Learning rules within the framework of environmental sciences

D. Riaño
Departament d'Enginyeria Informàtica
Universitat Rovira i Virgili
Carretera de Salou s/n, 43006 Tarragona, Spain
drianyo@etse.urv.es

Abstract. Machine Learning is defended as a suitable mechanism to deal with the control of environmental domains. Particularly, we use the system GAR to learn a set of descriptive rules such that they identify the internal states of a wastewater treatment plant. We also propose this mechanism for classifying the huge amount of possible inputs of the plant into a short number of input types in order to construct a state-transition diagram which can be used to control the plant.

Keywords: Machine Learning, Rule-based Systems, State-based Control Systems, Wastewater Treatment Plants and Environmental Engineering.

1 Introduction

There are many environmental domains that need particular treatments. Nevertheless, these domains use to have common features that make them to have similar control processes. Here, we highlight two of those features that we consider representative of environmental sciences and, at the same time, source of difficulty for a control process. On the one hand, they use to be *dynamic* domains in which changes follows one another in time. On the other hand, they are also *interdisciplinary* in the sense that they use to integrate knowledge and experiences which come from separated fields like Chemistry, Biology, Meteorology, Physics, Economics, Law, etc.

The control process acts on the problem by means of sensors and actuators. Sensors supply information to the control systems that decides which are the actions to perform on the actuators. In this process, we can distinguish three basic components which are the input to the system, the state of the system, and the action that the system executes. That is to say, we can ususally represent environmental systems as state-transition diagrams.

In Artificial Intelligence this behaviour is implemented by means of an Expert System. Artificial Intelligence has been traditionally involved in and worried about supplying reliable mechanisms in order to make the environmental control process as automatic as possible [Dym and Levitt, 1991, Hudson et al., 1997] and Expert Systems are one of these successful mechanisms that have received much attention because of its great aptitude for representing and deducing environmental knowledge.

Control Theory offers an alternative representation of environmental systems as *state-transition diagrams*, transition networks, or automatons [Gimeno et al., 1998] in which the states of the system are the nodes of the state-transition diagram that react in front of the sensor inputs triggering some actions and eventually changing the state.

State transition diagrams are represented as labeled directed graphs in which the nodes stand for the states of the system that is being represented and the arrows connecting nodes stand for transitions between states. Furthermore, the arrows are labeled with the events that promote the system to change its state.

Since environmental systems are dynamic and sensor-based, they use to manage a great amount of situations (i.e. states) and information (i.e. measures). Therefore, if we want it to be efficient we must identify states and input situations in order to simplify the control of the system and also to reduce its response time without loosing reliability.

In this paper we apply the learning system GAR [Riaño and Cortés, 1997] to the construction of rules that represent, on the one hand states and, on the other input classes. In section 2 the learning process is described. In section 3 it is applied to the Wastewater Treatment Plant WWTP domain, and the last section gives some conclusions of the work.

2 The learning process - GAR

Taking decisions to control a system is something which is based on the recognition of the state of the system. State recognition is based on the mixture of experience and sensor observation. While experience can be acquired from the observation of cases (inductive learning), sensor observation is a tedious task that should be automated. In this section we introduce machine learning as a feasible way of both acquiring experience and also constructing an automatic process that takes the values in the sensors and deduces which is the actual state of the system. Bearing these goals in mind, first we propose a rule-based representation, and second we introduce a system which is used to learn the rules.

Rules are a representation structure which is very common in the Machine Learning community because they possess some desirable features.

- *Modularity.* Rules represent the global knowledge as the union of partial *pseudo*-independent knowledges. For example, if we want to represent the knowledge involved in the control of a wastewater treatment plant we can separate the knowledge that describes *normal* days from the knowledges that represent *storm* days, *low-influent* days, etc. Furthermore, we can also separate one knowledge into its components by means of different rules, as rule (1) which represents those normal days in which the plant does not have much work or rule (2) which represents those normal days in which the water is very dirty (i.e. `dense-water` means that there are many suspended solids) but the plant has a pretty good global performance.

```
[RD-DQO-G > 83] & [COND-D > 1934] & [SS-E < 204] => NORMAL        (1)
[RD-SED-G > 90] & [DBO-D ∈ (153 166)] & dense-water => NORMAL     (2)
```

- *Knowledge-engineering.* Rules are very easy to create, interpret, verify, handle, and use. That is to say, knowledge engineers, domain experts, and final users are capable of creating rules as background knowledge, interpreting them when they are automatically generated, verifying that rules represent what they are supposed to represent, handling rules in the sense of editing and ordering them, and using rules with a standard reasoning processes.

