An iterative approach to build relevant ontology-aware data-driven models

Rallou Thomopoulos, Sébastien Destercke, Brigitte Charnomordic, Johnson Iyan, Joel Abecassis

An iterative approach to build relevant ontology-aware data-driven models

Rallou Thomopoulos[a,d], Sébastien Destercke[b,*], Brigitte Charnomordic[c], Iyan Johnson[a,c], Joël Abécassis[a]

[a] *IATE Joint Research Unit, UMR1208, CIRAD-INRA-Supagro-Univ. Montpellier II, 2 place P. Viala, F-34060 Montpellier cedex 1, France*
[b] *HEUDIASYC Joint Research Unit, UMR 7253, Centre de recherche de Royallieu, UTC, F-60205 Compiegne cedex, France*
[c] *MISTEA Joint Research Unit,UMR729, INRA-SupAgro, 2 place P. Viala, F-34060 Montpellier, France*
[d] *INRIA GraphIK, LIRMM, 161 rue Ada, F-34392 Montpellier cedex 5, France*

Abstract

In many fields involving complex environments or living organisms, data-driven models are useful to make simulations in order to extrapolate costly experiments and to design decision-support tools. Learning methods can be used to build interpretable models from data. However, to be really useful, such models must be trusted by their users. From this perspective, the domain expert knowledge can be collected and modelled to help guiding the learning process and to increase the confidence in the resulting models, as well as their relevance. Another issue is to design relevant ontologies to formalize complex knowledge. Interpretable predictive models can help in this matter. In this paper, we propose a generic iterative approach to design ontology-aware and relevant data-driven models. It is based upon an ontology to model the domain knowledge and a learning method to build the interpretable models (decision trees in this paper). Subjective and objective evaluations are both involved in the process. A case study in the domain of Food Industry demonstrates the interest of this approach.

Keywords: Ontology, machine learning, classification tree, expert knowledge, knowledge integration

1. Introduction

Expertise sharing and learning from data is of great importance for building efficient decision-support tools, especially in domains where extensive mathematical knowledge is not available. This happens in various fields: Life Sciences [28], owing to the great variability of living organisms and to the difficulty of finding universal deterministic natural laws in biology; decision problems dealing with complex environments and scarce experimental data, such as tunnel construction [31]; risk analysis problems

*Corresponding author
Email address: rallou.thomopoulos@supagro.inra.fr (Rallou Thomopoulos)

involving many variables and complex systems, such as radiological risks [10]. In food science, from which our case study arises, many areas (food processing, cultural practices, transformation processes) rely as much upon expertise and data as upon mathematical models. When no complete mathematical model is available, the increasing amount of available data makes it possible to use learning techniques to build predictive models. These data-driven models can then be embedded into decision-support tools, to predict the values of variables of interest. They offer various advantages (efficiency, compliance with experimental design ...) and alleviate the expert work load.

A first key issue of these models is their reliability. For experts to use models learnt from data, they must be confident in the results. How to improve this confidence? Even if such confidence can be partially obtained by a validation procedure, *interpretable* models have been shown to be more trusted by the experts, since they can understand the path of reasoning behind the prediction. Examples of interpretable learning models are decision trees, fuzzy rule bases [13], Bayesian networks, etc.

A second (related) key issue concerns the relevance of the data sets used by learning methods. Assuming that data are ideally structured and relevant to the situation are very strong assumptions, often unsatisfied, especially when data come from various sources and different experimental set-ups. In Life Sciences as in other areas involving complex systems, experimental data are frequently dedicated to the study of a specific scientific question, and not collected with a global approach. This is due to the fact that covering all system aspects would require very costly experiments. Therefore nothing in the raw data set guarantees that variables will be relevant when used in a learning model designed for a wider purpose. In [35], a statistical approach to deal with this issue is applied, based on the meta-analysis of prediction results obtained from mixed data from different sources. However this approach requires to have a validated predictive model (e.g. in [35], an equation modelling bacterial growth). In the present paper, we consider the more puzzling case when no validated model is available. Then expert interviews are fundamental to evaluate built model relevance. However they are also time-consuming tasks. AI researchers who have been interested in eliciting qualitative knowledge detained by the experts, know the difficulties of this task [11]. Therefore expert intervention must remain limited, and be guided to be efficient.

Other issues such as considering non-totally reliable data or incomplete information will not be addressed in this paper, but will be discussed in the perspectives.

Our contribution consists in a method to build data-driven models which is:

- collaborative, since it makes AI learning methods and experts interact;
- iterative, since it involves several cycles to improve the obtained results;
- hybrid, since it relies both on data and knowledge.

The proposed approach aims to achieve three related goals: (i) to find relevant explanatory variables, (ii) to structure and enrich the domain knowledge in an ontology, (iii) to increase the expert confidence in the model. Its principle is the following. Starting from an initial data set and knowledge represented in an ontology, an initial data-driven model is learnt (step 1). This model is first evaluated with objective numerical criteria. Then it is submitted to domain experts, who may enrich the ontology by suggesting

new relations between some variables (step 2). Transformations are applied to the data according to these new relations (step 3). The goal is to transform variables that were deemed irrelevant by the experts into significant ones. The whole process is repeated iteratively until no possible further improvement is detected, eventually reaching an accurate (in term of predictions) model matching expert knowledge and data. Figure 1 describes this process.

