Quality of the Information: The application in the winification process in wine production

JORGE RIBEIRO[1], PAULO NOVAIS[2], JOSÉ NEVES[3], MANUEL DELGADO[4]
[1]School of Technology and Management, Viana do Castelo Polytechnic Institute, Viana do Castelo, PORTUGAL, jribeiro@estg.ipvc.pt http://www.estg.ipvc.pt
[2]Department of Computer Science University of Minho, PORTUGAL, jneves@di.uminho.pt,
[3]pjon@di.uminho.pt, http://www.di.uminho.pt
[4]Department of Electronic and Computation , University of Santiago de Compostela, SPAIN,
manuel.fernandez.delgado@usc.es , http://www.dec.usc.es/

Abstract: - Knowledge representation techniques as a way to describe the real world, based on mechanical, logical or other means, will be, always, a function of the systems ability to describe the existing world. Knowledge and belief are generally incomplete, contradictory, or error sensitive, being desirable to use formal tools to deal with the problems that arise from the use of incomplete, contradictory, ambiguous, imperfect, or missing information. It is important to evaluate the quality-of-information of the knowledge around the time in order to analyze the best conditions of the universe of the discourse. Based on the vinification process of the wine production this paper analyzes the quality-of-information in the various stages of the process applying Extended Logic Programming as logic mathematical functions.

Key-Words: - Data Mining; Knowledge Representation, Quality of the Information, Wine vinification process

1 Introduction

Nowadays, with the improvement of information systems in general, the register of the wine production process is stored in databases that with time become bulky. The process of knowledge extraction in repositories with a large amount of information, extracting relationships between data helps organizations managers in their decision making and also to trace new marketing strategies as the evaluation of the internal processes of their organization. Although the information is stored in these repositories is quite common to find incomplete information in these data which hampers the performance of the tools for the extraction of analytical information. It is necessary to extract information of quality and for this reason the analysis of its quality can potentiate the creation of the best scenarios for production and trade in order to capitalize the business strategies. These types of features available in the data contribute to the performance of any data mining technique.

The vinification process is one of the stages of the wines production [1] that influence the achievement of wines quality. This assessment is traditionally realized by wine tasters that analyze some subjective parameters such as colour, foam, flavour and savour. However in such scenarios the existing information contains inaccurate or unknown values. Additionally, the values of some chemical attributes are related to others. Although the Data Mining techniques can be applied to various classical problems like classification, regression, optimization, etc, there are some data that are discarded because it contains null values. However, the quality of the information contained in the data set becomes very important to analyze in one hand, the quality of information throughout the various stages of the winemaking process (or even in the production of wine), and in another, consider what is the best scenario in terms of attributes values that can maximize the quality of information. Thus the traditional analysis of statistical data set out by managers of wine production can be complemented with the analysis of the quality of information analysed during the process.

This paper presents the analysis of the quality of the information obtained during the vinification process of “red green” wines in an agricultural cooperative in the north region of Portugal. The data set is characterized by a set of attributes where the information is contains a set of restrictions and known and unknown information that influences the evaluation of the quality of the wines. The objective of this paper is to present the knowledge representation of the information obtained through the process of wine production and review the quality of information. With this method it can be considered what is the best scenario to obtain wines with a better quality.

2 Knowledge Representation and Quality of the Information

Knowledge representation techniques as a way to describe the real world, based on mechanical, logical or other means, will be, always, a function of the systems ability to describe the existing world. Therefore, in the conception of a knowledge representation system, it must be object of attention different instances of knowledge [2]:

- Existent Knowledge: it may not be known in all its extension.
- Observed Knowledge: it is acquired by the experience, and obtained by contact or observation.
- Represented Knowledge: with respect to a certain situation, it may be relevant to represent a given set of information. In spite of all the limitations, it is possible that observations made by different individuals, with distinct education and motivations, show the same set of fundamental data, function of its utility.