- *Usefulness.* Rules are a representation structure which is well accepted by important Artificial Intelligence systems like Expert Systems. Moreover, rules have a simple structure that, on the one hand can be easily adapted to the requirements of many intelligent systems and, on the other hand can be extended with complementary methodologies like fuzzy logic or causal networks.

- *Increasing.* Several rule bases representing complementary knowledges can be merged into one. Therefore, if we want to represent a complex knowledge by means of rules, we can start with a kernel of rules that represent the basic concepts and then extend them or refine them by successive approximations to the target knowledge.

If we want to represent a certain domain knowledge by means of a rule-based system we must use a rule learning process which observes a sample of the domain cases, solves some inherent to the sample problems, and constructs a set of rules describing the sample. In supervised learning, those cases represent examples of the situations or concepts that we want to learn. On the contrary, in unsupervised learning, no help is supplied about what the cases in the sample represent and, therefore, we must use some automatic concept induction system like LINNEO+ [Sánchez et al., 1997]. For supervised samples, there are three most frequent approaches to learn rules [Riaño, 1997]: the *direct* methods, the *tree-based* methods, and the *selector-based* methods. Direct methods transform the sample into a set of rules, e.g. CN2 [Clark and Niblett, 1989]; tree-based methods transform the sample into a decision tree which is then converted into a set of rules, e.g. C4.5RULES [Quinlan, 1993]; and selector-based methods learn the elements required to make the rule premises (i.e. *selectors*) and then make the rules, e.g. GAR.

GAR [Riaño and Cortés, 1997] is a multistage rule learning process that generates conjunctive rules in a wide variety of ways. Figure 1 shows this process as a diagram of boxes. The boxes represent the sort of data that GAR can manage, and the arrows the methods that GAR uses to transform one sort of data into other one. Though we can interact with the process at any point in the diagram, the complete process starts with the repeated observation of all the domain sensors. GAR can storage this unsupervised information in a database in the form of a table in which columns represent sensors and each row contains a snapshot of the sensors in an instant, then GAR uses the clustering algorithm LINNEO+ [Sánchez et al., 1997] in order to supervise the information. If some domain expert is

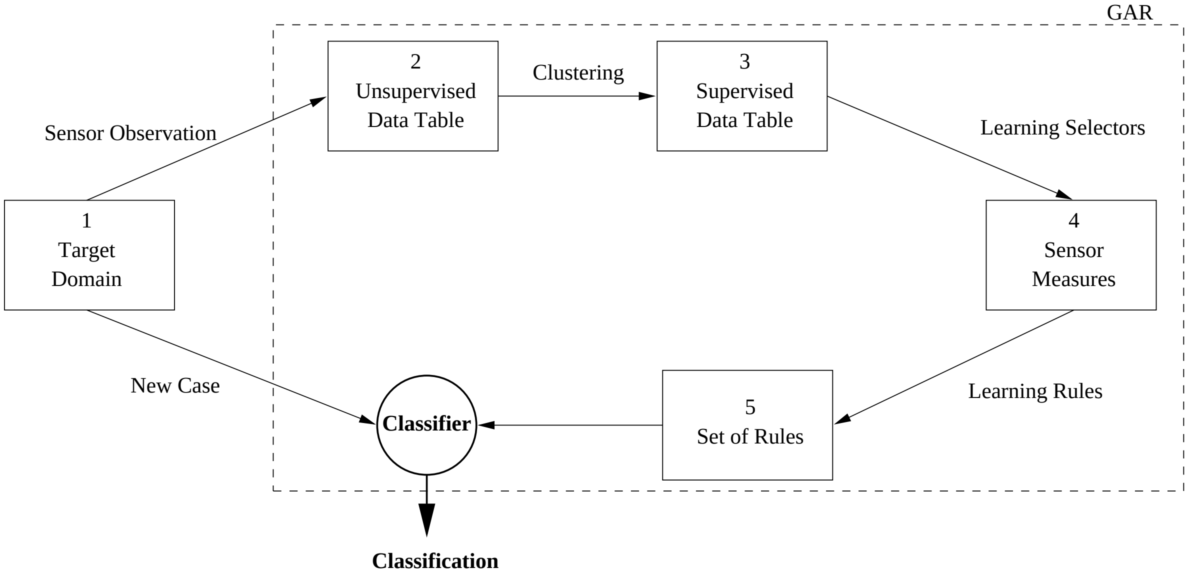

Figure 1: Diagram of the GAR process.