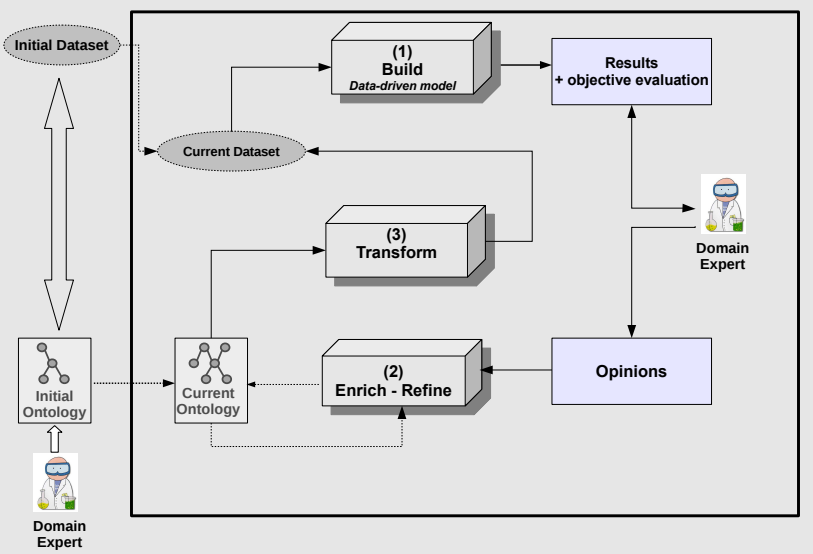

Figure 1: Outline of the proposed design approach

This paper is organized as follows: Section 2 discusses the relevant literature. Section 3 is dedicated to the ontology definition and its interplay with data. Section 4 formally describes the various data processing operations done using the ontology. Section 5 details the proposed approach and compares it to similar ones. Section 6 gives some elements about decision trees, which will be used as the learning method in the present paper. A case study concerning the impact of agri-food transformation processes on the nutritional quality of wheat-based products is presented in Section 7. A few possible steps towards an increased automation of the approach are discussed in Section 8, and some concluding remarks are given in Section 9. All along the paper, we illustrate the generic approach by taking examples from different fields.

2. Relevant literature

Both ontologies and data-driven models take more and more importance, due to the increasing amount of available (complex) data and to the need to model (qualitative) knowledge and structural information (especially in the World Wide Web). However, there are still very few attempts to propose generic methods combining both ap-

proaches. This is especially true in the fields of experimental sciences and in Life Sciences in particular. In this latter field, most existing approaches are in the domain of bioinformatics [24] with very specific concerns regarding biomedicine and genomics.

Independently from each other, ontologies and data-driven models take more and more importance in the field of Life Sciences [36], the former for their ability to model and structure qualitative domain knowledge, the latter for their ability to provide efficient predictive models without deep mathematical knowledge of the complex involved processes (e.g. [30] in hydrology).

2.1. Ontology

An ontology can be defined in multiple ways (see e.g. [12]). It may be limited to a simple taxonomy, or include more complex relations and be subdivided into subontologies.

An ontology can model expert and domain knowledge by the means of concepts and relations linking them. Using ontologies offers several assets [21]:

1. it allows to share a common understanding of structured information [20];
2. it makes the specificities of domain knowledge explicit;
3. it establishes a basic structure that facilitates the interaction with domain experts, and makes easier the identification of ambiguities or inappropriate model choices.

The ontology can be built manually or automatically. Semi-automatic design requiring expert validation presents many advantages [34]. Indeed, acquiring ontological knowledge is usually hard to achieve and time-consuming for experts, hence any advance towards more automated procedures is useful.

Still, ontological knowledge alone is not designed to provide accurate numerical or symbolic predictions, outside of logical inferences. It is therefore tempting to combine such ontological knowledge with learning methods extracting (statistical) relationships from data and resulting in more quantitative assessments. There are recent attempts to combine expert knowledge and learning methods that we now describe.

2.2. Use of ontologies to guide data mining

In data mining, differentiating significant extracted patterns from useless ones is a delicate task. It therefore seems natural to use available knowledge to recognize such significant patterns.

Some attempts concern cases where data are well-structured, making automation easier. An example is image classification, with the use of visual concepts [16]. Other works focus on problems where scalability is a main issue, and where the method performances can be automatically measured by numerical assessments. In fields where large amounts of data must be treated, such as Semantic Web mining [33], semi-automatic collaborative approaches to guide the data mining process have recently been applied. In [23], the proposal deals with ontology evaluation and enrichment. This is done using multiple ontologies together with a text-mining approach on domain-specific texts and glossaries. Algorithms and software for collaborative discovery from semantically heterogeneous information sources are described in [7].

