Knowledge discovery for control purposes in food industry databases

S. Guillaume, B. Charnomordic

▶ To cite this version:

Knowledge discovery for control purposes in food industry databases

Serge Guillaume[†] and Brigitte Charnomordic[‡]
Fuzzy sets and Systems, 122(3):487-497, 2001

[†]*Cemagref Laboratoire GIQUAL, 361 rue Jean-Franois Breton, 34033 Montpellier, France*
[‡]*INRA Laboratoire de Biomtrie, 2 Place Viala, 34060 Montpellier, France*
E-mail: serge.guillaume@cemagref.fr, bch@ensam.inra.fr

Abstract :

Sets of experimental data describing a product at various processing steps are widely available in Food Industry. Decisions taken by the human operator all through the process are implicitly contained in such a database, as well as the recorded consequences on the product.

The aim of this work is knowledge discovery. This knowledge must be uttered in a way that allows cooperation with the expert's knowledge.

The system is implemented as a self learning fuzzy controller, with the rule conclusions being optimized by a genetic algorithm. The role of the fuzzy controller architecture is to provide a learning framework, the database being used for rule validation, thus acquiring hidden knowledge.

In order to make inferred knowledge easy to understand, a rule and variable selection methodology has been developed.

Data from a cheesemaking process were used to test our approach.

KEY WORDS : Fuzzy Logic, Genetic Algorithm, Machine Learning, Knowledge Discovery, Process Control, Food Industry, Cheesemaking.

1 Introduction

Food Industry involves many strongly non linear multivariable complex processes.

Control of processes by classical methods is based on model use. But, for complex and imperfectly understood processes, phenomena are not sufficiently well known to be mathematically characterized, hence the importance of empirical knowledge of a human operator. To act, the operator relies above all on his know-how, in conjunction with the interpretation of physical measurements given by sensors and partly through subjective appraisal of sensory criteria.

The class of problems we are interested in involves a great number of variables. Expert knowledge does exist, but it does not cover all situations. Let us take as an example the case of a traditional cheesemaking process, such as the making of Comt cheese. The process can be viewed as a series of sequential subprocesses, complex and multivariable themselves. To reach a high quality product, the human operator must be fully aware of the characteristics of subprocesses and he must integrate them into the whole process.

He acquires deep knowledge on the influence of a particular control variable over the final product, as well as about the impact of some state variables, at a given step. What remains very difficult for him to appreciate concerns the interactions between variables which belong to different subprocesses. Nevertheless, these interaction factors are presumably important.

In an adaptive tracking perspective, for target following through and to correct drifting off, this process, as many similar food industry processes, needs a decision support tool.

Over the past several years, fuzzy control based on expert knowledge has proven a successful method of designing controllers (Mamdani, 1994),(Perrot et al., 1995). A similar approach was tried

out for Comt (Charnomordic et al., 1998). Expert knowledge introduced in the fuzzy inference system was not sufficient to give a satisfactory predictive quality, the lack of knowledge about variable interaction being strongly suspected for the poor predictive performance.

Even for non automated processes, process data are usually available. For quality control purposes field data are collected on a regular basis. Process databases constitute a potential huge source of information still to be made the most of.

Process data describe the product being processed. They contain implicit control decisions and their consequences on the product. In other words they hold expert knowledge. Knowledge discovery based on process data can help to build a realistic decision support tool. Extracted knowledge must be interpretable by experts to allow a better understanding of the process, and a profitable cooperation between human knowledge and computer learning.

Quality criteria for a knowledge discovery model include and go beyond quality criteria for a blind learning model. Common learning criteria must be respected including quality of fit for the learning database and use of validation data for robustness. Complementary criteria concern discovered knowledge. The validity range, the novelty as compared to what is already known by experts, the interpretability level are all important and not easily quantified.

Our method will use a fuzzy inference system framework and data from a database.

Training procedures have been applied to fuzzy inference systems for tuning fuzzy set parameters or learning the rule base. Training techniques were often inspired by optimization methods used in other fields. Neural networks learning procedures combined with fuzzy control led to a new family of fuzzy controllers, called neuro-fuzzy networks (Jang, 1994). Stochastic algorithms, for instance genetic algorithms, were used for automatically generating a rule base from training data. Although these approaches seem to be interesting, they have been applied to academic problems only, such as pole-cart balancing (Chiang et al., 1997), (Leitch & Probert, 1998). These involve a few variables. Data bases from food processes are likely to be different. Numerous variables are collected, data are not especially designed for our use : data are noisy, some values are missing, the experiments do not cover the input space.

When there are more than a few input variables, the number of generated rules dramatically increases with the number of variables. This is known as "the combinatorial curse of dimensionality", and it is a serious restriction to the applicability of automatic generation techniques, for real world problems, such as the ones we are interested in.