available, we can start with a supervised database which is the result of extending the unsupervised data table with a new column that indicates the state of the system when the data was captured. GAR detects and removes redundant information in the table. After that, it transforms the table in such a way that columns do not represent sensors but interesting sensor measures which are previously calculated by a *selector learning* process. In this paper we use two basic selector learning algorithms [Wong and Chiu, 1987], i.e. equal-width interval and equal-frequency interval, though GAR incorporates some others like pure-intervals, chi-square, or smooth. The algorithm *equal-width interval* partitions each sensor domain into k selectors of equal size. The algorithm *equal-frequency interval* generates k selectors that contain the same amount of data. After that, the dataset and the list of selectors are used to make the descriptive rules. The rule learning algorithm used here is based on the idea that useful rules represent at least one instance of the state that we want to represent. Rules are made by means of taking a state instance which is not yet represented. That instance defines a rule which describes it and which is the most general rule that does not describe instances of other state of the target domain. The instances that the rule represents are removed and the process is repeated until all the instances are represented by some rule.

GAR incorporates a classifier which is capable of taking the rules and deciding which are the states of the new situations according to the sensor measures. This property is used to determine the position in the state transition diagram that represents the control of the environmental system.

3 The wastewater treatment

3.1 Description

The foundation of the biologic Wastewater Treatment Plants (WWTP), as shown in figure 2 starts with a *pretreatment* stage in which some fine screens are used to filter the input water and separate it from sand, rocks, gravel and other floating elements. In the *primary settler* the water influent is calmed in order to permit the first sedimentation. After that, the biologic process starts and the organic material which is dissolved in the water demeans. This action is carried out in the *biological reactor* by some micro-organisms that use the water dirt as their food. In a later stage, these organisms that have been growing up must be separated from the water. This is carried out on the *secondary settler*, where the water remains quiet for some hours in order to let these micro-organisms coagulate into flocs. These flocs tend to pour off and fall to the bottom of the settler by their own weight. Flocs are structured on a kind of micro-organisms called filamentous bacteria. When there is an extensive amount of those bacteria, the floc has a spongy appearance and, therefore, it is capable of retaining water bubbles which avoid the micro-organisms to pour off. This phenomenon is called *bulking* and it is one of the most interesting abmormalities in the WWTP.

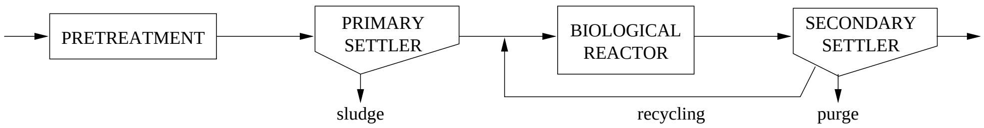

Figure 2: The Wastewater treatment process

We have been working with two data sets comming from the waste-water treatment plants of Manresa and Girona. Both plants perform the biological process known as *activated sludge process* in which the urban waters comming respectively from the towns Manresa and Girona (Catalonia, Spain) are treated with a WWTP. The data sets are described in table 1. `Man9091`[1] is a dataset containing the observation of 38 sensor measures in the plant of Manresa during 527 days at the rate of one observation per day in 1990/91. `Gir9596` is the dataset of the plant of Girona with the observation of 41 sensor measures during 343 days at the same rate in 1995/96.

Both datasets are described by a set of variables that contain information about the water along the WWTP process. So, the variables with names terminated in -E represent measures at the input of the WWTP, -P represent measures at the input of the primary settler, -D and -B represent measures at the input of and inside the bio-reactor respectively, and -S represent measures at the output of the WWTP. There are also, global variables (-G). `Man9091` does not have -B variables and `gir9596` does not have -P variables. Moreover, not all such variables contain sensor measures but performance values. Their names start with RD-. The measures refer to sensors that measure the water inflow (Q), pH (PH),

[1]This dataset is available at ftp://ftp.ics.uci.edu/pub/machine-learning-databases/water-treatment/.