This work is supported by the developments in [3, 4] where the representation of incomplete information and the reasoning based on partial assumptions is studied, using the representation of null values to characterize abnormal or exceptional situations. The identification of null values emerges as a strategy for the enumeration of cases, for which one intends to distinguish between situations where the answers are known (true or false) or unknown [4]. The representation of null values will be scoped by the Extended Logic Programming (ELP) [3, 5]. Logic Programming languages (LP) provide a powerful tool for the knowledge representation, since that the non-monotonous characteristic of the negation by failure makes possible to express several common situations not provided by the classical logic. The purpose of extending the LP is to explicitly represent negative information [4].

The review of logic programming accepted in this paper is strictly declarative. The adequacy of a body representing knowledge in a logic programming language means adequacy with respect to the declarative semantics of that language. Every program is associated with a set of abducibles. Abducibles can be seen as hypotheses that provide possible solutions or explanations of given queries, being given here in the form of exceptions to the exceptions of the predicates that make the program.

Definition 1 – Extended Logic Program for Knowledge Base Representation

The knowledge available is made of logic clauses of the form $P_{i+j+1} \leftarrow P_1 \wedge P_2 \wedge \ldots \wedge P_{i-1} \wedge not\ P_i \wedge \ldots \wedge P_{i+j}$, *where* $I, j, k \in N_0$, $P_1, \ldots, P_{i+j}$ *are literals, i.e. formula of the form p or* $\neg p$, *where p is atom, and where* r_k, *not,* P_{i+j+1}, *and* $P_1 \wedge P_2 \wedge \ldots \wedge P_{i-1} \wedge not\ P_i \wedge \ldots \wedge P_{i+j}$ *stand, respectively, for the clause's identifier, the negation-as-failure operator, the rule's consequent, and the rule's antecedent. If i=j=0 the clause is called a fact and is represented as* P_1.

i.e., with respect to the computational model it were considered extended logic programs with two kinds of negation: classical negation represented by "¬" and default negation represented by the symbol "*not*". Intuitively, "*not p*" is true whenever there is no reason to believe *p*, whereas "$\neg p$" requires a proof of the negated literal. An extended logic program (program, for short) is a finite collection of rules and integrity constraints, standing for all their ground instances, and is given in the form:

$$p \leftarrow p_1 \wedge \ldots \wedge p_n,\ not\ q_1, \ldots, not\ q_n\ and$$

$$?\ p_1 \wedge \ldots p_n \wedge not\ q_1 \wedge \ldots \wedge not\ q_m, (n,m \geq 0)$$

where ? is a domain atom denoting falsity, the p_i, q_j, and p are classical ground literals, i.e. either positive atoms or atoms preceded by the classical negation sign "¬" [3].

To reason about the body of knowledge presented in a particular knowledge, set on the base of the formalism referred to above, let us consider a procedure given in terms of the extension of a predicate called demo, using ELP as the logic programming language. This predicate has the objective to implement mechanisms for inference and to interpreter null values (or unknown). Given a question it returns a solution based on a set of assumptions. This meta predicate (definition 2) will be defined as a meta theorem-solver for incomplete information represented by signature demo:T,V $\rightarrow$ {true, false}, infers the valuation V of a theorem T in terms of false, true and unknown according to the following set of productions:

Definition 2 – Meta Theorem Problem Solver for an Universe of Discourse with Incomplete Information: A meta theorem-solver for incomplete information

demo(T, true) ← T.
demo(T,false) ← ¬ T.
demo(T,unknown) ← not T, not ¬T.

Where "¬" and "not" denote, respectively, strong negation and negation-by-failure. The first clause of the definition 2 establish that is a question recurring to a knowledge base of positive information; the second clause determines that the questions reveals false recurring to the negative information and the knowledge represented in the level; and the third clause is based on the concept of unknown or incomplete information is connected to that of null values. These elements are atoms that represent abstract concepts with no particular definition, i.e. elements which have a well-defined (or even non-defined) range of values have valid options.