The objective of this paper is to generate a set of interpretable fuzzy rules from a training data base. Training can be seen as a validation procedure based on available data: potential rules contain knowledge which has to be confirmed or infirmed by data. Each rule of the final fuzzy rule set describes a trajectory in the variable space. A procedure will be discussed for rule and variable reduction, to avoid a too high dimensionality and to insure consistency. This procedure is not yet completely automatic: experts are asked to confirm variable removal. Practical aspects will be discussed in a case study, using data from a cheesemaking process.

2   System Description

2.1   Fuzzy Logic for learning: why and how

The problem of control can be viewed as one of decision making, i.e. given the observation of the process state to decide which action to take. Decision making can be achieved by a fuzzy controller. A fuzzy controller is a rule based system whose inference engine benefits from the fuzzy logic capability to handle human reasoning.

A typical fuzzy control loop is shown on figure 1. The loop feeds back the process state into the fuzzy controller which in turn provides a control action.
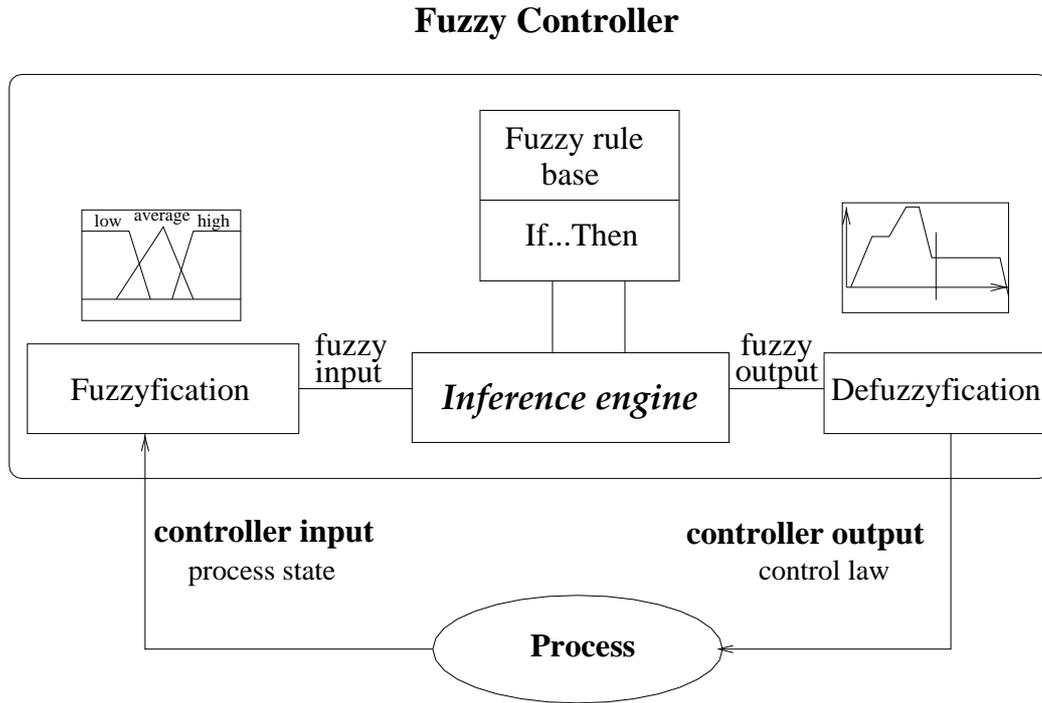
Fuzzy Controller

Figure 1: A fuzzy controller

The fuzzy controller architecture includes three principal components: a fuzzification procedure, an inference engine and a defuzzification step. The fuzzification procedure converts the values of the fuzzy controller input variables into linguistic values, corresponding to fuzzy sets. The fuzzy inference engine relies on a set of fuzzy rules to infer a fuzzy output. A fuzzy rule is divided into two parts : the antecedent part and the consequent part. Suppose that we have the two following rules :

$$\text{rule1 : if } X \text{ is } A_1 \text{ and } Y \text{ is } B_1 \text{ then } Z \text{ is } C_1$$
$$\text{rule2 : if } X \text{ is } A_2 \text{ and } Y \text{ is } B_2 \text{ then } Z \text{ is } C_2$$

