$ parameters. Each time we select a new attribute $\hat{a}_{j'}^h$ to discern data and classes, we add $\hat{a}_{j'}^h$ to $\hat{A}^h$. We stop when classes referenced by $L^{h,f,w_{je}}$ cannot be discerned using the set of available attributes ($\max_{1\leq j\leq p}(\Omega_j^h) = 0$).

Usually in a decision tree, classes are discerned by a selected attribute values at each step, until only isolated classes are kept at the bottom of the decision tree. The proposed approach is different because if only one decision tree couldn't discern a class $(c_l)_{1\leq l\leq k}$ using the current subset of selected attributes, then all decision trees should be extended by selecting an additional attribute, even if some of decision trees could discern c_l without the additional attribute. This redundancy is important to facilitate the global decision tree generation.

A node $F^{h,f,w_{je}}$ on which we stop denoted by $\hat{F}^{h,f,w_{je}}$ represent a rule node. We create edges to all classes referenced by each rule node $(\hat{F}^{h,f,w_{je}})_{1\leq e\leq q_j}$. If a rule node references exactly one class, then it represents a crisp rule. Otherwise it represents a rough rule. Finally we link each class $c_{f'}$ where $f' \in \hat{L}^{h,f,w_{je}}$ with a new node $F^{h+1,f'}$, and we repeat our

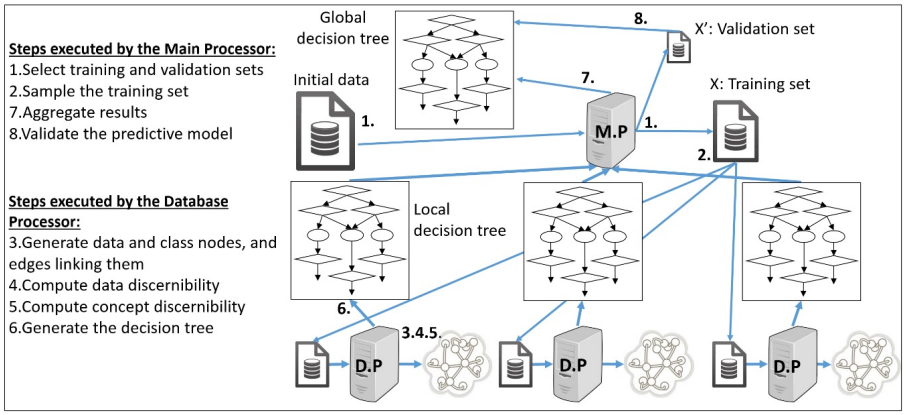

Fig. 1: The proposed approach: main steps

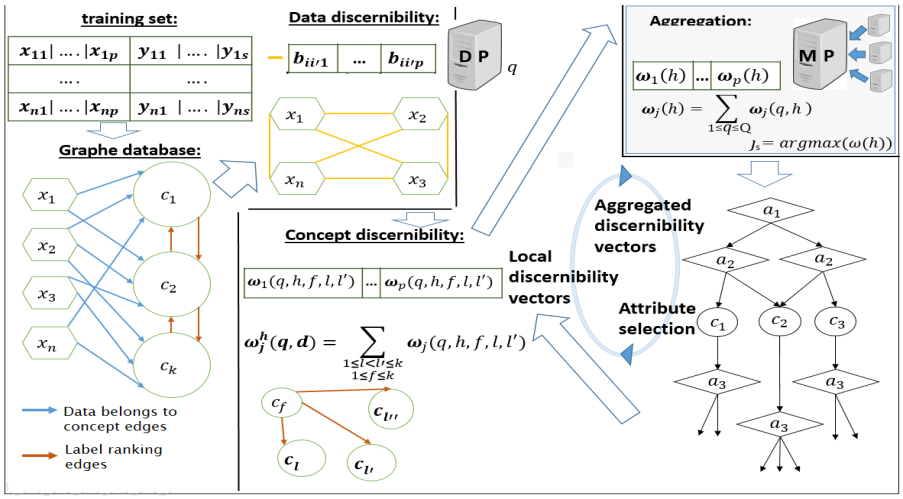

Fig. 2: The proposed approach: steps details

method from $step7$ starting at $F^{h+1,f'}$ instead of $F^{h,f}$. We stop if one of those conditions holds:

- $h = s$: We reached the last rank.
- $f' = 0$: $c_{f'}$ is the empty class, and then c_0 is the corresponding class for all next ranks $(r_{h'})_{h \leq h' \leq s}$.