As example of the applicability of these knowledge representation method, consider the following as a case study to show some examples of how null values can be used to represent unknown situations. This work adopted part of the data collected during a wine production [1] phase during four years in a wine estate which produces and markets "green red" wines. The Table 1 presents the most relevant attributes of the used dataset. Although this study uses these attributes, and in order to make the process simple and a intuitive representation of the knowledge and the analysis of the quality of information, this section will focus only on the representation of the attributes: pH (chemical pH), cA (chemistry age) and Folin-Cioclateau Index (fCI). The example is centred on the problem of the evaluation the quality of the wine vinification in wine process being measured by the attributes pH, cA and fCI:

pH: Month x Value
cA: Month x Value
fCI: Month x Value

where the first argument indicates the month and the second represents the value of the predicate for example, pH (january,3.29) indicates that the value of pH for the january month is 3.29.

Program 1 - Extension of the predicate that describes the pH for January

pH(january,0.32).
¬pH(X,Y) ← not pH(X,Y).

In the program 1, the "¬" symbol denotes the strong negation denoting that must be interpreted as false, and not the term that designates the negation-by-failure. On other perspective, situations of incomplete information may involve different kinds of nulls. One of the null types to be considered stands for an unknown value. As an example, let us suppose that the data set does not specify the value of the cA attribute and it is effectively unknown. Following the example of the program 1, it can be represented the fact that the value of the cA for January is not known. This situation will be represented by a null value, of unknown type, that will permit say that the cA for the January month exists but is unknown (Program 2).

Program 2: Information about the cA with an unknown value

cA(january, ⊥).
¬cA(X,Y) ← not cA(X,Y),
not $exception_{cA}$(X,Y).
$exception_{cA}$(X,Y) ← cA(X, ⊥).

¬ (cA(X,Y), cA>=67.7, cA<=914.3)

In the program 2, the "⊥" symbol represents the null value of unknown type, in the sense that is a representation that assumes any value but without having been given a clue to end with a value. Computationally, it is not possible to determine from the positive information, the cA value of January, under the description of the exceptional situation (third clause of the program 2). The fourth clause presents the condition where the values of the predicate cA must be between 67.7 and 914.3.

Another situation that can occur is that a value is unknown but it can be one of a set of values. This case is presented in the program 3. The second and third clause's represents the fact that the value for the predicate fCI in January is unknown but can be obtained from a set of two values 0.4 and 0.5. However we must specify an invariant (fourth clause of program 3) that implements the XOR operator, i.e. it states that in January the fCI value is either X or Y, but not an amalgam of both.

Program 3 - Extension of the predicate fCI – Folin-Ciocalteau Index

¬fCI(X,Y) ← not fCI (X,Y),
not $exception_{fCI}$(X,Y).

$exception_{fCI}$(january, 0.4).

$exception_{fCI}$(january, 0.5).

¬ ($exception_{fCI}$(january,X) ∨ $exception_{fCI}$ (january,Y)) ∧ ¬($exception_{fCI}$ (january,X) ∧ $exception_{fCI}$(january,Y))

Indeed, in the search for an answer, it is postulated that each solution to the problem is to be given in terms of a logic theory, built upon the extensions and the exceptions of the predicates that make their realm, i.e. for all problem solutions in memory and for each property inherited by them, their relevance to the answer to be evaluated will be given in terms of a measure of the quality of the information that a program carries along the time. A new solution is set in terms of the problem data, and given as the extension of a set of predicates and the exceptions to these extensions. The new solution is evaluated in terms of the quality of information. To model the universe of the discourse in a changing environment, the breeding and executable computer programs can be ordered in terms of the quality-of-information that steams out of them.

Definition 3 – Quality-of-Information of the Knowledge Base Representation

Let k (k ∈ {1,…,m}) denote the predicates whose extensions make an extended logic program or theory that model the universe of discourse. The quality of the information with respect to a generic predicate *k* can be analyzed in four situations and can be measure from the interval [0-1], when the information is positive and negative, when the information is unknown, when the information is unknown but can be selected from one or more values, and when the information is unknown but can be derived from a set of values, but only one can be selected. If the information is know (positive) or false (negative) the quality of the information for the predicate is "1" corresponding to the max value fro the known knowledge. For situations where the value is unknown the formula of the quality of information is given by the formula:

$$QI_k = \lim_{N \to \infty} \frac{1}{N} = 0 (N >> 0) \quad (1)$$

For the third situation the quality of the information of the predicate is given by the formula (2) QI=1/*Card*, where *Card* denotes the cardinality of the exception set for *k*, if the exception set is not disjoint. If the exception set is disjoint (fourth situation) the quality of information is given by:

$$QI_k = \frac{1}{C_1^{card} + \ldots + C_{card}^{card}} \quad (3)$$

where C_{Card}^{Card} is a card-combination subset, with card elements.