At time t, the process state is given by the (x_t, y_t) vector, represented by linguistic variables X and Y. Given the membership function for each linguistic label (A_1, A_2, B_1, B_2), membership degrees $\mu_{A_i}(x_t)$ and $\mu_{B_i}(y_t)$ can be computed for each rule. Combination of conditions in the antecedent part is done using a fuzzy logic AND operator, denoted as $\wedge$, most often chosen as a *product* or *minimum*. The matching strength of $Rule_i$, also called rule weight, is called w_i and defined as follows:

$$w_i = \mu_{A_i}(x_t) \wedge \mu_{B_i}(y_t) \tag{1}$$

The consequent part z_i of $Rule_i$ is calculated by applying a fuzzy implication mechanism. The fuzzy output is obtained by aggregating $Rule_1$ and $Rule_2$ consequent parts. The simplest defuzzification procedure is done by computing a weighted average. If we generalize to a set of n rules, the fuzzy controller output is

$$z = \frac{\sum_i w_i . z_i}{\sum_i w_i} \tag{2}$$

3

To be more useful in modeling process drifts, the output can be expressed in terms of a fixed target value moderated by a fuzzy inferred output, which comes to

$$O = T + \frac{\sum_i w_i.z_i}{\sum_i w_i} \qquad (3)$$

We will use the fuzzy controller as a knowledge learning framework. Available data are considered as an implicit recording of human decisions and process state all through the process. A rule base will be built that best reflects the data used for learning and the hidden knowledge they contain. The generated rules are checked for consistency and interpretability. A rule consistency criterion will be applied to avoid internal contradictions. For interpretability sake, the rule base must be kept to a reasonable size. Variable and rule reduction will be done to point out major trends and to make the rule base easier to be interpreted by experts.

It should be emphasized that the performance of the fuzzy rule base is highly dependent on the database quality. Ideally the database should represent an extensive coverage of all kinds of various situations.

2.2 Practical Implementation

The fuzzy controller is used as a fuzzy inference system: instead of a control action, its output is a context dependent target value.

Its input variables are state or action variables. For each variable, fuzzy sets need to be defined. Two possibilities arise for defining the fuzzy sets. Partitioning can be self-generated by observing the distribution of values in the database, optimized by using stochastic techniques such as genetic algorithms, or it can be established by taking advice from experts. We decided on the last method for a better readability, linguistic labels chosen by experts being more meaningful.

Once the partitioning decided upon, a first set of rules is generated by considering all possible combinations of input fuzzy variables.

Learning consists of running a genetic algorithm to optimize the consequent parts of the fuzzy rules in order to design the rule base best suited to the database. The best fit is considered in terms of performance as well as rule consistency and interpretability. Genetic algorithms offer two main advantages in our case. First a global optimization technique is needed because of the complete lack of knowledge about each rule consequent part. Moreover the same computational frame could be reused for automatically partitioning the input space of some variables.

The method used must be applicable to real world databases, which are often plagued by a non neglectable number of missing values. A strategy for dealing with missing values without loosing too much information is needed. It is essential not to increase dimensionality, thus we do not define a specific linguistic label for missing values. Missing values are mostly due to manual information collecting rather than sensor faults in a specific range. In order not to eliminate possibly important combinations in rule definition, missing values are considered as equally distributed over the existing fuzzy sets. Indeed each of the linguistic labels is meaningful, because of the partitioning being done by experts. This assumption would be questionable in the case of automatic partitioning.

The set of rules will be built accordingly. During the optimization step, missing values will be replaced by randomly picked values in the corresponding variable range, and then fuzzified. Of course, the accuracy of the results is dependent on the ratio of missing values. A reasonable limit seems to keep this ratio less than 20%.
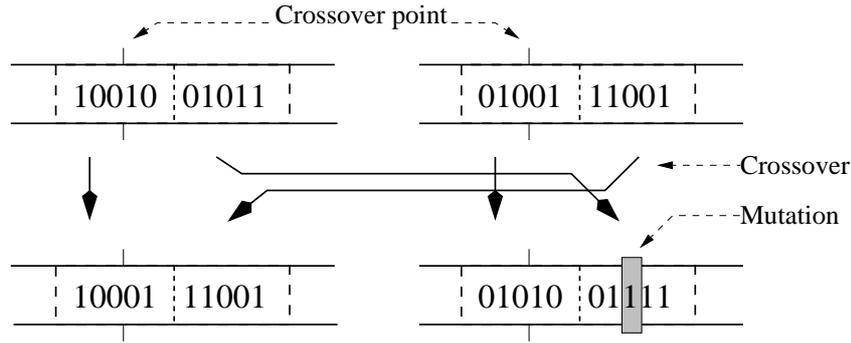
4

2.3 Genetic Algorithms

Since they have been proposed by Goldberg (Goldberg, 1989), Genetic Algorithms (GA) have been widely used in optimization and machine learning. GA are part of stochastic techniques. These are particularly useful when the number of possibly solutions of a problem is so large that it is impossible to test each of them. Their objective is twofold : explore the solution space using a random way to find better and better solutions and, optimize these by finding the local optima, that means the best solutions in their neighborhood.

