RULE GENERATION AND COMPACTATION IN THE WWTP

David Riaño
Departament d' Enginyeria Informàtica
Universitat Rovira i Virgili
e-mail: drianyo@etse.urv.es

Ulises Cortés
Departament de Llenguatges i Sistemes Informàtics
Universitat Politècnica de Catalunya
e-mail: ia@lsi.upc.es

ABSTRACT

In this paper we discuss our approach to learning classification rules from data. We sketch out two modules of our architecture, namely LINNEO$^+$ and GAR.LINNEO$^+$, which is a knowledge acquisition tool for ill-structured domains automatically generating classes from examples that incrementally works with an unsupervised strategy. LINNEO$^+$'s output, a representation of the conceptual structure of the domain in terms of classes, is the input to GAR that is used to generate a set of classification rules for the original training set. GAR can generate both conjuctive and disjunctive rules.

Herein we present an application of these techniques to data obtained from a real wastewater treatment plant in order to help the construction of a rule base. This rule will be used for a knowledge-based system that aims to supervise the whole process.

1 INTRODUCTION

For a long time, engineers and scientists have been developing complex models to describe the time-varying nature of environmental systems, including wastewater treatment plants (wwtp). New emerging and very innovative technologies in the fields of Artificial Intelligence and Computer Science have lead to the development of promising new concepts and tools, such as real-time and interactive simulation, Knowledge Based Systems (KBS), etc.

More specifically, expert systems, a kind of KBS, are becoming increasingly more popular for the design, operation and control of WWTP. In the development of an expert system, the achievement of procedural knowledge is a very crucial step, usually considered as a *bottle-neck*, even when dealing with a very structured domain. These problems become larger when we think about the environmental engineering in general and, particularly, in the WWTPdomain where information is usually incomplete and oftenly imprecise, and whose study requires specialized knowledge in diverse fields like chemistry, fluid mechanics, biology, mathematics, economy and laws.

An alternative to this expertise approach is the use of a Knowledge Acquisition Tool (KAT) capable of using the knowledge implicitly included in data and other sources of information and automatically build, from it, diagnostic rules. The WWTPis a good target domain for this kind of tools since it is very difficult to obtain descriptions (or rules) directly from the experts because of the diversity, quantity and complexity of the data involved. Moreover, the complex chemical and biological interactions within a WWTP are not always easily described using mathematical models, [12]. A great reliance on experience and intuition is required in many situations.

On the other hand, there exists a lot of recorded information for this domain that can be useful to initialize this KAT process but there is almost no information about the conceptual entities that could permit the modelization (from the AI point of view) nor is there a tradition for compiling the experience of experts using a formalism that could be easily generalized, and therefore, useful to generate a complete knowledge base. It is also difficult to export the experience from one plant to another due the technological or climatic differences, due to the changes on the environmental laws, *etc.* So, a non-supervised machine learning method seems adequate to deal with these problems and, when possible, to organize knowledge in concepts. This task is called *learning from data* and it deals with the task of learning descriptions (traditionally decision trees or rules) from raw data. This paper describes how some machine learning systems are linked together in order to automatically generate classification rules from data. The training set used to illustrate this work is a complex and real data set, obtained from an ill-structured domain, that includes both qualitative and quantitative attributes and whose classification structure is built-up by LINNEO$^+$. μ, the new architecture born from the merging of these systems, is depicted in figure 1. Although some modules are not described in this paper, as the Consensus Module [25] or the clustering module LINNEO$^+$ [2,21], they are well documented in the bibliography. **MILORD II** [14] and **Bolero**[10] are respectively a shell for Expert Systems and a Case-Based System to manage and reason with the acquired knowledge and the new incoming cases.

Our idea is to partially tackle the *bottle-neck* of knowledge acquisition in the construction of KBS and to have a tool that learns from raw data in real domains (*i.e.* creates a concept or builds-up a rule).

The organization of this work is as follows. Section 2 is used to describe the domain and data set that has been used to show the performance of μ within a real WWTP. In section 3 the Automatic Rule Generator Module GAR is introduced together with the description of how conjunctive and disjunctive rules are generated. Later, some details about the complexity of this process is given, and an alternative solution is presented with the extension of the *Duce* operators [24].

Section 4 summarizes the results obtained by μ when this is applied to the problem described in section 2, and shows the levels of knowledge compactation when disjuntive expressions are allowed. In section 5 some conclusions are commented.

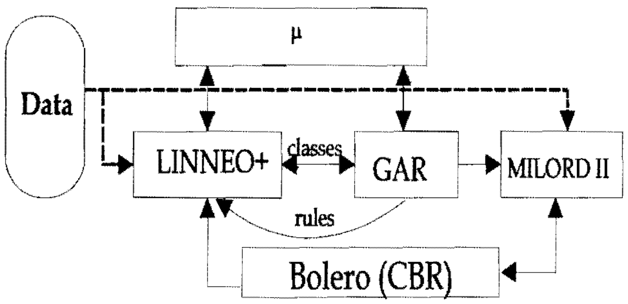

Figure 1: General overview of the μ system.

2 A CASE-STUDY: WASTEWATER TREATMENT PLANT

After some successful tests using the classification module LINNEO$^+$, to generate a concept structure at the level of *operation situations* in a real Wastewater Treatment Plant [21,20,23], now, μ has been applied to generate classification rules for those operation situations.

The studied urban wastewater treatment plant uses a biological process known as activated sludge process. Activated sludge is undoubtedly the most widely extended waste water treatment. In this process, a mixture of several microorganisms transforms the biodegradable pollutant (expressed in units of organic matter as Biological Oxygen Demand (BOD) or Chemical Oxygen Demand (COD)), into a new biomass, with the addition of dissolved oxygen supplied by any aeration system. Previous to the input in the biological reactor, a primary treatment is usually established.