To each predicate it is also associated a scoring function V^k_i [min_i, max_i] → 0…1, that gives the score predicate *k* assigns to a value of attribute *i* in the range of its acceptable values, i.e., its domain (for sake of simplicity, scores are kept in the interval 0…1). This characteristic of applying a scoring can be used to give importance to the predicates of an ELP program. The relative importance that a predicate assigns to each of its attributes under observation; w^k_j stands for the relevance of the attribute *i* for predicate *k* (it is also assumed that weights of all predicates are normalized, i.e.

$$\sum_{1 \le i \le n} w_i^k = 1, \text{ for all } i. \quad (4)$$

It is possible to define a predicate's scoring function, i.e, for a value x=(x_1,…,x_n) in the multi-dimensional space defined by the attributes domains, which is given in the form:

$$V^k(t) = \sum_{1 \le i \le n} w^k{}_i \times QI_k(t) \quad (5)$$

where *k* is the predicate, *i* the attributes of the predicate *k* and *t* the time. It is now possible to measure the quality of the information that stems from a logic program or theory, by posting QI_k values into a multi-dimensional space, whose axes denote the logic program or theory, with a numbering ranging from 0 (at the center) to 1.

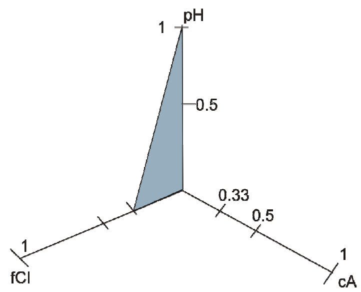

Fig. 1. A measure of the quality-of-information for January.

The area delimited by the arcs gives a measure of the quality of information carried out by each problem's solution under consideration (Figure 1). i.e. using Qk and the extension of predicates pH, cA and iFC, a mapping into an hyperspace is built, and the area delimited by the arcs gives a measure of the quality of information carried out by each case under consideration [4].

3. Materials and Methods

3.1 Wine vinification Data

This work adopted part of the data collected during the wine production phase during four years in a Wine Estate in Minho Region (North of Portugal) that produces and markets "green red wine". During the process of wine vinification it was used three kinds of wine maceration [1] (figure 1): Vinification Maceration by Pellicular fermentative (C), Vinification by carbonic maceration (CM) and by Rotary Cube (CR). For each maceration type it was used five types of glue or clarification type [1]: Polyvinilpolipirrolidona, albumin, gelatin, casein more the witness, without any glue. These characteristics are mentioned in the table 1 the *p*, *a*, *g*, *c* and *t* respectively. The data set has two types of attributes: attributes with chemical characteristics and subjective attributes (colour, foam, flavour and savour). Table 1 presents the attributes of the data set with maximum and minimum values.

Parameter Name	**Domain Values**	
	Min	**Max**
Sample Fermentation (time in months) - SFTM	{6, 8, 12, 14, 24, 30, 36}	
Clarification type – cT	{t, p, a, g, c}	
Vinification Type – vT	{C, MC, CR}	
pH *	3.29	3.9
Absorbency –a420	0.26	1,21
Absorbency –a450	0.31	2.48
Absorbency –a620	0.1	0.5
Chemical Age - cA	67.7	914.3
Folin-Ciocalteeau Index (iFC)	0.2	0.9
Anthocyanines – aNT * (mg/L)	12.6	69.8

Table 1 – The main wine's vinification indicators

Invariants

The invariant has the particularity to present restrictions or limitations to the variables of the data set or express some restrictions to preserve the coherency of the universe of the discourse. An example of an invariant is the condition that the pH value takes values on the interval 3.29 and 3.9 corresponding to the min and max values of the predicate domains (table 1). Another invariant states that the values of the clarificant type of value must be on the set {t, p, a, g, c}. The information about null values of unknown type but enumerated assume the restrictions of the XOR operator and only one value can be assumed from the set of possibles. These types of invariants are represented in the section 3.2.2.