The case of inductive learning using ontologies, data and decision trees has been addressed in [39], however it is limited to the specific case of taxonomies[1]. Similar studies are applied to Bayes classifiers in [38].

2.3. Use of subjective analysis for rule or data selection

There are also cases outside experimental sciences where the (fully) automated use of ontological knowledge appears difficult, and where the involvement of experts seems unavoidable in order to improve data-driven method results. In such situations, the use of models interpretable by experts is essential.

For instance, [1] propose an expert-driven validation of groups of rules to facilitate rule validation of rule-based models. Ling et al. [15] propose an approach for effective data selection or efficient Data Mining applicable to domains where data stores are too extensive and detailed, and existing knowledge too complex. It involves the human experts pervasively, taking advantage of their expertise at each step, while using Data Mining techniques to assist in discovering data trends and in verifying the expert findings. In a recent paper [17], the authors focus on the use of ontologies to facilitate post-processing of association rules by domain experts. They propose a hybrid pruning method involving the use of objective and subjective analysis.

Note that most of these methods are not based on a feedback procedure. There are two cases:

- ontological knowledge or expert knowledge are used to improve data-driven learning results;
- data-driven learning is used to identify potential elements of ontological knowledge or to supplement expert knowledge,

Both tasks, i.e., acquiring ontological knowledge and building understandable models from data in which the expert(s) can be confident(s), are usually hard to achieve and suffer from some limitations. It therefore seems natural to build methods that aims at taking the best of each.

2.4. Decision trees as interpretable models

Decision tree algorithms are efficient approaches for data-driven discovery of complex and non obvious relationships. Their readability and the absence of *a priori* assumptions explain their popularity. They are particularly useful for variable selection in highly multidimensional problems, therefore they are ideal to display statistically important variables on which the expert should focus. Decision trees can be pruned and, as thoroughly discussed in [2], not too complex. Such a low complexity is essential for the model to be interpretable, as confirmed by Miller's conclusions [18] relative to the *magical number* seven. They are therefore ideal candidates for a method in which experts have to interact with data-driven models.

However, data-driven models are highly dependent on the available data and on the learning method. While they can give pretty good predictions, they may do so by using

[1] Tree-structured ontologies.

variables or data modalities that the expert would consider as improper to describe a given phenomenon. This could be due to the fact that those variables appear to be only marginally involved in the process, or because they only become significant in some context, or in combination with other variables. Details about decision trees are given in Section 6.

3. Ontology definition in relation with data

We consider a domain description composed of two elements:

- a set of data descriptions, described by a list of experiments and the values they assume on a set of variables;
- an ontology containing the lexical domain knowledge.

The ontology Ω is defined as a tuple $\Omega = \{\mathcal{C}, \mathcal{R}\}$ where $\mathcal{C}$ is a set of concepts and $\mathcal{R}$ is a set of relations. An example of ontology used for food processes is shown in Figure 2 (arrows correspond to the subsumption relation 3.3). We will use this ontology as a running example in the sequel.

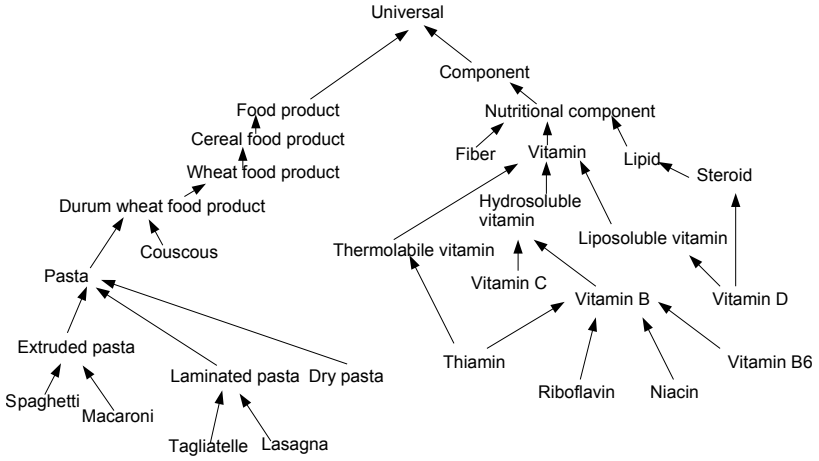

Figure 2: A small part of the ontology used for food processes

3.1. Concept range

A concept c may be associated with a definition domain by the $Range$ function. This definition domain can be:

- *numeric*, i.e. $Range(c)$ is a closed interval $[min_c, max_c]$;
- *hierarchized symbolic*, i.e. $Range(c)$ is a set of partially ordered elements, that are themselves concepts belonging to $\mathcal{C}$.

Spaces to which the range function maps a concept c correspond to the spaces on which an associated variable takes its values. That is, if X_k is associated to concept c, then $Range(c)$ is the space on which X_k takes its values.