In figure 2, scheme of a plant prototype is presented. As shown, after the primary settler, water is first treated in the bioreactor where, by the action of the microorganisms, the level of substrate is reduced. Next, the water flows to a secondary settler where the biomass sludge settles. Thus, clean water remains at the top of the settler and is carried out of the plant. A fraction of the sludge is returned to the input of the bioreactor in order to maintain an appropriate level of biomass, allowing the oxidation of the organic matter. The rest of the sludge is purged.

Real time control of the process constitutes a quite complex problem due to the lack of reliable instrumentation and the simplicity of the models to describe the microbiological processes that takes place in the bioreactor. In this context, although some advanced control techniques such as predictive control have obtained promising results, they are not able to handle a number of situations that need to consider qualitative knowledge [12]. Consequently, the personal expertise of the plant manager is necessary to attain an efficient management of the process.

Simultaneously to this problem, and taking into account the social importance of this kind of plants in order to preserve the ecological equilibrium of water bodies, a lot of variables related with the organic matter and microorganisms are measured in the plants, giving a lot of information that is difficult to manage.

The plant studied is located in Manresa, a town of 100,000 inhabitants, near Barcelona (Catalonia). The plant treats a flow of 35,000 m^3/day of mainly domestic waste water although waste water from industries located inside the town are also received in the plant.

A set of 38 system variables, 8 of which are quality indicators and 9 of which are percentages of performance indicated with the prefix *Rd*, are measured in several places of the plant (at the input P2 —variable suffixed with E—, after the pretreatment P3 —sufixed with P—, at the input of the biological reactor P4 —suffixed with D— and at the water output of the plant —suffixed with S—), with a daily frequency. In this study the behavior of the plant along 527 days has been considered.

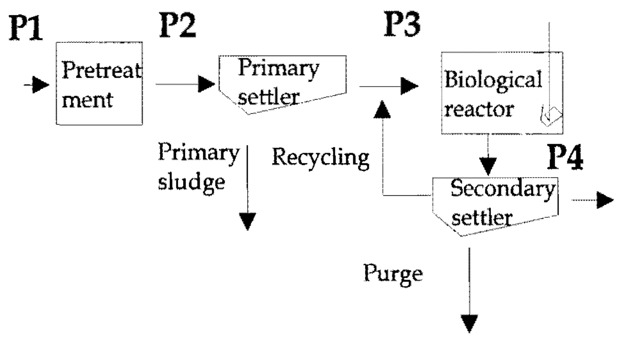

Figure 2: Wastewater treatment process.

The variables measure properties of the water: effluent (Q), zinc (Zn), acid level (Ph), biological oxygen demand (BOD), chemical oxygen demand (COD), suspended solids (SS and VSS), sediments (Sed), and general water conditions (Cond).

Combining the above properties with the four possible sites of measuring (P1, P2, P3, and P4 in figure 2), and with the sort of variable (indicator or performance), the meaning of each of the 38 system variables in table 2 can be interpreted.

Once the whole set of variables is stablished, their values for the plant are observed for 527 consecutive days. All these observations are taken as input data of the clustering module LINNEO$^+$ [1] as figure 1 shows.

The original classification was obtained after setting the radius to 4 —measure of the similitude among the objects of each class [21]—, without using a DT —set of predefined rules to direct the first step in the classification process [2], represented in figure 1 by the rules coming into the LINNEO$^+$ module—, and considering only the attributes whithin the process. The results were the 17 meaningful classes briefly described in table 1 (at the taxonomical level of *operation situation* [22,23]).

As an example of the classification obtained by LINNEO$^+$, the normalized center of the class number 13 is shown in table 2. According to the values of the prototype in that table, the class has been identified as those days in the WWTPwhich reflect a *NORMAL situation* of the plant with normal influent values and with a performance slightly over the average situation obtaining a normal effluent. This interpretation was confirmed when confronted with the daily log of the plant.

The results were used to help the experts to build-up a knowledge base for the automatic control and supervision of the wastewater treatment plant and reported in [20,22,23].

The data used in this example are available in the UCI Repository of Machine Learning Databases and Domain Theories, and can be obtained by *anonymous ftp* from **ftp.ics.uci.edu.**

Class	*Class name*	*Rules*	*Elem.*
1	*NORMAL-1*	86	254
2	*SECONDARY PROBLEM*	1	1
3	*SECONDARYPROBLEM*	1	1
4	*NORMAL-4*	22	81
5	*NORMAL-5*	38	108
6	*SOLIDS CRASH*	1	3
7	*BAD PERFORMANCE*	1	1
8	*STORM-1*	1	1
9	*DAY AFTER STORM*	1	1
10	*NORMAL-10*	18	57
11	*NORMAL-11*	1	3
12	*BAD PERFORMANCE*	1	2
13	*NORMAL-13*	3	10
14	*SOLIDS CRASH*	1	1
15	*STORM-2*	1	1
16	*NORMAL DAY1*	1	1
17	*NORMAL DAY2*	1	1

Table1: General information about classes.

3 The automatic rule generator module

GAR is the module of μ (see figure 1) capable of generating and manipulating rules. Knowledge induced by LINNEO$^+$ in the clustering process must be internally represented into GAR for its future use.

The input to GAR is a set of classes, each one described by both the center of the class (also called concept or prototype), and the list of objects in the class.