Importance of the parameters

The managers of the wine state specify the importance of the attributes of the parameters (predicates) as follows: 30% for the pH, 10% for the TC (type clarification) and VT (vinification type) and 50% for the value of the predicate. However, the parameterization of the importance of attributes can be changed dynamically. In this sense values will be assigned to each predicates defined in formula (2).

3.2 Knowledge representation of the wine vinification data

The knowledge representation of the data set presented in the section 3.1 is mapped in 7 predicates corresponding to the main parameters of the data set: pH, A420, A520, A620, aNT, cA and iFC. Knowledge Representation for the predicates is represented as:

pH: SFTM $\times$ cT $\times$ vT $\times$ Value
a420: SFTM $\times$ cT $\times$ vT $\times$ Value
a520: SFTM $\times$ cT $\times$ vT $\times$ Value
a620: SFTM $\times$ cT $\times$ vT $\times$ Value
cA: SFTM $\times$ cT $\times$ vT $\times$ Value
fCI: SFTM $\times$ cT $\times$ vT $\times$ Value
aNT: SFTM $\times$ cT $\times$ vT $\times$ Value

where the first parameter corresponds to the Sample-Fermentation-Time-Months value, the second corresponds to the Clarificant Type, the third the vinification type and the fourth corresponds to the value of the predicate.

3.2.1 Positive and Negative Information

The positive information corresponds to information that is known at a specific time. Based on the data presented in table 9 on which the information is known, in the table 2 we present a short information about the positive information representation. These predicate states that the value of the predicate pH with the clarification type "t" and the vinification type "c" in 6 months of the sample fermentation is known and has a value.

pH(6, t, c, 3.472). pH(6, p, c, 3.475). … a420(6, t, c, 0.714). a420 (6, p, c, 0.674). … a520(6, t, c, 1.514). a520 (6, p, c, 1.441). …	a620(6, t, c, 0.31). a620 (6, p, c, 0.262). … aNT(6, t, c, 827.3). aNT (6, p, c, 724.6). … cA(6, t, c, 3.472). cA (6, p, c, 3.475). … iFC(6, t, c,50.5). iFC(6, p, c, 44.7).

Table 2 – positive information.

The negative information is one that specifies that a given information is false. Has an example of this type of information the following predicates states that the value of the predicate pH and a520 with the

ISSN: 1790-5117

ISBN: 978-960-474-088-8

clarification type "t" and the Vinification Type "c" in 6 months of the sample fermentation is false with the value 5.6 and 0.4 respectively (table 3).

```
not pH(6, t, c, 5.6).
not a520(6, p, c, 0.4).
```

Table 3 – negative information.

The quality of the information for the predicate with positive and negative information is evaluated by the definition (1). Following these guidelines, the quality of the information for the clarification type "t" and the cinification type "c" in 6 months of the sample fermentation is: $QI_{pH(6,"t","c")} = 1$

3.2.2 Invariants, restrictions and representation of null values

As we mentioned the invariants corresponds to constraints or restrictions to guarantee the consistency of the universe of discourse. For the representation of the restrictions we present in the table 4 two examples: the first corresponds to the range (min, max) values presented in the table 1 in special for the parameter Chemistry Age (cA). The first invariant states that the cA predicate takes that the Sample-Fermentation-Time-Months argument is in the set {6,8,12,14,18,20,24,26,30,36}, the Clarificant Type is in the set {t,p,a,g,c} and the vinification type is in the set {c, mc, cr}. The second type of restriction occur when the cA has the value 6 and independently of the clarification type and the vinification type, the value of the cA must be in the interval [3.29-3.5].

$\neg$(cA(X,Y,Z,W) $\wedge$ (W$\geq$0.2 $\wedge$ W$\leq$0.29) $\wedge$ (X=6 $\vee$ X=8 $\vee$ X=12 $\vee$ X=14 $\vee$ X=18 $\vee$ X=20 $\vee$ X=24 $\vee$ X=26 $\vee$ X=30 $\vee$ X=36) $\wedge$ (Y=t $\vee$ Y=p $\vee$ Y=a $\vee$ Y=g $\vee$ Y=c) $\wedge$ (Z=c $\vee$ Z=mc $\vee$ Z=cr)).