3.2. Relationship between concepts and variables

We consider a data set coming from actual observations on N experiments, with K variables. Each row is an instance of an experiment, and includes the corresponding observations. We assume that each variable X_k, $k = 1, \ldots, K$, is a concept $c \in \mathcal{C}$ in the ontology Ω. The nth value of the kth variable is denoted by $x_{k,n}$, as illustrated in Table 1. $x_{k,n}$ belongs to $Range(X_k)$, either symbolic or numeric. Thus the set of concepts $\mathcal{C}$ includes the set of variables used in data descriptions.

observation	X_1	X_2	$\ldots$	X_K
exp_1	$x_{1,1}$	$x_{2,1}$	$\ldots$	$x_{K,1}$
exp_2	$x_{1,2}$	$x_{2,2}$	$\ldots$	$x_{K,2}$
$\ldots$	$\ldots$	$\ldots$	$\ldots$	$\ldots$
exp_N	$x_{1,N}$	$x_{2,N}$	$\ldots$	$x_{K,N}$

Table 1: Experimental data set where $\{X_k\}, k \in [1, K]$, is the set of variables and $\{exp_n\}, n \in [1, N]$, is the set of observations from the experiments.

3.3. Set of relations

The set of relations $\mathcal{R}$ is composed of:

1. the *subsumption* relation, also called the 'kind of' relation and denoted by $\preceq$, which defines a partial order over $\mathcal{C}$. Given a concept $c \in \mathcal{C}$, we denote by $\mathcal{C}_c$ the set of sub-concepts of c, such that:

$$\mathcal{C}_c = \{c' \in \mathcal{C} | c' \preceq c\}. \quad (1)$$

When c represents a variable with hierarchized symbolic definition domain, we have $Range(c) = \mathcal{C}_c$. For the sake of conciseness, we use $\mathcal{C}_c$ in the sequel whenever possible. For example, in Figure 2 and for $c =$ *Laminated pasta*, we have $\mathcal{C}_{\text{Laminated pasta}} = \{$*Tagliatelle*, *Lasagna*$\}$;

2. a set of *functional dependencies*. A functional dependency FD expresses a constraint between two sets of variables and is represented as a relation between two sets of concepts of $\mathcal{C}$. A set of concepts $X = \{X_{k_1}, \ldots, X_{k_2}\} \subseteq \mathcal{C}$, $1 \leq k_1 \leq k_2 \leq K$ is said to functionally determine[2] another (disjoint) set of concepts $Y = \{Y_{k_3}, \ldots, Y_{k_4}\} \subseteq \mathcal{C}$, $1 \leq k_3 \leq k_4 \leq K$ if and only if, $\forall n_1, n_2 \in [1, N]$:

$$(\forall X_k \in X, x_{k,n_1} = x_{k,n_2}) \Rightarrow (\forall Y_{k'} \in Y, y_{k',n_1} = y_{k',n_2}).$$

This specific kind of relations is necessary in our approach to formalize some dependencies between two variables. Functional dependency FD between X and Y performs a mapping from the specific values taken by the X variables onto the specific values taken by the Y variables. This function will be denoted by $DetVal_{FD}$:

$$DetVal_{FD} : Range(X_{k_1}) \times \ldots \times Range(X_{k_2}) \rightarrow Range(Y_{k_3}) \times \ldots \times Range(Y_{k_4}).$$

Two instances of such functional dependencies are required in our approach:

- a *property* relation $\mathcal{P} : \mathcal{C} \rightarrow 2^{|\mathcal{C}|}$ that maps a single concept to a set of other concepts, which represent a set of associated properties.
  For each concept that has some properties, i.e., $\forall c \in \mathcal{C}$, $\mathcal{P}(c) \neq \emptyset$, we denote by p_c the number of properties and by $\mathcal{P}(c)_i$ the ith element of $\mathcal{P}(c)$, $i = 1, \ldots, p_c$.

  Example 1. *Consider the concept $Couscous$ in Figure 2, a kind of $Durum\ wheat\ food\ product$. Couscous can be characterized by its grain size (small, medium or large) and its type (white or whole-grain), i.e.*

  $$\mathcal{P}(\text{Couscous}) = \{\textit{Grain size, Type}\},$$

  and we have $\mathcal{P}(\text{Couscous})_2 = \text{Type}$.

  Example 2. *In chemistry, the concept $Molecule$ could have the following properties, e.g.,*

  $$\mathcal{P}(\text{Molecule}) = \{\textit{Liposoluble, Molar mass}\},$$

  and many others.

  The function $DetVal_{\mathcal{P}}$ will be denoted by $\mathcal{HP}_c$ (for $HasProperty$). It maps a particular element of $Range(c)$ to the particular property values it takes[3] in the ranges of the concepts of $\mathcal{P}(c)$:

  $$\mathcal{HP}_c : Range(c) \rightarrow Range(\mathcal{P}(c)_1) \times \ldots \times Range(\mathcal{P}(c)_{p_c}) \quad (2)$$

  We denote by $\mathcal{HP}_{c\downarrow i} : Range(c) \rightarrow Range(\mathcal{P}(c)_i)$ the restriction of $\mathcal{HP}_c$ to its ith property, that is $\mathcal{HP}_{c\downarrow i} = \mathcal{HP}_c \cap (Range(c) \times Range(\mathcal{P}(c)_i)$.