`man9091`

town:	Manresa
inhabitants (1990):	100.000
No. days:	527
No. variables:	38
missing values:	2.9 %
observations:	1990/91
No. classes:	13

`gir9596`

town:	Girona
inhabitants (1995):	125.000
No. days:	343
No. variables:	41
missing values:	31.49 %
observations:	1995/96
No. classes:	6

Table 1: Wastewater domain descriptions.

biological oxygen demand (DBO), chemical oxygen demand (DQO), suspended solids (SS), volatile suspended solids (SSV), sediments (SED), and conductivity (COND). For example, DBO-S is the quantity of substrate at the output of the WWTP with values *normal* if it is below 35 mg/l, and *high* if it is above 35 mg/l; RD-SED-G is the percentage of global performance of the WWTP with respect to the settling efficiency; etc.

In `gir9596` there are some other important measures like the sludge volume index (IM), the mass load (CM1), the volume load (CM2), and the hydraulic residing time or time that the water is in the plant (TRH).

Both datasets have been processed with LINNEO+ [Sánchez et al., 1997], a conceptual clustering system that groups similar days into classes on the basis of a distance criterion and the size of the classes. In `man9091`, for a class size 4 LINNEO+ discovered thirteen plant situations that the domain experts identified as the states of the plant in table 2. These results were studied and the conclusions published in [Riaño and Cortés, 1997], but we introduce this dataset in this work for comparative purposes with `gir9596` and also to supply a historical perspective of our research on the wastewater treatment plant domain.

Nevertheless, the rest of this paper is centered on the generation of GAR descriptive rules for the WWTP states discovered by LINNEO+ in `gir9596`. Next, we recover the idea that the control of environmental systems can be automated with both expert systems and state transition diagrams. At the request of the domain experts, we have studied two particular cases of the dataset: the first case `gir9596a` studies how we can obtain a rule set describing the plant states when the WWTP is controlled by an Expert System. The dataset is restricted to only contain values about the following 20 variables: Q-E, QB-B, PH-E, SS-E, SSV-E, DQO-E, DBO-E, PH-D, SS-D, SSV-D, DQO-D, DBO-D, PH-S, SS-S, SSV-S, DQO-S, DBO-S, V30-B, MLSS-B, and MLVSS-B. The second case `gir9596b` studies the rules describing the states when the WWTP is controlled with a state-transition diagram as explained in the introduction. In this case, we only consider the following 15 variables: QA-G, PH-D, SS-D, SSV-D, DQO-D, DBO-D, PH-S, SS-S, SSV-S, DQO-S, DBO-S, V30-B, MLSS-B, MLVSS-B, and MCRT-B.

`man9091`

No.	State identification	No. days
1	Normal situation	275
2	Secondary-settler problems 1	1
3	Secondary-settler problems 2	1
4	Secondary-settler problems 3	4
5	Normal situation with performance over the mean	116
6	Solids overload 1	4
7	Secondary-settler problems 4	1
8	Storm 1	1
9	Normal situation with low influent	69
10	Storm 2	1
11	Normal situation	52
12	Storm 3	1
13	Solids overload 2	1

Table 2: States of the `Man9091` wastewater plant.

3.2 Learning and identifying the WWTP states

LINNEO+ is used to find the state classes of the plant for the datasets `gir9596a` and `gir9596b`. Domain experts refere to the state classes as operational situations. Broadly speaking, an *operational situation* is a working state of the plant described by means of the relevant variables of the system.

When the size of the classes is set to 3.5 the dataset `gir9596a` is classified in six different operational situations: NORMAL, STORM, BULKING, LOAD CRASH, CLOUDINESS, and LOW LOAD. When the size of the classes is set to 2.5 `gir9596b` describes six operational situations of the plant: NORMAL, PRIMARY LOW PERFORMANCE, LOW SETTLING, STORM, LOW PERFORMANCE, and BAD INFLUENT.

Before rules are made, the missing values in the supervised datasets are filled with the mean values of the respective variables in order to have a complete dataset which is used to discover the interesting conditions on the variables. These conditions are called *selectors* and they have the form of a range of values. For example, `SSV-S=[34.899998,68.199997]` indicates that there are many suspended solids at the output of the plant, `V30-B=[250.25, 423.50]` indicates a wrong settling of micro-organisms within the bio-reactor, etc.