5) Generating global decision trees (step 9): After generating all local decision trees using the same attributes for each rank, we generate the global decision tree as follows:

- We generate global F nodes: ${F_G}^{h,f,w_{je}}$ taking into consideration all data to discern from all z DB.S: $c_G^{h,f,w_{je}} = \bigcup_{1 \leq d \leq z} c^{h,f,w_{je}}(d)$, and all classes to discern from all z DB.S: $L_G^{h,f,w_{je}} = \bigcup_{1 \leq d \leq z} L^{h,f,w_{je}}(d)$.
- We copy all edges linking F nodes from the z local decision trees (without duplicates) to the global generated decision tree.

A partial view of 2 local decision trees is shown in (Fig. 3). The resulting global decision tree is shown in (Fig. 4).

6) Model validation (step10): Some classes could overlap with too much other classes at different ranks. This make the decision tree larger and less efficient. A solution could be to build another classifier for those classes with higher overlap degrees. The overlap parameter $\theta(c_l)$ for a class c_l represents the proportion of classes that could overlap with c_l. We introduce the parameters defined below:

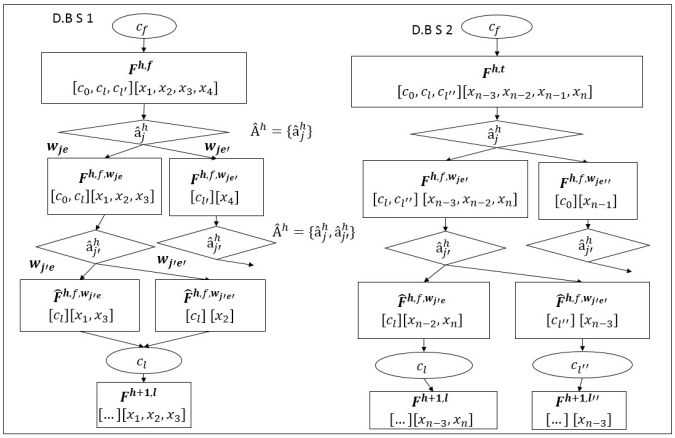

Fig. 3: Local decision trees

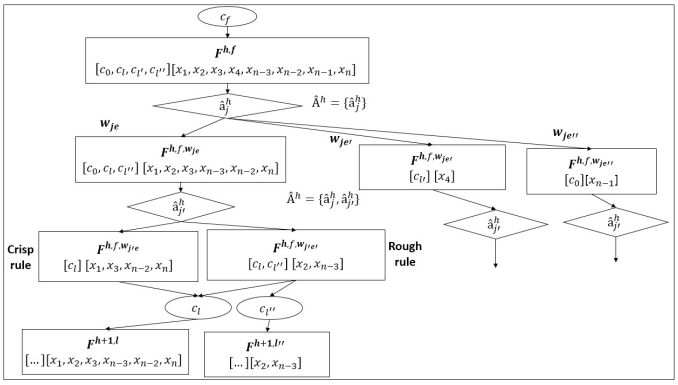

Fig. 4: Global decision tree

data	r_1	r_2	r_3	r_4	r_5	r_6
x_1	HERBAC					
x_2	FLORAL	FRUITY				
x_3	FRUITY	TEA	HERBAC			
x_4	BALSAM	FRUITY	FLORAL	HERBAC		
x_5	HERBAL	FRUITY	FLORAL	ORANGB	JASMIN	
x_6	FRUITY	ROSE	BALSAM	WINEY	LEAFY	FLORAL