[2] X is often called the determinant set and Y the dependent set.

[3] Note that a given set of particular properties does not uniquely define a sub-concept satisfying those properties, i.e. the relation is not injective.

Example 3. *For the particular sub-concept White small-grain couscous* $\in Range(Couscous)$*, we have*

$$\mathcal{HP}_c(\textit{White small-grain couscous}) = \{small, white\}$$

and $\mathcal{HP}_{c\downarrow 2}(\textit{White small-grain couscous}) = white$

Example 4. *Pursuing with the chemistry example, we would have for* $NaCl$ *(salt)*

$$\mathcal{HP}_c(\textit{NaCl}) = \{Yes, 58.4\}$$

- a *determines* relation $\mathcal{D} : 2^{|\mathcal{C}|} \rightarrow \mathcal{C}$ which maps the values of a subset of concepts to the value taken by another concept. Typical examples of such relations are equations linking some input parameters to some output parameters.

  Example 5. *When characterizing gas transfer through some material, permeation is defined as the product between the material thickness and permeance (which characterizes the gas transfer rate through the material by unit of partial pressure differences). If* $permeation$*,* $permeance$ *and* $thickness$ *are three concepts, then*

  $$\mathcal{D}(\{\textit{permeance,thickness}\}) = \textit{permeation}$$

  models the fact that permeation value can be inferred from permeance (a material property) and thickness values.

  The function $DetVal_{\mathcal{D}}$ will be denoted by $\mathcal{HD}_C$ (for $HasDetermination$). $\forall C \in 2^{|\mathcal{C}|}$ such that $\mathcal{D}(C) \neq \emptyset$, we define the function $\mathcal{HD}_C$ such that:

  $$\mathcal{HD}_C : Range(c_1) \times \ldots \times Range(c_{|C|}) \rightarrow Range(\mathcal{D}(C)). \qquad (3)$$

  with c_i and $|C|$ being respectively the ith element and the number of elements of C. The function $\mathcal{HD}$ defines the values of $\mathcal{D}(C)$, given the values of the determinant variables. In the *permeation* case given previously, the result is the product of the two determinant variables.

4. Data processing using ontologies

When using algorithms on data to train a predictive model, some input variables and/or their modalities may be irrelevant to the problem at hand, at least for the expert, even if statistically significant. Indeed, experimental data reported in papers, reports, etc., were usually collected for highly specialized research objectives and may not entirely fit in a knowledge engineering approach. The aim of the data processing techniques proposed below is to transform irrelevant variables into significant ones for the expert. Doing so, it can be hoped that their future treatment will lead to more meaningful models, in which the expert will be more confident.

Techniques presented here require both the ontology and expert feedback. Such feedback may be stimulated by a third-party data treatment method, e.g., fuzzy rule

bases, decision trees (the case here),.... The appropriate techniques to use on given data then depend on the expert feedback. Here we present some relations and transformations that we think are common to most experimental sciences, however there may be situations where additional specific relations are needed.

4.1. Replacement of a variable by new ones

This process consists in replacing a given variable by some of its (more relevant) properties that become new variables. Let X_k be a variable such that $\mathcal{P}(X_k) \neq \emptyset$. For each property $\mathcal{P}(X_k)_i$, $i \in [1; p_{X_k}]$ (or a subset of them), we create a new variable X_{K+i} such that:

$$\forall n \in [1; N] \quad x_{K+i,n} = \mathcal{HP}_{X_k}(x_{k,n})_{\downarrow i} \tag{4}$$

with $\mathcal{HP}_{X_k}(x_{k,n})_{\downarrow i}$ the projection of $\mathcal{HP}_{X_k}(x_{k,n})$ on $Range(\mathcal{P}(X_k)_i)$, recalling that $\mathcal{P}(X_k)_i$ is the ith element of $\mathcal{P}(X_k)$.

Example 6. *Let X_k = Couscous be the (non relevant) variable to be replaced and $\mathcal{P}(Couscous) = \{$Grain size, Type$\}$ its retained properties. The two new variables created from Couscous are X_{K+1} = Grain size and X_{K+2} = Type.*

Now, assume that for the ith experiment, $x_{k,i}$ = White small-grain couscous, then the two new values for the ith experiment are $x_{K+1,i} = \mathcal{HP}_{X_k\downarrow 1}(x_{k,i}) = small$ and $x_{K+2,i} = \mathcal{HP}_{X_k\downarrow 2}(x_{k,i}) = white$. The initial variable X_k = Couscous is removed.

4.2. Grouping the modalities of a variable using common properties

In some cases, it may be useful or significant to group modalities by using one of their specific features, simply because this feature is suspected to play an important role in the process. Note that this is different from creating a new property with this feature, since in this case it is desirable to keep the original variable (which may convey additional information), only rearranging it. Also, the grouping may be done according to multiple properties, while creating a new property focuses on only one property. Formally, this is equivalent to considering elements of the power set of modalities, these elements being chosen w.r.t. some properties of the variable.