We have generated rules for the datasets `gir9596a` and `gir9596b` using the learning selector algorithms *equal-width interval* and *equal-frequency interval* and fixing the number of selectors. Few selectors indicate that, on the one hand they must be more general because if they must be mutually exclusive, the ranges that they define are wider and, on the other hand there is a tendency towards producing less and larger rules describing the same operational situations because general selectors are less precise than specific selectors. In other words, when we force using general selectors, rules use to contain many selectors in the premise and they use to describe many cases. So, we need less rules in order to represent the WWTP operational situations. As far as the experts opinion is concerned, large rules with general selectors are more concrete than small rules with specific selectors and this makes the former to take more risks than the latter. Concrete rules produce two effects on the domain experts: they agree very much with the meaning of the rules, or they do not understand the rules.

In table 3 we show the rules that GAR obtains for the BULKING state of the WWTP with the dataset `gir9596b`. BULKING is detected with the V30 test which indicates the micro-organism settling capacity. High V30 values means a wrong settling. BULKING can also be detected from its consequences which are a great amount of micro-organisms at the outflow (i.e. high SSV-S and SS-S values), and probably a great amount of organic material at the outflow (i.e. high DQO-S and DBO-S values).

```
R31:  DBO-S = [63.50,84.00] => BULKING
R32:  SSV-S = [34.89,68.19] & V30-B = [250.25,423.50] => BULKING
```

```
R27:  DBO-S = [56.67,84.00] => BULKING
R28:  SSV-S = [45.99,90.39] & DBO-S = [2.00,29.34] &
      V30-B = [308.00,539.00] => BULKING
R29:  SSV-S = [45.99,90.39] & DQO-S = [111.66,163.00] => BULKING
```

```
R43:  DBO-E = [216.83,987.00] & SSV-D = [68.00,134.00] &
      DBO-D = [124.00,274.0] & PH-S = [7.6,8.0] &
      SSV-S = [12.65,134.80] & V30-B = [270.00 770.00] => BULKING
R44:  DBO-D = [124.00,274.00] & PH-S = 7.50 & SSV-S =
      [12.65,134.80] &
      DQO-S = [55.00,163.00] & V30-B = [270.00,770.00] => BULKING
R45:  SSV-D = [62.00,68.00] & PH-S = [7.60,8.00] & DQO-S =
      [55.00,163.00] &
      V30-B = [270.00,770.00] => BULKING
R46:  DBO-D = [116.00,124.00] & PH-S = 7.60 & V30-B =
      [270.00,770.00]
      => BULKING
```

Table 3: GAR rules describing the BULKING state

The first group of rules use the selectors which we obtain when we force the *equal-width interval* selector learning algorithm to learn four selectors for each variable. Rule `R31` says that BULKING occurs when there is a high biologic oxygen demand at the output of the plant and rule `R32` says that BULKING is also detected when there is a high V30 value combined with many volatile suspended solids at the output. Both rules not only are according to the above theoretical explanation of BULKING, but also describe all the cases of BULKING.

The second group of rules is made with the selectors proposed by the *equal-width interval* selector learning algorithm when it is asked to give three selectors for each variable. Rule `R27` is very similar to `R31`. Rule `R28` is similar to `R32`, but it replaces precision in the SSV-D range of values with considering the new variable DBO-S. Observe that DBO-S acts as an excluding factor which separates the cases that `R27` represents, from those cases that `R28` represents. Finally, rule `R29` describes other BULKING situations in terms of the SSV-S and the DQO-S variables. Experts agree that this last is the best rule about BULKING.

In the last group of rules, we used the selectors obtained with the *equal-frequency interval* selector learning algorithm and proposing three selectors for each variable. At first sight, experts were surprised of the possible dependency of BULKING on variables at the input of the plant as DBO-E, variables before the bio-reactor as SSV-D and DBO-D, or the variable PH-S. They showed their interest in studing these dependencies in detail, but they soon realized that the variable ranges were as wide as to mean anything. We interpret that the ranges of the important BULKING variables are not well established by the selector learning algorithm and the rest of variables must act as reinforcing variables.

For the sake of completeness, in table 4 we also supply the rules obtained for the rest of states of the `gir9596b` dataset, though we will not explain the results.

4 Conclusions

We have applied the rule inductive learning system GAR to the generation of rules within the wastewater treatment domain. In spite of that, this work is presented as a first approach to the problem, we aim to construct an automatic system that, from the observation of the sensors in a WWTP it is capable of discovering the internal states of the plant or operational situations, classifying the sensor inputs into prototypes, and integrate states, prototypes and actions on the plant in order to generate a state-transition diagram that could be used to control the WWTP.