The solution found is not guaranteed to be the best one, but the aim of the iterative process is to reduce the set of solutions to the global optima.

GA use biologically inspired concepts.

A set of solutions, called a population, is alive all along the process. Each solution, called a chromosome, is made of a set of genes which encode the problem.

Rule 2   Rule 5

| 0 | 1 | 1 | 0 | 1 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 1 | 0 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | 1 |

Rule 1   Rule 4

Figure 2: Chromosome encoding

In our `n-rule` problem, a chromosome is made of n substrings, each substring representing the encoded consequent part of a rule. All substrings have the same length. Figure 2 illustrates a `5-rule, 5-bit substring` chromosome.

Each step of the iterative process is called a generation.

At the end of a step, each chromosome is evaluated through an objective function. Those which have the best performances according to this function are allowed to breed. This is done by two kinds of operators : crossover and mutation.

The sexual crossover operator produces two children from two parents chosen among the best candidates, a mutation operator is applied to bring disorder.

In the framework of our application, GA are used to optimize the consequent part of the rules, their antecedent being fixed. The chromosome is made of a set of encoded numerical values, each one being associated to a rule. A value is encoded as a binary string. The chromosome may thus be considered as a sequence of substrings.

The crossover operator we use is called 'OnePointCrossover' : the crossover point is randomly chosen within the chromosome length. The chromosome of the first child is made by combining the part of the chromosome of the first parent located before the crossover point and the part of the chromosome of the second parent located after this point. The chromosome of the second child is its counter part.

This operator seems to be suitable for substring handling : at most one substring is modified.

The mutation is done by inverting some genes, randomly chosen according to a given probability.

Figure 3 shows the effect of these operators on the substrings located near the crossover point for a 5-bit encoding. The bit inside the grey box has been changed by the mutation operator.

Each rule consequent part is allowed to vary within two limits `Low` and `High`. If `l` is the length of the binary string, the consequent part associated to a rule is :

5

Figure 3: Genetic operators

$$Consequent\ part = Low + (High - Low) * \frac{Binary\ value}{2^l - 1} \tag{4}$$

Learning is based on an objective function. The data base is a set of multidimensional data points, including input and output variables. The objective function will assess the chromosome performance by comparing the observed output with the fuzzy system output. The fuzzy system output is given by equation (2), where z_i is the consequent part of rule i. The best chromosomes are those which minimize the square root of the squared sum of errors, also called RSSE, given in the following equation:

$$RSSE = \sqrt{\sum_n \left(Predicted\ output - Observed\ output\right)^2} \tag{5}$$

where n is a given number of data points either randomly chosen in the base or taken in a sequential order.

The complete architecture of the controller is given in figure 4.

3   Space reduction to make inferred knowledge easy to understand

An exhaustive system would be quite complex. Such a system presents some drawbacks :

- The size of the system : the larger it is, the more difficult the finding of an optimal solution is. Let us consider a process involving 12 input variables, each of them described using 3 membership functions. Then the number of rules in the system is 3^{12}, more than 500000. Let us suppose than the consequent part of a rule may take 32 values, then the number of combinations is 32^{500000}.

- The robustness : a complete system is likely to be over dimensioned. Extra variables bring more noise than information. Training in these conditions is less robust, as it is likely to learn noise instead of the underlying process represented by the training set.

- The knowledge : instead of getting the best fit we are rather interested in finding out major trends. Our aim is knowledge discovery and not the design of an accurate predictive system. This knowledge may be used, in cooperation with the experts one, to control the process.

The proposed methodology aims to reduce dimension in both spaces : the variable space (number of inputs) and the rule space.