GAR generates, for each concept (or class), a set of rules with the purpose of entirely describing such a concept. One of the limitations to the process of rule construction comes from the syntax of the rules, described as follows:

Term = (op *lval* $attribute_i$) | (rel *exp*)

| (range (*val val*) $attribute_i$)

Clause = (or $Term^+$)

Premise = Term | *Clause*

Rule = ($Premise^+ \rightarrow Cj$)

where *op* could be one of "=", "neq, ">",

"$\leq$" , "$\geq$", or "<". The term "(rel *exp*)"

stands for whatever relational expression between attributes; *lval* stands for a non-empty list of modalities in the case of qualitative attributes and a single value in the case of quantitative attributes; $attribute_i$ is the target attribute and, C*j* is a dummy identifier for the set of objects that satisfies this rule. The **"range"** constructor restricts the $attribute_i$ valuations to the interval defined by the pair of values *val*. Clauses permit the introduction of a disjunction of terms, and rules contain a conjunction of premises which are terms or clauses.

Attrib.	Value	Attrib.	Value
Q-E	0.49	VSS-D	0.57
Zn-E	0.03	Sed-D	0.07
pH-E	0.35	Cond-D	0.21
BOD-E	0.17	pH-S	0.33
COD-E	0.23	BOD-S	0.01
SS-E	0.05	COD-S	0.07
VSS-E	0.50	SS-S	0.03
Sed-E	0.06	VSS-S	0.76
Cond-E	0.07	Sed-S	0.07
pH-P	0.20	Cond-S	0.03
BOD-P	0.15	Rd-BOD-P	0.67
SS-P	0.07	Rd-SS-P	0.00
VSS-P	0.50	Rd-Sed-P	0.24
Sed-P	0.04	Rd-BOD-S	0.55
Cond-P	0.07	Rd-COD-S	0.62
pH-D	0.36	Rd-BOD-G	0.86
BOD-D	0.13	Rd-COD-G	0.77
COD-D	0.15	Rd-SS-G	0.90
SS-D	0.12	Rd-SedS-G	0.98

Table2: Normalized Center for the class NORMAL.

GAR inherits this syntax to maintain the coherence and to facilitate the change of information between μ the modules. One can expect LINNEO$^+$ to use the rules produced by GAR to bias a classification or to help to modify some rules given by the expert.

A second limitation, this time more subjective, is the purpose of having rules that are useful, understable and meaningful at the same time. *Useful* in the sense that they are produced to define a future reasoning system; *understable* in the sense that experts have to understand them as a step previous to validation; and *meaningful* to justify the use of GAR as well.

3.1 Conjuntive rules

Mostly, the process of building a descriptive conjunctive rule up can be faced under several strategies, each one producing not necessarily the same rule or set of rules. Some of these strategies are grouped into *specific-to-general* vs. *general-to-specific*, *instance-based* vs. *selectors-based*, and *exhaustive* vs. *heuristic* methods.

Specific-to-general methods [6] start with the most specific rule, which is the result of making a conjunction with all the available terms, and then the methods perform generalizations to include those positive examples missing in the initial description. In *general-to-specific* methods [5,16] an initial empty rule set is successively modified by specialization techniques to adjust the target concept, this time by means of refusing negative examples previously accepted.

Some methods take a disturbing example, i.e. a positive example rejected or a negative example accepted, and perform a modification in the rule set to fix this situation. This is the case of *instance-based* methods like [6]. Alternatively, *selectors-based* methods take terms to modify a deficient rule set ant then they evaluate the modified subset with the training set to decide whether the change is maintained or not.

Finally, we can find methods which search the best rule set *exhaustively* accross the whole searching space defined by the training set for sample-based methods, and by both the available terms, and the way these terms can be combined into rules, for selectors-based methods. Unfortunately, the time complexity of exhaustive search is exponential respect to the size of the searching space.

In the case of *heuristic* methods a criterion is taken to direct the search. Two big alternative criteria have directed the research: those based on statistics [4], and those based on information theory [15, 5, 16].

Focusing our attention on the *specificity* [26] of a rule, generally defined as the percentage of negative examples not explained by the rule, the methods producing rules can be catalogued depending on the specificity of the rules they produce. After the analysis and comparison of conjunctive, k-term-DNF, k-DNF, and k-CNF rules in [17], the following conclusions have been reached:

a) the *effectiveness* of rule generation (defined here as the specificity of a rule normalized in time) is the largest for conjunctive rules [17];

b) when given to the expert, conjunctive rules are qualified as more understable than other sort of rules [23];

c) when applying conjunctive rules, the reasoning process is faster [14]; and

d) conjunctive rules structure knowledge in a more atomic way.

These considerations intuitively justify the production of conjunctive rules, like the majority of the rule learning systems do. Nevertheless, some other considerations which are more *ad hoc* in the μ system (e.g. we want to produce rules which could feed-back the clustering process -see figure 1-) drive our whole system to bear rules following the syntax on section 3.

Use and understanding of WWTPrules was observed to work one in the opposite of the other. This is, short rules perform well in the reasoning but they are hardly accepted by experts as good descriptive rules. In the contrary, long rules are uncomfortable but, at the same time, their meaning is clear to the experts.

Any rule partitions the training set into those examples accepted, and those examples rejected. A conjunctive rule is said to be *the least specific conjunctive rule* (LSC) if none term can be removed from it without a change in the above partition. As a complement, *the most specific conjunctive rule* (MSC) [7] is defined as a conjunctive rule where none term can be added to it without a change in the partition of the training set.

LSC rules can be defended since they repesent a memory safe and a faster reasoning process, though Haussler [7] attacks this position and introduces the MSC rules for the sake of human understanding.

Both standpoints are pursued by GAR under two different working modes. One producing a set of short conjunctive rules, and other one as a step previous to the compactation process described in the next section.