$\neg$(cA(X,Y,Z,W) $\wedge$ W$\geq$3.29 $\wedge$ W$\leq$3.5).

Table 4 – restrictions representation of restrictions

The principal situation of the occurrence of the invariants is presented in the

3.2.3 Exceptions or null values

The situations of incomplete information may involve different kinds of nulls [5]: null values of unknown type, null values of unknown type but enumerated.

a) **Null values of unknown type**

The values presented in the table 9 as NT-1 represent situations with a null value of the type unknown that should allow the conclusion that the value of the wine exists but to which it is not possible to be affirmative. This type of information is indicated in the table 9 by NT-1* and is represented by:

cA (6, g, c, $\perp$).

$\neg$ cA(X,Y,Z,W) $\leftarrow$ not cA(X,Y,Z,W),

not exception$_{cA}$(X,Y,Z,W).

exception$_{cA}$(X,Y,Z,W) $\leftarrow$ cA(X,Y,Z,$\perp$).

Table 5 – Null values of unknown type

where the symbol $\perp$ represents a null value of an undefined type, in the sense that any value is a potential solution but without concluding about which value one is speaking about. Computationally, it is not possible to determine from the positive information. For this type of nulls, the quality of the information is given by the formula (3):

$$QI_{cA,6,"g","c"} = \lim_{N\to\infty} \frac{1}{N} = 0 (N >> 0)$$

b) Null values of type unknown values but enumerated

The values presented in the table 8 as NT-2 represent situations of null values of unknown type but enumerated. This situation suggests that the lack of knowledge may only be associated to an enumerated set of possible values. For example, for the predicate iFC the value is unknown but is in a range between 0.2 e 0.4. This situation is represented in the Table 6 as NT-2* and situations of these type as NT-2.

$\neg$ iFC(X,Y,Z,W) $\leftarrow$ not iFC(X,Y,Z,W),

not exception$_{iFC}$(X,Y,Z,W).

exception$_{iFC}$(6,Y,Z,W) $\leftarrow$ W $\geq$ 0.2 $\wedge$ W $\leq$ 0.4.

Table 6 – Null values of unknown type but from a range of possibilities

In this situation, the quality of the information is given by the formula (1):

$$QI_{fCI,6} = \lim_{N\to\infty} \frac{1}{N} = 0 (N >> 0)$$

However there are situations in which the value of an attribute is unknown but may be defined by a finite set of situations. In this scenario there are two situations. The first one is that it's possible to select one and only one value of the set and the other situation is that in certain circumstances it is possible to assume not one value but more than one. In the table 7 we present the situation for the predicate a520 with SFTM equal to 8 months, the clarification type equal to "t" and vinification type equal to "c" where it's value is unknown but can be derived by an enumerated set of possibilities with an inclusive selection, i.e. it can be chosen one or two values at the same time.

$\neg$ a520(X,Y,Z,W) $\leftarrow$ not a520(X,Y,Z,W),
not eception$_{a520}$(X,Y,Z,W).
exception$_{a520}$(8,t,c,0.44).
exception$_{a520}$(8,t,c,0.45).

Table 7 – Null values of unknown type but

In this case the Quality of the information of the null values of unknown type but enumerated (situation 1) is given by the formula (1): QK=1/*Card* = ½ = 0.5

The other situation is when we have an unknown value from an enumerated type and it only can be selected one value. For example, for the predicate a520 the value for the SFTM equal to 8, the clarification type equal to “t” and the vinification type equal to “c” is unknown but can be derived from a finitive set of values. However, only one value can be assumed.

$\neg$ a520(X,Y,Z,W) $\leftarrow$ not a520(X,Y,Z,W),
not eception$_{a520}$(X,Y,Z,W).
exception$_{a520}$(8,t,c,0.44).
exception$_{a520}$(8,t,c,0.45).
$\neg$ (exception$_{a520}$(8,t,c,A) $\vee$ exception$_{a520}$ (8,t,c,B)) $\wedge$ $\neg$(exception$_{a520}$ (8,t,c,A) $\wedge$ exception$_{a520}$ (8,t,c,B))

Table 8– Null values of unknown type but enumerated with exclusive selection.