TABLE I: Molecules dataset: A view of initial data

- the set of classes which overlap at rules nodes with a class c_l considering only the DB.S d: $\hat{L}_{c_l}(d) = (\bigcup_{1\leq h\leq s} \bigcup_{1\leq f\leq k} \bigcup_{j\in\hat{j}^h} \bigcup_{1\leq e\leq q_j} \{\hat{L}^{h,f,w_{je}} - \{l\}, l \in \hat{L}^{h,f,w_{je}} \text{ and } 2 \leq |\hat{L}^{h,f,w_{je}}|\})$.
- The parameter $\hat{L}_{c_l}$ considering all the z DB.S: $\hat{L}_{c_l} = \bigcup_{1\leq d\leq z} \hat{L}_{c_l}(d)$.
- $X^l = \bigcup_{1\leq h\leq s} X^{h,l}$: The set of data having the class c_l at any rank $(r_h)_{1\leq h\leq s}$.

The overlap parameter $\theta(c_l)$ is then defined as: $\theta(c_l) = |\hat{L}_{c_l}|/|C|$. The lower and the upper approximations of each class c_l are computed on each DB.S d as follows:

- $c_l{''}(d) = \bigcup_{1\leq h\leq s} \bigcup_{1\leq f\leq k} \bigcup_{j\in\hat{j}^h} \bigcup_{1\leq e\leq q_j} \{\hat{c}^{h,f,w_{je}} \cap c_l^{h,f}, 2 \leq |\hat{L}^{h,f,w_{je}}|\}$.
- $c_{l*}(d) = X^l - c_l{''}(d)$.
- $c_l{^*}(d) = c_{l*}(d) \cup c_l{''}(d)$.

The M.S aggregate the previous parameters as follows: $c_{l''} = \bigcup_{1\leq d\leq z} c_l{''}(d)$, $c_{l*} = \bigcap_{1\leq d\leq z} c_{l*}(d)$, $c_l{^*} = c_{l*} \cup c_l{''}$. Finally, the rough coefficients described in III-A are computed.

Next, we let the global decision tree decide classes for data in X_v. Depending on classes precision parameter, we could get in addition of the correct classification, some other possible classifications for each data.

IV. Experiments

The proposed approach has been experimented on molecules data from [20] (TABLE. I). 2000 molecules were used as a learning dataset and 400 molecules were used to validate the model. Each molecule could have one or more odors. The maximum number of odors is 6 per molecule.

rank	selected attr	compared	discernible	indiscernible
1	13	1805296	1798832	6464
2	13	111431	110656	775
3	12	66025	65607	418
4	12	12706	12607	99
5	8	1128	1120	8
6	5	61	60	1

TABLE II: Selected attributes

rank	rules	crisp rules	rough rules	avg rough classes per rule	validation
1	1627	412	1215	4,15	100%
2	1913	408	1505	5,07	32%
3	1669	337	1332	4,83	23%
4	1017	240	777	4,18	56%
5	413	128	285	3,52	87%
6	95	53	42	2,62	98%

TABLE III: Rules validation

Using a local network with 1 server and 10 similar computers, we could sample data to 4 samples of 500 elements. 4 computers were used to store and compute discernibility inside each sample. The left 6 computers were used to compute discernibility between elements from each 2 different samples. Unfortunately initial data don't contain descriptive attributes, but we could generate 15 attributes from only the name and formula of each molecule. Amoung them there is a boolean attribute determining if a molecule is cyclic or not, frequences of some atoms, and the molecular weight. Numerical attributes were discretized to have less than 10 equal range values. Since we didn't address the discretization problem, this is the simplest discretization strategy for us even if it's not the best one.