Nevertheless, the computation of *the least specific conjunctive rule* is a NP-complete problem which is heuristically relaxed to a more handled problem via a Hill-climbing algorithm in the form of [4,11] and called *the best-first descriptive conjunctive rule* (BDC). The algorithm in table 3 describes how the BDC rules are obtained in an iterative process. At each iteration, the term with the greatest reduction of negative examples accepted is called to be part of the conjunctive rule. All the terms that, when added to the conjunction, accept none negative example, or do not decrease the number of negative examples accepted, become useless and they are removed from the set of possible terms in the third step.

On the other hand, the MSC rule computation, also a complex problem [7], is approximated by an algorithm that takes the *m* terms of the BDC rule, as well as those feasible terms whose addition to the rule do not modify its *accuracy* [26].

The algorithm in table 4 takes the BDC rule and it adds all the terms in the set of possible terms which do not modify the rule criterion of acceptation or rejection for any example.

One of the main successes of this work is the unification of these two, apparently independent, valuable methods for rule generation under a global algorithm showing the falsehood of such independence. The unification is done in such a natural way that both methods seem to be one the complement of the other.

More explicitely, the BDC rule obtained by the algorithm in table 3 is the conjunctive rule used by the algorithm in table 4 after a slight change in the step 1, which now selects all the terms accepting all the examples and none counterexample.

INPUT: training set, set of possible terms
OUTPUT: the **BDC** rule
1. *Select* the best term. 2. *Add* such term to the up to now conjunctive premise 3. *Reduce* the set of possible terms. 4. *Repeat* steps 1 to 3 while rule in not completed.

Table 3: The BDC algorithm.

INPUT: training set, set of possible terms
OUTPUT: the **MSC** rule
1. *Select* all terms accepting all the examples. 2. *Join* all the above terms under the *and* operator.

Table 4: The MSC algorithms.

These two methods are applied to produce a rule (or a set of rules) for the same concept, which comes in this case from the WWTPdomain when classified with LINNEO$^+$ [23], and only considering the attributes relevant for the characterization (in this case DBO-P, DBO-D, SSV-D, and ZN-E [1]). The *complementarity* of both BDC and MSC methods, is empirically depicted in table 5.a, where the first column shows the rule permises obtained for isolate BDC, isolate MSC, and combined BDC+MSC methods. The second column contains the *positive examples* accepted by the above rule premises (true positives), and the third column all those *negative examples* not rejected (false positives).

Apparently, the combined performance BDC+MSC (third row) produces a *worst* premise, if compared with the one produced by the single BDC algorithm (first row), because that one describes the same positive examples, and negative examples, but it needs more information to do the work. But this is not so if we want to generate safe methods. Consider, for instance, the benefits of taking into account the combined performance of both methods, i.e. BDC+MSC, when a new object O=[DBO-P = 0.3, DBO-D = 0.6, ...] is asked to be within the class, or not. Pure methods will thoughtlessly answer 'yes', but the combined method will want to know about the attribute ZN-E for coming to a conclusion, which is a more proper behaviour if we confront O with the examples in table 5.b.

Some other considerations can be taken into account and are expressed in the next lines in the form of properties that each of the methods furnish to the symbolic unification, showing the complementarity of both methodologies.

Three basic features can be remarked in the compilation of BDC expressions. One concerning the *low computational cost*, when this method is compared with other traditional searching algorithms. This benefit, someone can think, comes to the detriment of the *goodness of the results*, assumption whose falsehood has been empirically shown[2].

The third feature is *minimality*. BDC rules are built using Hill Climbing which stops when no improvement is feasible at that point. The final BDC rules are then concrete and experts read them easily, although they rarely accept them as good descriptive rules.

Apart from those previous features, BDC rules present several lacks that the complementation whit the MSC

	Conjunction	*Positive examples*	*Negative examples*
Pure BDC	(≥ (0.22) DBO-P) (≥ (0.5) DBO-D)	*d-29/1/91 d-31/1/91*	*none*
Pure MSC	(≥ (0.22) DBO-P)	*all*	*all*
BDC+MSC	(≥ (0.22) DBO-P) (≥(0.5) DBO-D) (range(0.05 0.10)ZN-E	*d-29/1/91 d-31/1/91*	*none*

(a). BDC and MSC rules.

	DBO-P	DBO-D	SSV-D	ZN-E
d-29/1/91	0.4659793814433	0.6293436293436	0.6378446115288	0.07485029940120
d-31/1/91	0.4371134020618	0.5289575289575	0.7192982456140	0.07185628742515
d-10/12/90	0.3591904993755	0.3397683397683	0.7305764411027	0.02964071856287

	DBO-P	DBO-D	SSV-D	ZN-E
d-2/5/91	0.2226804123711	0.2548262548262	0.6616541353383	0.11976047904191
d-6/3/90	0.3591904993755	0.3720026926440	0.6015037593985	0.04191616766467
d-27/5/91	0.1938144329897	0.2277992277992	0.5112781954887	0.05988023952096

(b). Examples and Counterexamples description.

Table 5: Combined performance of BDC and MSC methods.

[1]DBO-P, DBO-D, are respectively the *biological oxigen demand* at the primary and secondary settler (see figure2), SSV-D the percentage of floating solids at the secondary setter, and ZN-E the degree of input *zinc*.