Let X_k be a given variable such that $\mathcal{P}(X_k) \neq \emptyset$ and let $i \in [1; p_{X_k}]$. We replace X_k by X'_k such that, for $n \in [1; N]$:

$$z_n = \mathcal{HP}_{X_k}(x_{k,n})_{\downarrow i}, z_n \in Range(\mathcal{P}(X_k)_i) \tag{5}$$

$$x'_{k,n} = \mathcal{HP}_{X_k\downarrow i}^{-1}(z_n) \tag{6}$$

Equation (5) expresses that we first get z_n, the ith property value associated with $x_{k,n}$. Equation (6) expresses that we then search for all the antecedents of this value, i.e. all the $x_{k,l}$ ($l \in [1; N]$) whose ith property value is equal to z_n, which includes $x_{k,n}$ but may also include other values. Finally, elements $x'_{k,n}$ of the partition induced by the grouping may be modelled in the ontology by new concepts related to the property.

Example 7. *Let $X_k = Water$ and $pH \in \mathcal{P}(Water)$. Suppose that we want to keep track of the types of water used in the processes, but that it would be desirable to group them by pH. We can have $\mathcal{HP}_{Water}(Tap\ water)_{\downarrow pH} = Basic\ pH$, and $\mathcal{HP}_{Water}(c)_{\downarrow pH} = Neutral\ pH$ for any other type of water (e.g., Deionized water,Distilled water, Distilled deionized water). The new variable X'_k would have the following two modalities:*

$$\{\textit{Tap Water}\} \quad and \quad \{\textit{Deionized water, Distilled water, Distilled deionized water}\}.$$

Classifying plants by their genotypes or radioactive elements by their speed of decay are other examples. Note that in general, the main purpose of such grouping is to have an increased readability by providing a partitioning of modalities making sense to the expert.

Note that another possibility would be to work with the whole power set of the modalities of variable X_k, to select the most relevant partition by optimization procedures (for example, selecting the partition that induces the lowest error rate) and then to check for its significance with an expert. However, such a procedure, if not guided by ontology knowledge to reduce the number of partitions to consider, is a greedy one, and would imply an exponential complexity growth with the number of modalities. Moreover, the result of such a procedure is not guaranteed to be meaningful for the experts.

4.3. Selection of a level of abstraction for a symbolic variable

Sometimes the description granularity of a variable within a data set is too fine for a purpose different from the one for which measurements were originally collected. In this case, it makes sense to choose a coarser description for the variable, that would remove some unnecessary information from the data and facilitate result interpretation. The selection process for choosing the right level of abstraction can be either automated according to accuracy and complexity criteria or decided according to expertise (with a preference for this latter procedure in our case).

In any case, such a processing assumes, of course, that a proper coarsening can be defined. Therefore, for this particular operation, we will restrict ourselves to hierarchized symbolic variables X_k, such that $\mathcal{C}_{X_k}$ has the structure of a rooted tree (w.r.t. the order induced by $\preceq$). For example, in Figure 2, we can apply the procedure to $\mathcal{C}_{Pasta}$, but not to $\mathcal{C}_{Vitamin}$ of the same figure, because the $Thiamin$ concept has two parents.

Definition 1. *A* **path** *in $\mathcal{C}_{X_k}$ is an ordered sequence of concepts from the subtree root to a leaf. Two concepts are* **disjoint** *if there is no path they both belong to.*

Definition 2. *A* **cut** *in $\mathcal{C}_{X_k}$ is a set of disjoint concepts such that every path goes through one and only one concept of the set.*

A cut describes a partition of tree leaves and, therefore, a coarsening of the original modalities. Let $\mathcal{U}$ be a chosen tree cut. We replace each element $x_{k,n}$ by $x'_{k,n}$ in the

following way:

$$\exists c \in \mathcal{U}, x_{k,n} \preceq c \Rightarrow x'_{k,n} = c \tag{7}$$

$$\exists c \in \mathcal{U}, c \preceq x_{k,n} \Rightarrow x'_{k,n} = x_{k,n} \tag{8}$$

The tree structure, together with Definitions 1 and 2, ensures the unicity of c in Eq. (7).