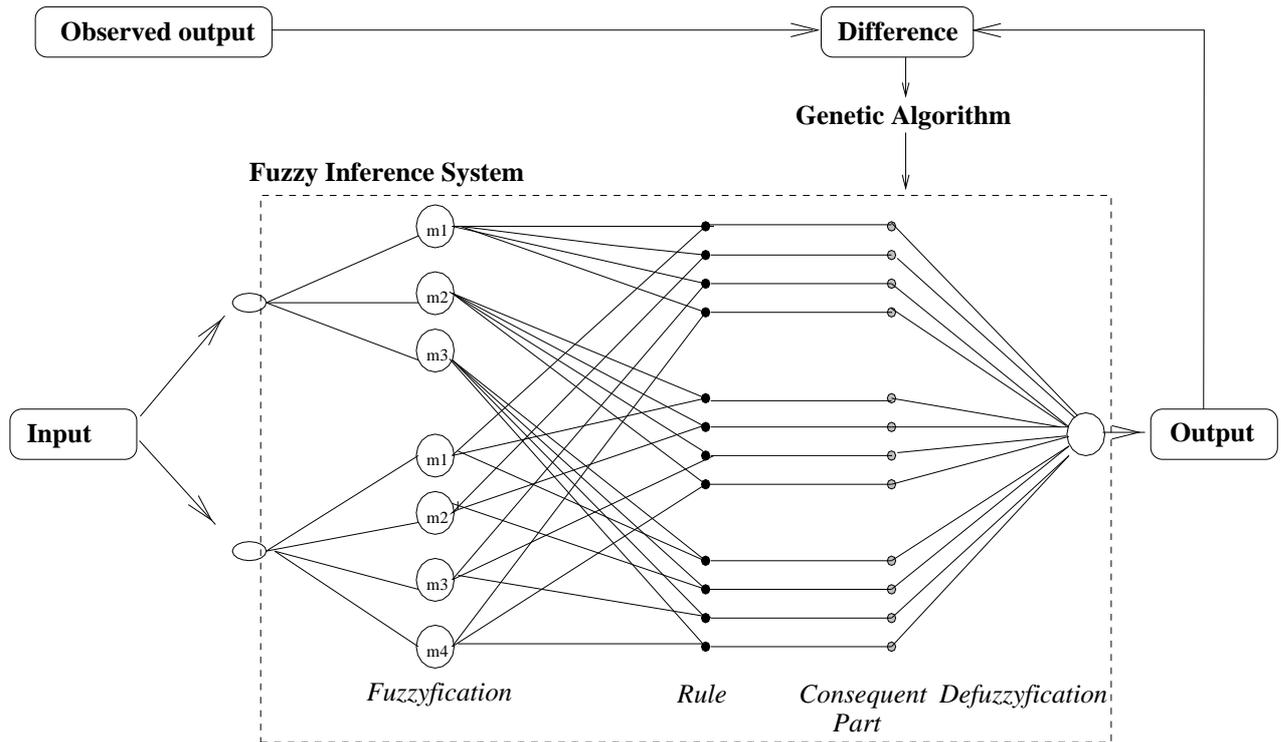The selection is done through 2 iterative procedures.

6

Figure 4: The fuzzy controller

3.1 Rule selection

Considering a given set of input variables and the corresponding partitioning, all the possible rules are implemented.

Let us give some basic definitions :

- activity : a data point activates a rule when the corresponding rule weight is non zero.

- active data point : a data point is active when it activates at least one rule of the rule base.

- active rule : a rule is active when it is activated by at least one data point.

Obviously, the only rules which may be validated by the procedure are the active ones.
The rule selection is done according to 3 criteria :

1. Influence :
   The first criterion is the influence of the rule in the data base. For a given data point the rule weight is the product of all input membership degrees, as shown in Equation 1. For a given rule, the influence is measured by its cumulated weight in the data base. Only the most influent rules are kept. Let us note that this selection step does not involve GA.

2. Consequent part level :
   GA associate a consequent part C_i to each rule R_i in the system. Remember that C_i represents a gap to be added to a base target value (as in equation 3). GA being random based techniques, they are non deterministic : successive runs may lead to different results. We choose to repeat r times the GA with the same configuration to improve reliability. At the end of the repetitions,

we have, for each rule R_i, r values for C_i : $C_{i_1}, C_{i_2}, \ldots, C_{i_r}$. The final value is taken as the median of the $C_{i_j}, j = 1 \ldots r$ values. Thus a value about zero might result from a combination of negative and positive values, and cannot be trusted.

We can distinguish between two kinds of rules :

- neutral rules : a rule R_i is considered as neutral when $median(C_{i_j}) < C_{th}, j = 1 \ldots r$, where C_{th} is a given threshold. Neutral rules contain standard knowledge.

- sensitive effect rules are rules whose consequent part final value is above a given threshold. They interest us most as they are likely to explain deviations from the standard product being processed.

In our approach, we choose to remove neutral rules.

3. Consistency :
The last criterion takes into account the rule consistency. It is expected for rules activated by the same group of data points to have a similar behavior. The link between two rules `i` and `j` is defined as follows :

$$Link_{i,j} = \frac{N_{i,j}}{max(N_i, \ N_j)}$$

where N_i, N_j, are cardinals of the sets $E_i, \ E_j$ of data points which activate respectively rules i and j; $N_{i,j}$ is the cardinal of $E_i \cap E_j$.

When two rules are strongly linked, $Link_{i,j}$ greater than a given threshold, their consequent parts either have the same sign or are inconsistent. Inconsistent rules are eliminated before next iteration.