[2]During the testing of this algorithm, three different domains (*marine sponges* [2], *metal illnesses* [19], and *wastewater treatment plants* [23]) were chosen with equivalent good results.

algorithm helps to overpass. They are abstracted here in the form of four considerations: initially MSC *completes the* BDC *algorithm* in the case that few positive examples are available, e.g. in the extreme of one only example to be described,

d-13/3/90, BDC produces the rule

((= (1406.0) COND-P) -> SECONDARY PROBLEMS)

which is extended by MSC

((< 36.4 RD-SSED-G) (= (43.0) RD-SS-G)

(= (36.0) RD-DQO-G) (< 19.6 RD-DBO-G)

(= (45.8) RD-SS-P) (< 0.6 RD-DBO-P)

(= (1406.0) COND-P) -> SECONDARY PROBLEMS)

This is reflected in the fact that the mixture of both methods allows a *more accurate description* of the concept in the line stablished by table 5.a , and by the assertion 'MSC behaves in a closer way to how humans do' [3].

In fact, the property that most entrusts us to use the combined form is that it *makes the further construction of or-clauses easier* in the form discussed in the next section.

3.2 DISJUNCTIVE CLAUSES

In the above section, conjunctive expressions have been chosen appealing to the argument of the best *effectiveness*. Despite the truity of this position, it can receive several criticisms. One of them comes related to the horizontal (number of terms per rule) and vertical (number of rules) size of the set of rules. If only conjunctive rules are permited, both horizontal and vertical sizes use to be great. In front of this situation, Shapiro [24] defined *Duce*, a machine learning system based on the truth-preserving transformation of a set of conjunctive rules into a more compact set of rules.

These transformations are synthesized into the set of operators given in table 6. **Inter and Intra-construction** permits the introduction of a new concept Z? with the common terms of the initial rules. **Absortion** represents the extreme case of inter-construction where one rule subsumes the other one completely. **Identification** searches for a relation between the different terms of the rules (**C D E**, and **Y**). **Dicotomization** is related to the case that rules concluding positively (**X**) and negatively (**-X**) coexist. And **truncation** is used to forget the different terms of very similar rules.

Inter-construction	Intra-construction	Absortion
B C D E -> X A B D F -> Y	B C D E -> X A B D F -> X	A B C D E -> X A B C -> Y
— — — — — — —	— — — — — — —	— — — — — — —
C E Z? -> X A F Z? -> Y B D -> Z?	B D Z? -> X C E -> Z? A F -> Z?	Y D E -> X A B C -> Y
Identification	**Dicotomization**	**Truncation**
A B C D E -> X A B Y -> X	A B C D -> X A C J K -> -X A B C L -> - X	A B C D -> X A C J K -> X
— — — — — — —	— — — — — — —	— — — — — — —
A B Y -> X C D E -> Y	A C Z? -> X A C -Z -> -X B D -> Z? J K -> -Z? B L -> -Z?	A C -> X

Table 6: Shapiro transformations.

Some of the operators in table 6 are the foundation for *or*-compactations as we will see later. Disjunctive clauses here are basically used to compact knowledge. Two sort of methods can be pointed out for such purpose: methods *a priori* where rules are directly produced containing disjunctive clauses [17,18] and methods *a posteriori* where rules are initially borne in conjunctive form and compacted afterwards to disjunctive expressions [9,13]. GAR is able to produce disjunctive clauses *a priory*[17], and *a posteriory* by the use of the operators in table 7, which are inspired in those of Shapiro.

The operators, extended to manage *or* connectors and depicted in table 7, are a transformation of those suggested by Shapiro.

It is easily verifiable that compactations will be more breathtaking for long rules since the possibility of sharing descriptors, i.e. $|I'|$ or common part of the rules, is greater.

Among the six rules in table 6, we are particularly interested in those that conclude about the same, and only with positive conclusions, i.e. *Intra-construction, Identification and Truncation.* All three rules are modified to accept *or*-operators, keeping in mind the initial idea of Shapiro.

Look at the similarities between **Intra-construction*** and **Inter-construction** defined in tables 6 and 7 respectively, and how this similarity comes reflected in the number of symbols reduced. Furthermore, the **Identification*** transformation is only safely applicable when the symbol **Y** is not found within the other rules, and the rule produced by the last operator must be verified not to be in contradiction with other rules in the rule base.

The respective symbol reduction comes expressed by the equations:

$$V_{intra*} = |I'| - 1) \,.\, (|R'| - 1) - 2 = V_{inter}$$
$$V_{ident*} = |I'| \,.\, (|R'| - 1) = V_{ident}$$
$$V_{trunc*} = \mathrm{total}(R') - |I'| - 1 = V_{trunc}$$

where R' stands for the subset of the rule base R to be modified, I' for the common subset of all the bodies of rules within R', and *total* (R') represents the number of descriptors in R'.

Extending the set of possible operators in *Duce* [13] with the above three new or-operators and modifying the treatment of rules to accept disjunctions, the system *Duce* can come to grips with the compactation of and/or-rule bases within its initial time cost.

Nevertheless, the above modifications are feasible only in the assumption that *or*-operators are present in the set of rules to simplify, which is not the case in the process of conjunctive rule learning. This gap is easily filled with the incorporation of three new **disjuntion-introductor** operators (see table 8), with a saving of

$$V_{di1} = (|I' \,.\, (|R'| - 1),$$
$$V_{di2} = (|I'| + 1) \,.\, (|R'| - 1) - 3, \text{ and}$$
$$V_{di3} = (|I'| - 1) \,.\, (|R'| - 1) - 2.$$

This methodology has been proved hightly time consuming and useless for medium and large sets of rules, but very fuitful for small sets of rules where the degree of compactation is of about 10% and the time needed not very high.