Example 8. *Consider the variable $Pasta$ together with $\mathcal{C}_{Pasta}$ (see Figure 2). Then Pasta, Laminated pasta, Tagliatelle form a path. Spaghetti and Laminated pasta are disjoint, while Laminated pasta and Tagliatelle are not. $\{$Extruded pasta, Tagliatelle, Lasagna$\}$ is a set of disjoint concepts but does not form a cut, since there are paths (e.g., Pasta, Dry pasta) to which none of them belongs. The completed set $\{$Extruded pasta, Tagliatelle, Lasagna, Dry pasta$\}$ does form a cut. Using this cut is interesting if Spaghetti and Macaroni are considered as too detailed and thus replaced by Extruded pasta. In the choice of this cut, it is judged important to know which kind of laminated pasta we are considering, while just knowing the type of other pasta is enough (e.g., because the impact of the studied process is particularly sensitive for laminated pasta).*

Example 9. *Store data sets typically contain detailed bills of their costumers. However, it may be useful in data-mining approaches to group some products by type and/or brands. For instance, it may be meaningful for the variable $Drink$ to group together Orange juice, Grape juice, ... in a unique concept Fruit juices, while keeping the granularity for the soda brands. These concepts would remain separate and not be grouped together in a Soda concept.*

4.4. Merging of variables in order to create a new one

It may be interesting to merge several variables into another variable, with the values of the latter defined by the values of the former. It both facilitates the interpretation (as less variables are considered) and avoids considering as significant a single variable that is only significant (at least from an expert point of view) in conjunction with other variables. Let $C = \{X_{k_1}, \ldots, X_{k_{|C|}}\} \in 2^{\mathcal{X}}$ such that $\mathcal{D}(C) \neq \emptyset$. We define a new variable:

$$X_{K+1} = \mathcal{D}(\{X_{k_1}, \ldots, X_{k_{|C|}}\}) \tag{9}$$

such that for n $\in [1; N]$:

$$x_{K+1,n} = \mathcal{HD}_C(\{x_{k_1,n}, \ldots, x_{k_{|C|},n}\}) \tag{10}$$

Example 10. *When using manufactured packaging material, thickness is usually already settled and the material specifications include its permeance. However, what ultimately interests the packer is the permeation of the film. Therefore it makes sense to replace the two variables (permeance and thickness) by a new one (permeation). Recall that it is the product of the other two. If the film $thickness = 0.5(mm)$ and its permeance (to some gas) is $permeance = 1.29 * 10^{-13}(mol \cdot m^{-2} \cdot s^{-1} \cdot Pa^{-1})$, then we have $\mathcal{HD}_C(\{0.5, 1.29 * 10^{-13}\}) = 6.45 * 10^{-17}(mol \cdot m^{-1} \cdot s^{-1} \cdot Pa^{-1})$, replacing every item where $thickness = 0.5$ and $permeance = 1.29 * 10^{-13}$ by $permeation = 6.45 * 10^{-17}$. Note that even if they are all numeric here, elements of $\mathcal{HD}_C$ can in general be numeric or symbolic.*

4.5. Discretization of a numeric variable

Discretizing a numeric variable may be useful, either because learning methods requires symbolic variables or because the expert thinks that ranges of numerical values make more sense than precise measurements. Choosing the right discretization can be a tricky question, and in some cases the use of expert knowledge and ontology can help to perform this step.

Let X_k be a variable whose range is numeric, or a variable that usually takes numerical values. The discretization process consists in turning this numeric range into a symbolic one. Let $\mathcal{C}_{X_k}$ be a set of sub-concepts of X_k, each of them having as range a particular interval included in $Range(X_k)$.
Let $\mathcal{P}_{X_k} = \{P_1, \ldots, P_I\} \subset \mathcal{C}_{X_k}$ be a set of concepts whose ranges form a partition of $Range(X_k)$ [4]. The partition $\mathcal{P}_{X_k}$ is then the range of the new symbolic variable X'_k replacing X_k, such that: $\forall n \in [1; N]$: $x'_{k,n} = \{P_i \in \mathcal{P}_{X_k} | x_{k,n} \in Range(P_i)\}$ with $x_{k,n}$ the initial numerical value.
For instance, in Example 1, the three concepts *Small-grain couscous*, *Medium-grain couscous* and *Large-grain couscous* with the respective ranges $[130, 180]\mu m$, $[180, 450]\mu m$ and $[450, 700]\mu m$ form such a partition and can replace the numerical value of measured couscous (mean) grain sizes.

Note that, in the cases when an ontology-driven discretization is impossible, one can still use classical cluster analysis methods such as a k-means [14] algorithm in order to build discretized variables. These methods are useful to have a first lead on the possible characterization of an unknown variable. Ideally, they are transitional methods pending the knowledge acquisition.

5. A collaborative, iterative, hybrid approach

In this section, we detail the approach outlined in Figure 1. We then compare it to other methods mentioned in Section 2.

5.1. Initial step (iteration 0)

The first step is to build an initial domain knowledge ontology and to learn a first data-driven model, based on a learning method and on an initial data set. Let us denote the "iteration 0" stage by:

$$\mathcal{I}_0 = \{\Omega_0, \mathcal{D}_0, \mathcal{M}_0\},$$

where Ω_0 is the initial ontology, $\mathcal{D}_0$ the initial data set, and $\mathcal{M}_0$ the initial data-driven model.

Ω_0 is built in a semi-automatic way. The method is based on several steps: (i) identifying, within the schema and the values of the data descriptions, the relevant concepts at a most general granularity level; (ii) organizing the identified concepts into a hierarchy, using structural and syntactical criteria; (iii) suggesting relevant complementary concepts, based on textual descriptions. The ontology formal definition and its relation

[4] such as $\cup_{i=1}^{I} Range(P_i) = Range(X_k)$ and $Range(P_i) \cap Range(P_j) = \emptyset$ if $i \neq j$

to data are detailed in Section 3. The ontology may be further completed by including additional bibliographical information and expert knowledge. More details about ontology building are given in [34].