It is important to note that as the number of rules is reduced, the number of active data points decreases.

3.2   Variable selection

Once the rule selection is done, the selected rules can be analyzed in order to assess the relative importance of the input variables.

The variable selection step is necessary. Databases are constituted of field data, and recorded variables do not necessary present a real interest for control purposes. The goal of this procedure is to keep, as a data point descriptor, the variables which are useful for control, and is at the heart of knowledge discovery.

The variable selection is based on the following axiom, which requires a rule distance function:

`IF`	*two rules are close*	
`AND IF`	*they are weakly linked*	
`AND IF`	*their consequent parts have opposite signs*	
`THEN`	*the inputs which differ are of major importance for the output, within the context described by the other inputs.*	

8

Similarly, if the rules are close and weakly linked, and if their consequent parts have the same sign, then the inputs which differ do not have a noticeable effect on the output, according to the whole context.

The distance between two rules can be defined as the number of variables whose linguistic label values differ in the rule descriptor. A rule descriptor is defined as a sequence of integers, and only describes the antecedent part of the rule. Each integer corresponds to the linguistic label value for the corresponding input variable. Let us consider the following example, where we have four input variables.

$$Rule\ 1: \text{if } Var_1 \text{ is } A_3 \text{ and } Var_2 \text{ is } B_2 \text{ and } Var_3 \text{ is } C_2 \text{ and } Var_4 \text{ is } D_3$$
$$Rule\ 2: \text{if } Var_1 \text{ is } A_1 \text{ and } Var_2 \text{ is } B_4 \text{ and } Var_3 \text{ is } C_2 \text{ and } Var_4 \text{ is } D_2$$

Rule 1 and *Rule 2* have the following descriptors: 3223 and 1422. They differ in three digits: first, second and fourth digits. The distance between rule 1 and rule 2 is considered as equal to three.

Generalizing to a n-input system, the distance between any two rules is given by

$$\sum_{i=1}^{n} (1)\{m_i \neq l_i\} \tag{6}$$

where m_i and l_i are respective linguistic label values taken by *variable i* in *Rule 1* and *Rule 2*, and n is the number of input variables.
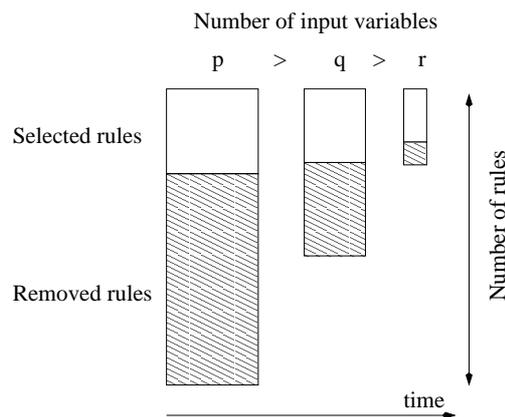

Figure 5: Selection of rules and variables

Variables should be removed which have not a noticeable effect on the output, whatever the context. The variable removal is currently achieved with the help of experts to prevent errors in the elimination of variables. Indeed, some rules in the system may be activated by a very small numbers of data points. Upon removal, a new reduced more significant system can be built. The complete reduction procedure for rules and variables, shown in figure 5, is reiterated.

4   Application to a cooked and pressed cheesemaking process (Comt)

Cheesemaking for cooked and pressed type cheese such as Comt runs over at least 4 months until 6 to 12 months including ripening. The first process stage that produces young cheese takes about 20 hours. This stage is divided into several steps: milk maturation, coagulation, draining in vat-cooking-stirring, and pressing, as shown in figure 6.
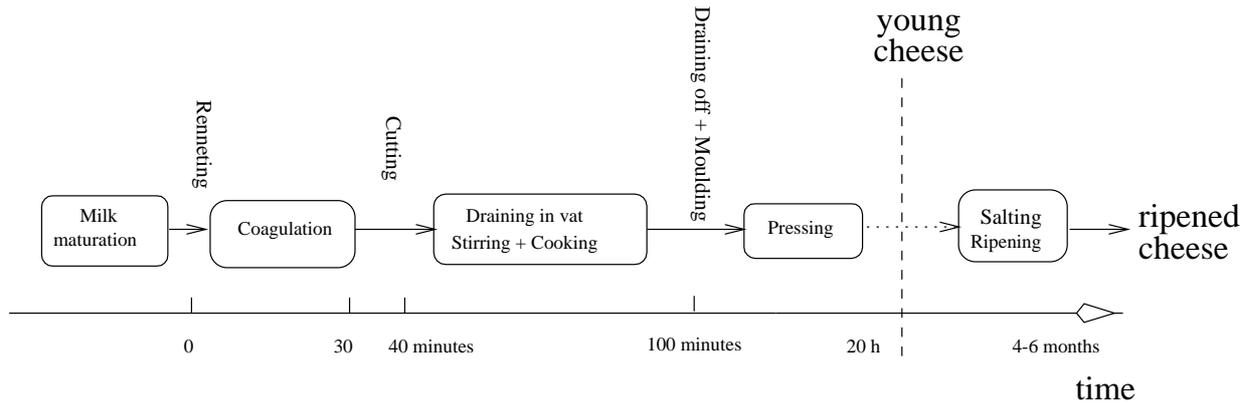
Figure 6: Summary of Cheesemaking