Intra-construction*	Identification*	Truncation *
C (B + D) E ->X	(A + B) C D E -> X	B (A + C) D -> X
A (B + D) F ->X	(A + B) Y-> X	(A + C) J K -> X
— — — — — —	— — — — — — —	— — — — — — —
C E Z? -> X	(A + B) Y-> X	A + C ->X
A F Z? -> X	C D E -> Y	
B + D ->Z?		

Table 7: New operators for or-clauses treatment.

disjuntion-introductor-1	disjunction-introductor-2	disjunction-introductor-3
A B C -> X	A B C E -> X	A B C D -> X
A D C -> X	A B D E -> X	A E F D -> X
	A B Y -> Y	
A (B + D) C -> X	A B Y -> X	A (Y? + Z?) D -> X
	(C + D) E -> Y	B C -> Y?
		E F -> Z?

Table 8: Transformations to introduce disjunctive expressions.

A third alternative approach to or-clauses is to transform the set of rules into a decision tree [8] and transform this tree to contain *external* or operators.

3.3 Complexity of rules generation

Here the temporal cost of producing the set of rules describing a class (or concept) is analyzed in the worst case of asymptotic time.

Since one rule describes at least one positive example, the temporal cost would be, in the worst case, o times the temporal cost of compiling a rule. This compilation is done following the four steps of the BDC algorithm which are the selection of the best term (as the number of terms is proportional to the number of features, the final cost is $O(f\,.\,o)$). After that, the incorporation of the term to the premise and the reduction of possible terms are respectively $O(f\,.\,o)$ and $O(f)$. Thus, the global cost of the BDC algorithm is:

$$O(o\,.\,(o\,.\,f + o\,.\,f + f)) = O(o^2\,.\,f)$$

Furthermore, the MSC process is computed in $O(f\,.\,o)$ and, whether it is also applied together with the BDC process the cost is:

$$O(o^2\,.\,f + o\,.\,f) = O(o^2\,.\,f)$$

When operators in tables 6,7 and 8 are applied, any subset of rules within the rule base R is candidate for the application of one of the 12 operators. Thus the search-space [13] for the *best* operator application is of size $2^{|R|}$, the size of the power set of R.

4 Results

Let us give some ideas about the kind of rules that are generated by the automatic rule generator module GAR attending the classification in table 1.

In our case the goal is just to classify *situations* within the plant's day-to-day operation conditions. An alternative objective is to generate rules to *predict* the plant's operation out from some changes in the tendency of a set of given attributes. This objective could be achieved through the use of some background knowledge as for example to avoid, during the rule generation process, all the attributes related with the plant's output and identifying each class with a certain behaviour.

Three considerations araise at the sight of table 1, and divide the classes into differentiate groups.

- Classes where the MSC part of the general algorithm in section 3.1 affect the rules.
- Classes where the disjunction transformations in section 3.2 are feasible.
- Classes where the use of disjunction transformation is innapropriate, alternative methodologies are needed.

Concerning the first item, only the marginal classes like *SECONDARY PROBLEMS*, whose description appears in section 3.1 as a rule, take profit of this methodology. Alternatively, LINNEO^{+}'s classes 6, 11and 12 (i.e. *SOLIDS CRASH, NORMAL 11, and BAD PERFORMANCE* are particular cases where this consideration was unsuccessfully expected to happen.

4.1 Compactation

For this particular work, the second consideration is where the most interesting classes are. Classes with few rules where the transformations to compact conjunctive rules (table 6), to introduce or-clauses (table 8), or to modify disjunctive rules (table 7) does not spend much time.

For example, when the BDC+MSC algorithm (see § 3.1) is applied to learn the concept underlying in the class *NORMAL-13* given by the system LINNEO$^+$, the following three conjunctive rules are obtained;

```
((RANGE (126.92 170.27) DQO-D)
(RANGE (100.75 117.24) DBO-P)
(RANGE ( 7.46  7.61) PH-E)
(RANGE ( 7.49  7.60) PH-P)
-> NORMAL-13)

((< 7.52 PH-D) (< 41.74 SSV-P)
(RANGE (134.35 289.64) SS-E)
-> NORMAL-13)

((RANGE (100.75 117.24) DBO-P)
(RANGE (782.09 907.10) COND-P)
-> NORMAL-13)
```

meaning the last one, that days whose DBO-P index is between 100.75 and 117.24, and whose **COND-P** index is between 782.09 and 907.10, are days which represents a *NORMAL-13* situation, i.e. they are in the 13th class.

A semantical analysis of these rules, within the **wastewater plant** tested problem, shows their *sensitivity* (percentage of the correctly described elements) indexes to be 50%, 40%, and 50% respectively. **Truncation** and **Identification** are two possible transformations of this small set of rules with a safe of 6 and 1 symbols, respectively, when applied to the above first and third rules. With these rules the ten elements of *NORMAL-13* are fully covered.

However, when we want to obtain relevant examples of how the transformations in tables 6 y 7 are applied, a more difficult class must be chosen, e.g. *NORMAL-10,* which needs 18 conjunctive rules to completely describe the 57 days (see table 9 for more details.

The iterative application of the operators in table 6 produces an averaged reduction of classes *NORMAL-4, NORMAL-10, and NORMAL-13* of 8%. This reduction is up to 11.27% when only the size of the premises is considered.

	class NORMAL-10 (57 elements)																	
Rule number	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
Correct elements	5	7	5	2	3	4	2	4	2	6	2	3	2	3	3	2	2	6
Wrong elements	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Sensitivity (%)	8	12	8	3	5	7	3	7	3	10	3	5	3	5	5	3	3	10

Table 9: Rule-to rule description of class NORMAL-10.