At the attributes selection step, only 2 attributes were discarded, and a considerable number of data pairs could not

class	α	β	γ	θ
GREEN	0,21	0,21	0,04	0,85
FRUITY	0,26	0,23	0,06	0,78
HERBAC	0,24	0,22	0,05	0,78
...	...	...	...	...
LILAC	0,67	0,57	0,43	0,11
MOSSY	0,5	0,36	0,18	0,1
SULFUR	0,9	0,84	0,76	0,06

TABLE IV: Classes evaluation

rank	classes	α	β	γ	θ
r_1	54	0,24	0,23	0,08	0,45
r_2	70	0,75	0,59	0,48	0,30
r_3	72	0,71	0,72	0,54	0,23
r_4	57	0,79	0,83	0,69	0,12
r_5	25	0,92	0,94	0,88	0,04
r_6	5	0,97	0,97	0,94	0,16

TABLE V: Classes evaluation by rank

	Time (min)		
	Sequential and centralized method	Distributed and parallel method	Saved time
Data discernibility	1844	108	97%
Classes discernibility	1855	55	94%
Total	3699	163	95,6%

TABLE VI: Computational time reduction

be discerned using the selected attributes (TABLE II). The attributes we could generate were not relevant. We obtained too much rough rules (Table III). Some classes overlap with almost all other classes ($\theta \uparrow$) by means of many data ($\beta \downarrow$) (TABLE IV. The results is that classes could not be well discerned and the average of classes imprecision coefficient γ was very low (TABLE V). By using both the selected attributes and the information about the class in the previous rank, we could reduce the average of rough classes per rule(TABLE III). The classifier could always find the correct class for the first rank, because even if classes at r_1 are rough in a high degree, experts don't make mistakes while noticing them. Noticed classes at r_1 are correct. The difficult task for experts is to determine classes at next ranks, The giving classes could be wrong. That explain why the accuracy of ranks r_2 and r_3 is low. The accuracy of ranks $(r_h)_{4 \leq h \leq 6}$ is higher because there is not too much data having more than 4 classes.

In order to evaluate the computational time gain of the proposed approach, we run it on a single machine in a sequential and centralized way, and we turned it on similar 11 machines (10 Database servers + 1 main server). The generated global decision tree by aggregation of local decision trees and the generated decision tree by the centralized method were similar. TABLE. VI shows that the distributed approach could save $95,6\%$ of the centralized approach computational time. This is because except some few simple operations, all operations in the proposed approach could be executed in a parallel and decentralized way.

V. Conclusion and future works

In this paper, an adapted framework to rough classifications with dominance relations is proposed. The proposed approach work on a distributed and parallel way, and as shown in experiments, the computational time is clearly reduced. Using Neo4j server, we could build a special decision tree adapted to our problem definition. Experiment on a real chemical dataset shows that thanks to dominance relations, our model can find the correct classification among some few other possible classifications. In our future works, we will try to reduce the number of needed attributes, by integrating the positive region concept to our model. We also plan to find a good discretization strategy adapted to our problem. Finally, more accurate results would be obtained by generating more descriptive attributes, and by aggregating decision trees corresponding to each rank. The aggregation should be weighted because decision trees at lower ranks are more accurate.

References

[1] L. Zadeh, "Fuzzy sets," *Information and Control*, vol. 8, no. 3, pp. 338 – 353, 1965.

[2] S. Ghosh, S. Biswas, D. Sarkar, and P. P. Sarkar, "A novel neuro-fuzzy classification technique for data mining," *Egyptian Informatics Journal*, vol. 15, no. 3, pp. 129 – 147, 2014.

[3] H. Núñez, C. Angulo, and A. Català, "Rule-based learning systems for support vector machines," *Neural Processing Letters*, vol. 24, no. 1, pp. 1–18, 2006.

[4] J. shing Roger Jang, "Anfis: Adaptive-network-based fuzzy inference system," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 23, pp. 665–685, 1993.

[5] Z. Pawlak, "Rough sets," *International Journal of Computer and Information Science*, vol. 11, pp. 341–356, 1982.