In this situation the quality of the information is given by the formula (4):

$$QI_{a520,8,t,c} = \frac{1}{^{card}C_1 + \ldots + ^{card}C_{card}} = \frac{1}{3} = 0.33$$

The quality of the information can be represented in the hyperspace represented by the axis corresponding to the predicates or parameters of the dataset.

In the figure 2 (a) we present a simple example of the quality-of-information for the month 8 of the vinification process and in the (b) the quality of information for the period 6, 8 and 12 months of the fermentation process.

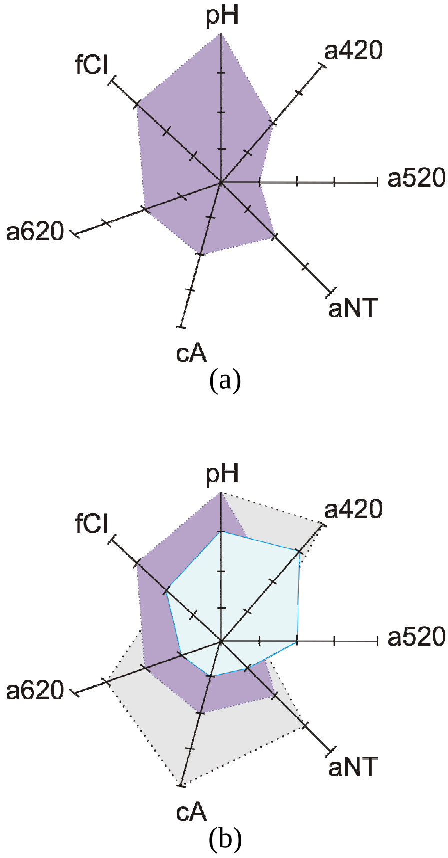

Fig. 2 – Quality-of-information for all

4 Conclusion

Knowledge representation techniques as a way to describe the real world, based on mechanical, logical or other means, will be, always, a function of the systems ability to describe the existing world. Knowledge and belief are generally incomplete, contradictory, or error sensitive, being desirable to use formal tools to deal with the problems that arise from the use of incomplete, contradictory, ambiguous, imperfect, or missing information. It is important to evaluate the quality-of-information of the knowledge around the time in order to analyze the best conditions of the universe of the discourse. In this paper we apply the incomplete knowledge representation to a vinification process in wine production. We were able to evaluate the quality of the information during the process and identify the best scenario to maximize the quality-of-the information. It will be possible to identify some incomplete information from a set of unknown information and complement the Knowledge Discovery tools with information traditionally discarded. In the future it should be interesting also to consider a new set of chemical parameters in the wine production

References:

[1] Castillo-Sanchez, J.X., Arantes J. et Maia, M.O. Étude de l' Évolution des Composés Phénoliques des Vins du Nord du Portugal Issues des Différentes Processus de Vinification. In: Polyphenols Comunications 96 Vol. I, 18th International Conference on Polyphenols, July 15-18, Bordeaux, pp: 55-56, 1996.

[2] Neves, J., A Logic Interpreter to Handle Time and Negation in Logic Data Bases. In: Proceedings of ACM 1984 Annual Conference, San Francisco, USA (October 24-27, 1984).

[3] Neves J., Machado, J., Analide C., Novais, P., Abelha, A., Extended Logic Programming applied to the specialization of multi-agent systems and their computing environments, in proceedings of the International Conference on Intelligent Processing Systems, pp:159-164, 1997.

[4] Analide, C., Novais, P., Neves, J., Quality of Knowledge in Virtual Entities, Encyclopedia of Communities of Practice in Information and Knowledge Management, Elayne Coakes and Steve Clarke (Eds), Idea Group Reference, pp:436-442, 2006.

[5] Baral, C., Gelfond, M., Logic Programming and Knowledge Representation, research report, computer science department, University of Texas at EL Paso, USA, 1994.