4.2 The analysis of *NORMAL-10*

Concretely, and for class *NORMAL-10* in table 9, the average number of conditions in the premises passes from 7.82 before the compactation, to 5.36 after the compactation. This is a reduction of about 3 conditions per rule from 38 attributes. The number of rules, instead, grows up from 18 to 23 and five new concepts (see table10), some of them well recognized by the experts as **dense-water** (this is water with a great quantity of biomass), **good-extreme-performance** (standing for a good performance of the global treatment of the water in extreme circumstances), and **normal-performance**, appear. Thus, the **normal-performance** is represented by the two last rules in table 10, the first rule reflecting a normal-performance when not much work is deserved and, the second, when the water is very dense (dirty) and the plant achieves between 90.07 and 99.90 % of global efficiency.

Some concepts are hardly recognized by the expert as typical cases, e.g. **CONCEPT-2**, and **CONCEPT-4**.

((> 80.38 SSV-D)(> 73.12 SSV-E) -> dense-water)

((< 340.81 DQO-D)(> 91.94 RDSS-G)(< 39.61 RD-DBO-P)(> 8.14 PH-E)
-> CONCEPT-2)

((> 99.90 RD-SSED-G)(< 77.40 SSV-D)(< 0.54 SED-D)(range (80.44 83.03)
RD-DQO-G)(< 1779.42 COND-D)(> 33869.71 Q-E)(< 1751.60 COND-E)
-> CONCEPT-2)

((range (90.53 91.94) RD-SS-G)(> 99.90 RD-SSED-G)(> 109.31 SS-D)
(> 299.60 DBO-P) -> good-extreme-performance)

((< 90.53 RD-SS-G)(range (99.07 99.90) RD-SSED-G)
(< 1779.42 COND-D)(< 5.42 SED-P)(range (32066.87 33869.71) Q-E)
(< 340.81 DQO-D)(< 497.28 DQO-E) -> good-extreme-performance)

((< 0.54 SED-D) normal-performance -> CONCEPT-4)

((> 1939.99 COND-P)(> 99.90 RD-SSED-G)(< 80.44 RD-DQO-G)(< 204.13 SS-E)
(< 32066.87 Q-E)(< 497.28 DQO-E) -> CONCEPT-4)

((< 8.04 PH-D)(> 83.03 RD-DQO-G)(> 1934.33 COND-D)(< 204.13 SS-E)
-> normal-performance)

((range (90.07 99.90) RD-SSED-G)(range (153.36 166.01) DBO-D) dense-water
-> normal-performance)

Table 10: New concepts in NORMAL-10 after compactation.

The new concepts make the description of the class easier, e.g. rule number 1, initially in the form

((< 340.81 DQO-D)(> 91.94 RD-SS-G)
(< 1773.82 COND-P) (< 39.61 RD-DBO-P)
(> 8.14 PH-E) (> 8.15 PH-P)(> 8.09 PH-D)
-> NORMAL-10),

after the description of the concept **CONCEPT-4**, is reduced by its use to the form

((< 1773.82 COND-P) (> 8.15 PH-P)
(> 8.09 PH-D) CONCEPT-4 -> NORMAL-10).

In a second stage, new compactations are possible through the introduction of disjunctions. Despite an average compactation of 14.33% after having applied operations on tables 7,8 and, the results does not comply with the requirements of *understability* and *meaningfulness* in section 3. This is, rules obtained are hardly understable by the experts and the reasoning is assumed bizarre.

5 Conclusions

The problem of finding conjunctive expressions to represent the classification process has been proved to have good approximative solutions using *hill-climbing* techniques. A representation is obtained much faster and the goodness of the results are comparable to the goodness of the best alternative. In fact, the possible errors commited by the greedy decisions are mended in the long term with the incorporation of new extra rules.

One similar solution for *or*-representations remains unsolved, but the application of Shapiro transformations, combined with the ones introduced here, represents a first aim at the problem solution. However, a stronger effort is needed to improve such methods to increase their understanding.

In the future we will explore the introduction of some bias in the process, using the typicallity of the objects within a class and thus make easier the construction of rules. Also we will intend to use these ideas to generate rules for a hierarchy of classes.

μ has been used to generate classification rules from data in other domains with promising results, for example classification and diagnostic of mental illnesses [19] and classification of marine sponges[2]. Nevertheless, much work remains undone in the prosess of joining the μ system (LINNEO$^+$ + GAR) with the rest of the components in figure 1.

6 Acknowledgements

The authors are grateful to P. Serra, whose skilful comments and interpretations of the results have been of great wealth in this work. We also feel in debt to him for supplying the data related to the wastewater treatment plant at Manresa, Spain. On the technical side, J. Béjar must be mentioned as the producer of the classification, main source of the study here. Also to M. Sánchez, whose description of the domain was considered while section refplanta was being written.

This work was developed with the support of the European Community, under the contract VIM: CHRX-CT93-0401.

References

[1] J. Béjar. LINNEO: a classification methodology for ill-structured domains.Technical Report LSI-93-22-R, Facultat d'Informàtica de Barcelona, 1993.

[2] J. Béjar, U. Cortés, and M. Domingo. Using domain theories to bias classification processes in ill-domains. In *Proceedings of the 4th Iberoamerican Congress on Artificial Intelligence (IBERAMIA '94)*, 1994.

[3] J. S. Bruner, J. Goodnow, and G. A. Austin. *A study in thinking*. New York Wiley, 1956.

[4] J. Cendrowska. PRISM: An algorithm for inducing modular rules. *International Journal on Man Machine Studies*, 27, 1987.

[5] P. Clark and T. Niblett. The CN2 induction algorithm. *Machine Learning*, 3:261—271, 1989.

[6] P. Domingos. Unifying instance-based and rule-based induction. *Machine Learning*, 24, 1996.