[6] A. E. Hassanien and J. M. H. Ali, "Rough set approach for generation of classification rules of breast cancer data." *Informatica, Lith. Acad. Sci.*, vol. 15, no. 1, pp. 23–38, 2004.

[7] A. Zaki, M. Salama, H. Hefny, and A. Hassanien, "Rough sets-based rules generation approach: A hepatitis c virus data sets," in *Advanced Machine Learning Technologies and Applications*, ser. Communications in Computer and Information Science, A. Hassanien, A.-B. Salem, R. Ramadan, and T.-h. Kim, Eds. Springer Berlin Heidelberg, 2012, vol. 322, pp. 52–59.

[8] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *Journal of the Royal Statistical Society: Series B*, vol. 39, pp. 1–38, 1977.

[9] N. Shokouhi, A. Sathyanarayana, S. Sadjadi, and J. Hansen, "Overlapped-speech detection with applications to driver assessment for in-vehicle active safety systems," in *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*, May 2013, pp. 2834–2838.

[10] B. Patrice, "Etude de la représentation pyramidale," Ph.D. dissertation, Université de Paris-Dauphine, France, 1986.

[11] B. Patrice and M. F. Janowitz, "The k-weak hierarchical representations: an extension of the indexed closed weak hierarchies," *Discrete applied mathematics*, vol. 127, no. 2, pp. 199 – 220, 2003.

[12] P. Maji and P. Garai, "It2 fuzzy-rough sets and max relevance-max significance criterion for attribute selection," *Cybernetics, IEEE Transactions on*, vol. PP, no. 99, pp. 1–1, 2014.

[13] G. Geng, N. Li, and S. Gong, "Feature selection method for network intrusion based on fast attribute reduction of rough set," in *Industrial Control and Electronics Engineering (ICICEE), 2012 International Conference on*, Aug 2012, pp. 530–534.

[14] J. Qian, D. Miao, Z. Zhang, and X. Yue, "Parallel attribute reduction algorithms using mapreduce," *Information Sciences*, vol. 279, no. 0, pp. 671 – 690, 2014.

[15] Z. Sun, "Parallel feature selection based on mapreduce," in *Computer Engineering and Networking*, ser. Lecture Notes in Electrical Engineering, W. Wong and T. Zhu, Eds. Springer International Publishing, 2014, vol. 277, pp. 299–306.

[16] J. Dean and S. Ghemawat, "Mapreduce: Simplified data processing on large clusters," in *Proceedings of the 6th Conference on Symposium on Opearting Systems Design & Implementation - Volume 6*, ser. OSDI'04. Berkeley, CA, USA: USENIX Association, 2004, pp. 10–10.

[17] S. Gugnani, D. Khanolkar, T. Bihany, and N. Khadilkar, "Rule based classification on a multi node scalable hadoop cluster," in *Internet and Distributed Computing Systems*, ser. Lecture Notes in Computer Science, G. Fortino, G. Di Fatta, W. Li, S. Ochoa, A. Cuzzocrea, and M. Pathan, Eds. Springer International Publishing, 2014, vol. 8729, pp. 174–183.

[18] S. Maharjan, P. Shrestha, T. Solorio, and R. Hasan, "A straightforward author profiling approach in mapreduce," in *Advances in Artificial Intelligence – IBERAMIA 2014*, ser. Lecture Notes in Computer Science, A. L. Bazzan and K. Pichara, Eds. Springer International Publishing, 2014, vol. 8864, pp. 95–107.

[19] L. Han and H. Ong, "Parallel data intensive applications using mapreduce: a data mining case study in biomedical sciences," *Cluster Computing*, vol. 18, no. 1, pp. 403–418, 2015.

[20] S. Arctander, *Perfume and Flavor Chemicals: (aroma Chemicals)*, ser. Perfume and Flavor Chemicals: Aroma Chemicals. Allured Publishing Corporation, 1969.