[6] Analide C., Novais P., Machado J., and Neves, J., Anthropopathy and its Assessment in Virtual Entities“, proceedings of the IRMA 2004 International Conference, New Oreleans, USA, 2004

[7] Zoecklein, B., Fugelsang, K., Gump, B. , Wine Analysis & Production, New York : Chapman & Hall, 1995.

Apendix A

Table 6: Data set values of the wine vinification production case study

Sample-Fermentation-Time-Months	Clarificant Type	Vinification Type	pH	A_{420}	A_{520}	A_{620}	Ant(mg/L)	cA	iFC
6	t	C	3,472	0,714	1,514	0,31	827,3	0,701	50,5
6	p	C	3,475	0,674	1,441	0,262	NT-2	0,734	44,7
6	a	C	NT-1	0,728	1,572	0,279	781,3	0,745	42,6
6	g	C	3,47	0,718	1,548	0,276	762,2	0,727	49,4
6	c	C	3,469	0,744	1,535	0,286	670,1	0,726	NT-2
…	…	…	…	…	…	…	…	…	…
12	t	C	3,499	0,667	1,35	0,23	355,3	0,556	54,2
12	p	C	NT-2	0,63	1,209	0,205	288	0,546	48
12	a	C	3,523	0,659	1,119	0,222	313,1	NT-2	44,1
12	g	C	3,5	0,671	1,399	0,239	318,6	0,48	57,1
12	c	C	3,582	0,666	1,395	NT-2	266	0,521	44,6
18	t	C	3,48	0,633	1,031	0,271	341	0,415	45,5
18	p	C	3,486	0,529	0,923	0,216	NT-2	0,455	37,2
…	…	…	…	…	…	…	…	…	…
18	a	C	3,484	0,569	0,981	0,222	311,7	0,442	39,7
18	g	C	3,482	0,561	0,98	0,229	313,7	0,444	39,5
18	c	C	3,481	NT-2	0,96	0,221	260,3	0,445	38,3
24	t	C	3,484	0,559	0,85	0,228	225,7	0,382	41,9

24	p	C	3,486	0,51	0,828	NT-1	230,3	0,413	34,4
24	a	C	3,488	0,519	0,835	0,205	222,3	0,359	NT-1
24	g	C	3,489	0,515	0,812	0,201	236,3	0,399	37,7
24	c	C	3,489	0,535	0,833	0,214	217,7	NT-2	37,1
30	t	C	3,493	0,546	0,811	0,211	195,6	0,371	41,7
30	p	C	3,495	0,515	0,795	0,2	165,1	0,394	30,2
30	a	C	3,496	0,52	0,797	0,199	177,6	0,395	32,7
…	…	…	…	…	…	…	…	…	…
30	g	C	3,499	0,529	0,739	0,213	176,3	0,392	26,4
30	c	C	3,497	0,511	0,8	0,199	179,4	0,387	NT-1
36	t	C	3,513	0,566	0,778	0,219	142,3	0,325	43,9
36	p	C	3,515	0,502	NT-1	0,201	126,3	0,324	32,6
36	a	C	3,516	0,527	0,751	0,207	138,3	0,321	36,8
36	g	C	3,511	0,522	0,739	NT-2	125,1	0,321	38,2
36	c	C	3,517	0,516	0,731	0,205	143,2	0,341	39,5
8	t	MC	3,562	0,431	0,696	0,182	439,9	0,51	41,8
8	p	MC	3,611	0,374	0,618	0,153	307,4	0,533	33,4
8	a	MC	3,568	0,394	0,644	0,154	320	0,502	37,2
8	c	MC	NT-2	0,419	0,663	0,167	306,1	0,502	37,9
14	t	MC	3,577	0,429	0,665	0,169	201,6	0,416	61
14	a	MC	3,573	0,421	0,641	0,153	185,7	0,394	56,5
14	g	MC	3,584	0,455	0,658	0,161	171,1	0,363	52,9
14	c	MC	3,582	0,454	0,649	0,165	168,3	0,367	54,1
…	…	…	…	…	…	…	…	…	…

ISSN: 1790-5117

ISBN: 978-960-474-088-8