Each step is characterized by a set of variables, described in table 1. All variables correspond to numerical measurements, except those marked with a bullet (•) in the second column. These variables are routinely measured and available in technical report sheets. At the end of pressing, some analytical tests may be applied to selected young cheeses, in order to assess properties of a cheese set. The most common tests are physico-chemical analyses (*pH, moisture content, fat content and lactate ratio* being the most common ones). The present paper is restricted to generating the rules which mark the influence of process variables onto moisture content (in young cheese). Moisture content, abbreviated as MC, is the fuzzy system output variable.

Input variable/*Step*	qualitative variable	abbreviated as
Maturation of milk		
Milk (in vat) type	•	Milk
Starters combination	•	Starter
Coagulation		
Renneting temperature		RT
Renneting pH		RpH
Cutting time to clotting time ratio		CCR
Curd grain size	•	GS
Draining/Cooking/Stirring		
Maximum temperature		MT
Cooking Duration		CD
Stirring Duration		SD
Draining Temperature		DT
Pressing		
Dornic acidity (of serum)		DA
Output variable		
Moisture Content		MC

Table 1: Available variables in database

Partitioning the input space was done with the help of experts. The database included 172 data

points, with a number of missing values distributed among all variables. The method described in section 2 was used to generate a first rule base, including all possible 11 input variables. The variable selection procedure was applied to reduce the number of relevant variables and to build a second rule base through reiteration of the rule selection procedure. The third rule base was built again in a similar way. Table 2 gives a summary of the results. For each rule base, the number of variables, the percentage of missing values are given in the first two lines. The total number of possible rules corresponding to the input space partitioning is indicated. Data points available in the database only activate some of these rules, called active rules. The fuzzy system output is computed from a base value, chosen as 55, which is modified by all active rules (see equation 4). Each rule is allowed to influence the base value within the range [-5,+5].

Rule Base	1	2	3
Variables	11	7	4
Percentage of missing values	10%	6%	1%
Possible rules	500 000	2 200	54
Active rules	38 000	500	33
Selected rules	30	33	19
Active data points	70	73	79
Maximum error	3.2	2.5	3.0
RSSE	8.5	7.6	9.4
SEC	1.0	0.91	1.1
Variables to be removed before next step	RT,RpH,GS,DT	Milk,CCR,DA	
Variables found of interest	MT	Starter	CD,SD

Table 2: Summary of the results for the three rule bases

Let us give some comments about the first rule base. Two indices give a measurement of the system performance: the total RSSE, defined as in equation 5, and the standard error of calibration SEC, given by

$$SEC = \frac{RSSE}{\sqrt{(n)}}$$

where n is the number of active data points.

Considering the high dimension of the initial rule base (500 000 possible rules), the performance of the method appears quite satisfactory. In the fuzzy system, each rule is allowed a given influence on the output ([-5,+5]). Thus a n-rule system output might range between [-5*n, +5*n] about the base output, which, in our case, is far from reality: moisture content is a percentage and only varies from 0 to 100. No validity control is done, convergence is achieved by the optimization algorithm. Even so, the maximum error (3.2) and the total RSSE (8.5) resulting from the generated rule set are very reasonable. The relative error given by SEC is 1.0, which is close to the known accuracy of moisture content measurements (0.6). The variables designated for removal at the end of first step are given in column 1 of table 2. In concert with experts, that choice was confirmed. The first three variables to be removed have in common to be poorly measured for different reasons. RpH suffers from a high inaccuracy in measurements due to the small observed variability. RT is not of much use in practice and has many missing values. The grain size GS could certainly be very informative, but is currently attributed on subjective criteria with no established common reference.

11

The second system built shows the best performance of the three: all error markers decrease. The third system is characterized by a small set of four variables. Variable selection procedure shows that, although available variables all contribute to the process state, moisture content is essentially determined by the cooking stage.

Even though error markers increase in value, they remain reasonable. This last system was useful to bring out some new elements of knowledge about the process.