[7] D. Haussler. Quantifying inductive bias: Artificial intelligence algorithms and Valiant's learning framework. *Artificial Intelligence*, 36:177-221, 1988.

[8] L. F. Imam and R. S. Michalski. Learning decision trees from decision rules: a method and initial results from a comparative study. *Journal of Intelligent Information Systems*, 2:279-304, 1993.

[9]S. Kundu. Discovering new concepts in rule learnings. In *Proceedings of the ISAI, Mexico*, 1993.

[10] B. López and E. Plaza. Case-based planning for medical diagnosis. In *Proceedings on the Int. Symposium of Methodologies for Intelligent Sistems, ISMIS'93, University of Todheim, Norway, J. morowski and Z. W. Ras Eds.*, pages 96-105, 1993.

[11] A. J. Mooney. Integrating theory and data in category learning. *The psicology of learning and motivation*, 29, 1993.

[12] R. Moreno, C. de Prada, J. Lafuente, M.Poch, and G. Montagne. Non linear predictive control of dissolved oxygen in the activated sludge process. In *ICCAFT 5/IFAC-BIO 2 Conference. Keystoke (USA)*, 1992.

[13] S. Muggleton. *Inductive Acquisition of Expert Knowledge*, chapter 9, Duce, pages 153—171. Ed. Addison Wesley Publishing Co., 1990.

[14] J. Puyol. *Modularization, Uncertainty, Reflective Control and Deduction by Specification in Milord II, a language for knowledge-based sysmtems.* PhD thesis, Departament d'Informàtica, Universitat Autònoma de Barcelona, 1994.

[15] J. R. Quinlan. Induction of decision trees. *Machine Learning*, 1:81-106, 1986.

[16] J. R. Quinlan. *C4.5: programs for machine learning.* Morgan Kaufmann Publishers, 1993.

[17] D. Riaño. Automatic knowledge generation from data in clasification domains. Master's thesis, Facultat d'Informàtica de Barcelona. Universitat Politécnica de Catalunya, 1994.

[18] D. Riaño. Knowledge abstraction using heuristic search: A specialization approach. Technical report, Centrum voor Wiskunde in Informatica, Amsterdam. CS-R9469, 1994.

[19] E. Rojo. *Aplicación del software LINNEO a la clasificación de transtornos mentales.* PhD thesis, Divisió de Ciencies de la Salut. Facultat de Medicina. Universitat de Barcelona, 1993.

[20] M. Sànchez, Ll. Belanche, P. Serra, and U. Cortés. A knowledge-based system for the diagnosis of waste-water treatment plant. In *Proceedings of the Fifth international conference of industrial and engineering applications of AI and Expert Systems IEA/AIE-92. Paderborn, Germany.* Springer-Verlag, 1992.

[21] M.Sánchez, U. Cortés, J. De Gracia, J. Lafuente, and M. Poch. Concept formation in wwtp by means of classification techniques: a compared study. *Applied Intelligence*, 1996 (accepted).

[22] P. Serra. *Desenvolupament d'un sistema basat en el coneixement per al control i supervisió de plantes depuradores d'aigües residuals urbanes.* PhD thesis, Departament de Química. Facultat de Ciencies. Universitat Autònoma de Barcelona, 1994.

[23] P. Serra, M. Sánchez, J. Lafuente, U.Cortés, and M. Poch. DEPUR: a knowledge-based tool for wastewater treatment plants. *Engineering Applications of Artificial Intelligence*, 7(1):23—30, 1994.

[24]A. D. Shapiro. *Structured Induction in Expert Systems.* Addison-Wesley, 1987.

[25] V. Torra and U. Cortés. Towards an automatic consensus generator tool: EGAC. *IEEE* transactions on Systems, Man and Cybernetics, 25(5):888—894, 1995.

[26] S. M. Weiss. *Computer systems that learn.* Morgan Kaufmann Publishers, Palo Alto, CA, 1990.

__David Riaño.__ Received a B.Sc. in Computer Science in 1992, a M.Sc. in Computer Science in 1994, both from the Universitat Politècnica deCatalunya (UPC). Since 1992, he is Associate Professor for Technical Studies in the Computer Engineering Department at the Universitat Rovira i Virgili (URV). Now, he is about to finish his Ph.D. Thesis in Artificial Intelligence. He is member of AEPIA (Spanish Association for Artificial Intelligence), and pioneer member of ACIA (Catalan Association for Artificial Intelligence). He has been working in several projects supported by the Catalonia government, by the Spanish council, and by the EU. He has some publications, conference papers, and other contributions. His main research topics are Machine Learning, Rule Induction and Processing, and Data Mining.

__Ulises Cortés.__ Received a B.Sc. in Industrial and Systems Engineering from the Instituto Tecnológico y de Estudios Superiores de Monterrey (ITESM) in 1982, and a Ph.D. from the Universitat Politècnica de Catalunya (UPC) in 1984. He is an Associate Professor in the Software Department of the UPC since 1988, Head of the Artificial Intelligence Ph.D. program and Vice-dean of the International Affairs of the Faculty of Computer Science of Barcelona (FIB). He participates in the ECC-funded Interest Group on EuLisp an is the Spanish representative onth e JTC1/SC22/WG16/LISP ISO Group. He is member of the AEPIA (Spanish Association for Artificial Intelligence). He is pioneer member of ACIA (Catalan Association of Artificial Intelligence). He is member of SMIA (Mexican Society of Artificial Intelligence). He has an extensive list of publications, conference papers, and tutorial and supervisory contributions. His main research topics are machine learning, knowledge acquisition and LISP-like languages.