Here we explain the first step in this process. We use rules to represent the states of the plant that the conceptual clustering system LINNEO+ obtains. We want these rules not only to be useful, but also to be undestandable by domain experts. We think that the next step of the work is to operate exclusively on -E variables, so we can discover input prototypes with LINNEO+ and represent them with GAR rules.

5 Acknowledgements

On the personal, I want to thank the help and comments of Prof. U. Cortés. The `gir9596` dataset was supplied by J. M. Gimeno. Prof. M. Poch, J. Comas, and very specially Dr. I. R. Roda were the WWTP experts that interpreted the rules that GAR obtained. On the institutional, this work was founded with the CICYT project No. TIC96-0878.

```
R27:  DQO-D=[27.00,154.75] & DBO-D=[95.50,155.00] &
      PH-S=[7.00,7.25]
      => PRIM-LOW-PERF
R28:  DQO-D=[27.00,154.75] & DQO-S=[47.50,86.00] &
      V30-B=[250.25,423.50]
      => PRIM-LOW-PERF
R29:  DQO-D=[27.00,154.75] & DBO-D=[95.50,155.00] &
      PH-S=[7.25,7.50] &
      V30-B=[250.25,423.50] & => PRIM-LOW-PERF
R30:  QB-B=[19883.00,27973.40] => PRIM-LOW-PERF

R31:  DBO-S=[63.50,84.00] => BULKING
R32:  SSV-S=[34.89,68.19] & V30-B=[250.25,423.50] => BULKING

R33:  DBO-E=[528.00,757.50] & PH-S=[7.50,7.75] => STORM

R34:  DBO-D=[95.50,155.00] & PH-S=[7.75,8.00] &
      DQO-S=[86.00,124.50] &
      DBO-S=[2.00,22.50] & V30-B=[250.25,423.50] => LOW-PERF
R35:  PH-S=[7.250000,7.50] & SSV-S=[34.89,68.19] &
      DBO-S=[2.00,22.50] => LOW-PERF
R36:  DBO-D=[155.00,214.50] & PH-S=[7.75,8.00] &
      DQO-S=[47.50,86.00] &
      V30-B=[250.25,423.50] => LOW-PERF
R37:  SSV-S=[34.89,68.19] & DQO-S=[9.00,47.50] &
      DBO-S=[22.50,43.00]
      => LOW-PERF

R38:  SSV-D=[13.00,43.25] & PH-S=[7.00,7.25] &
      V30-B=[77.00,250.25]
      => BAD-INFLOW
```

Table 4: Rule base describing the states of the WWTP

References

[Clark and Niblett, 1989] Clark, P. and Niblett, T. (1989). The CN2 inductive algorithm. *Machine Learning*, 3:261–271.

[Dym and Levitt, 1991] Dym, C. L. and Levitt, R. E. (1991). *Knowldge-based Sytems in Engineering*. McGraw-Hill.

[Gimeno et al., 1998] Gimeno, J. M., Béjar, J., Sànchez, M., Cortés, U., and Roda, I. (1998). Discovering and modeling process change: An application to industrial processes. In *Proceedings of the 2nd International conference on Practical Applications of Knowledge Discovery and Data Mining*.

[Hudson et al., 1997] Hudson, A. D., Sanders, D. A., Golding, H., Tewkesbury, G. E., and Cawte, H. (1997). Aspects of an expert design system for the wastewater treatment industry. *Journal of Systems Architecture*, 43:59–65.

[Quinlan, 1993] Quinlan, J. R. (1993). *C4.5: programs for machine learning*. Morgan Kaufmann. ISBN 1-55860-238-0.

[Riaño, 1997] Riaño, D. (1997). *Automatic construction of descriptive rules*. PhD thesis, Universitat Politècnica de Catalunya. http://www.etse.urv.es/~drianyo/publication/PhD.ps.Z.

[Riaño and Cortés, 1997] Riaño, D. and Cortés, U. (1997). Rule generation and compactation in the WWTP. *Computación y Sistemas*, 1(2):77–90.

[Sánchez et al., 1997] Sánchez, M., Cortés, U., Béjar, J., de Gracia, J., Lafuente, J., and Poch, M. (1997). Concept formation in WWTP by means of classification techniques: a compared study. *Applied Intelligence*, 7(2):147–166.

[Wong and Chiu, 1987] Wong, A. K. C. and Chiu, D. K. Y. (1987). Synthesizing statistical knowledge from incomplete mixed-mode data. *IEEE Transactions on Pattern Analysis and Machine Learning*, 9(6):796–805.