Discussion

At each step, the procedure highlighted some variables, as being more of interest for process control. The most valuable elements of knowledge discovered were:

- *statement 1*: Use $High$ maximum cooking temperature (MT) to $raise$ moisture content.
- *statement 2*: Use $LowActivity$ starters to $raise$ moisture content, if MT is not $High$.
- *statement 3*: Stirring Duration (SD) and Cooking Duration (CD) are $complementary$.

From the first system a well known fact in process characteristics came out. The maximum cooking temperature MT is a dominant feature, and tends to shadow other processing factors. Our variable selection procedure also identified three variables (RT, RpH and DT) as of little importance. After discussion experts confirmed the dominant influence of MT, as well as the secondary role of the three variables listed above for the output being studied (Moisture Content). They also suggested GS (curd grain size) removal, due to a lack of established reference values. More was expected concerning starters influence. Starters influence may be hidden by MT effects, but should come out somehow. Experts then realized that the initial fuzzy partitioning of starters space was not well adapted and could be improved. A new partitioning was suggested.

After removal of four variables, the phenomenon described in the second statement came out clearly. Rule Base 2 shows that starters behavior is context dependent. It may influence moisture content if and only if Maximum Temperature is not High. A high temperature slows down starters activity. The starters variable results from a multidimensional analysis of various characteristics. The new partitioning was done by giving more weight to some of them. Results show that they are more important for modifying moisture content that the ones used in the previous partitioning.

The third statement is still under study. It somehow reflects the diversity of know-how. Depending on the manufacturing unit, cheesemakers seem preferably to use either one or the other of CD and SD to control moisture content.

Both rule and variable selection may be improved.

The main drawback with rule selection is that neutral rules are eliminated by the consequent part level criterion, as explained in section 3.1. However they contain the knowledge to get a standard product, and are activated by numerous data points. Once the reduction is done, it would be interesting to expand the selected rule set by reintroducing some of the neutral ones according to new criteria yet to be developed.

Variable selection is based upon rule distance. Rule distance definition given in equation 6 can be improved. One way would be to consider than an elementary distance, for a given factor, between `0` ant `2` is double than the elementary distance between `0` and `1`, or between `1` and `2`. This is particularly true if the factors `0`, `1` and `2` correspond respectively to the linguistic concepts of `SMALL`, `MEDIUM` and `LARGE`. Another complementary way would introduce a weight on variables, which could be adjusted during the variable selection step.

5   Conclusion

The objective of this work was knowledge discovery in food industry process data bases. A self learning fuzzy inference system selects among all the possible rules those which are best suited to the data. A variable selection step eliminates variables which do not present a real interest for control purposes. These two iterative procedures lead to a small set of selected rules and variables.

The method has been applied to a cheesemaking process. The discovered knowledge has been confronted with the expert knowledge. The discovered knowledge partly confirms the expert one. It also brings out new questions, especially by highlighting context dependent effects of some variables. Later on, the rule base will be introduced into the process control loop to build the fuzzy inference engine. On line learning should be used to validate and improve it.

The rule selection procedure eliminates neutral rules. Nevertheless, these are important for the process, they contain the recipe to make a standard product. Further work will have to develop new criteria to keep neutral rules.

References

Charnomordic, B., Glaudel, M., Renard, Y., Noel, Y., & Vila, J. (1998). Knowledge-based fuzzy control in food industry : A decision support tool for cheesemaking. In *Bio-Decision*, Montpellier, France.

Chiang, C.-K., Chung, H.-Y., & Lin, J.-J. (1997). A self-learning fuzzy logic controller using genetic algorithms with reinforcements. *IEEE Transactions on Fuzzy Systems*, *5 (3)*, 460–467.

Goldberg, D. E. (1989). *Genetic Algorithm in Search, Optimization and Machine Learning*. Addison Wesley.

Jang, J.-S. R. (1994). Structure determination in fuzzy modeling: a fuzzy cart approach. In *IEEE International Conference on Fuzzy Systems*, Orlando, Florida.

Leitch, D. D. & Probert, P., J. (1998). New techniques for genetic development of fuzzy controllers. *IEEE Trans. on Sustems, Man and Cybernetics : Part C, Applications and Reviews*, *28 1*, 112–123.

Mamdani, E. (1994). Twenty years of fuzzy control: Experiences gained and lessons learnt. In *IEEE Technical Update Series, Fuzzy Logic Technology and Applications*, (pp. 19–24)., New York City, New York.

Perrot, N., G.Trystram, Guenec, D. L., & Guely, F. (1995). Evaluation temps rel de la couleur de biscuits. comparaison entre deux approches : floue et baysienne. In *Rencontres Francophones sur la Logique Floue et ses Applications*, (pp. 169–176).

Acknowledgments