Objective evaluation of the data-driven model is carried out with numerical indicators. Still, one can hardly expect to include all domain and expert knowledge in a first ontology, simply because experts are not able to formalize all their knowledge from scratch, or because some potential problems or needed knowledge have not yet been detected. *Subjective* evaluation is undertaken by confronting the expert with the model and its objective evaluation, in the iterative process.

5.2. The iterative process

An iterative collaborative procedure is started (see Figure 1). It can be summarized as follows:

1. discuss the results with domain expert(s);
2. enrich or refine the ontology on the basis of expert opinions;
3. apply transformations to data;
4. build new data-driven model from transformed data;
5. evaluate the results with objective numerical criteria;
6. start again from step 1, until no further possible improvement is detected.

For example, the experts may detect that, instead of a variable selected as important by the learning process, it is actually one of its properties that plays an important role in the process. Inversely, they can assess that a selected variable is not relevant by itself, but rather that it is the interaction between this variable and another one that is relevant.

The ontology will then be enriched according to expert suggestions, and the data transformed appropriately. The transformations available to modify original data according to the expert feedback are those formalized in Section 4.

More formally, starting from the "iteration n" stage $\mathcal{I}_n = \{\Omega_n, \mathcal{D}_n, \mathcal{M}_n\}$, the "iteration n+1" stage $\mathcal{I}_{n+1} = \{\Omega_{n+1}, \mathcal{D}_{n+1}, \mathcal{M}_{n+1}\}$ is determined as follows.

Definition 3. *Given* $\mathcal{I}_n = \{\Omega_n, \mathcal{D}_n, \mathcal{M}_n\}$,

- $\Omega_{n+1} = \Omega_n \cup \{\mathcal{C}', \mathcal{R}'\}$*, where* $\mathcal{C}'$ *is a set of concepts and* $\mathcal{R}'$ *a set of relations identified by the experts as useful and to be added to the ontology.* $\mathcal{R}'$ *is composed of relations described in Section 3. Note that the knowledge associated to such relations (range,* $\mathcal{HP}_c$*,* $\mathcal{HD}_c$*) also need to be declared along with the relations, otherwise data transformation cannot be performed;*
- $\mathcal{D}_{n+1}$ *is obtained from* $\mathcal{D}_n$ *by applying the following operations:*
  - *if* $\mathcal{R}'$ *includes* property *relations, replace current variables by new ones, according to Section 4.1;*
  - *if* $\mathcal{R}'$ *includes* determines *relations, merge variables into a new one, according to Section 4.4;*
  - *if* $\mathcal{C}'$ *includes re-declarations of a hierarchized symbolic variable by the addition of concepts related to modality grouping, rename variable values according to Section 4.2;*

  - *if $\mathcal{C}'$ includes re-declarations of hierarchized symbolic variable* ranges*, select a level of abstraction, according to Section 4.3;*
  - *if $\mathcal{C}'$ includes re-declarations of numerical variable* ranges*, discretize the concerned variables, according to Section 4.5.*

- *$\mathcal{M}_{n+1}$ is learnt from the data set $\mathcal{D}_{n+1}$.*

Note that, all along the process, expert opinions are essential, as domain experts are the only ones able to pinpoint inconsistencies and non-relevant features (both in the ontology and in the learning method results), and to guide data processing.

5.3. Comparison with existing frameworks

Some approaches that combine ontologies and data mining were mentioned in Sections 2.2 and 2.3. As such combinations are unfrequent, there is no established way to compare the various frameworks. In order to enable such a comparison, some features are summarized in Table 2.

References	Objective		Automation level	Expert feedback	Ontology complexity
	Knowledge Oriented	Predictive Model			
[39],[38]		x	High	None	Taxonomy
[16]		x	High	None	Taxonomy & Spatial Relations
[33],[23]	Ontology learning Instance learning Ontology mapping		Medium	Some	Taxonomy
[17]	Association rule relevance		Low	High	Taxonomy
Proposal	Discriminant feature relevance Ontology refinement	x	Low	High	Taxonomy & Functional Relations

Table 2: Comparison with some other similar frameworks

The first key element in Table 2 is related to the objectives of the approach, which can be classified into two main categories: Knowledge-oriented, or Predictive Model design. When falling into the first category, the framework can be aimed towards more precise goals, either associated to the ontologies themselves (learning, enrichment or mapping) or in relation with them (instance learning, relevance of association rules or discriminant features). In that case, the automation level is low to medium, as such approaches typically requires at least some validation. For approaches under the second category, the automation level is higher, as the procedure is based on numerical criteria, and the expert involvement is not required, contrary to what happens in the first category. The complexity of the ontology is also a factor to be taken into consideration to distinguish between frameworks. In most cases, the ontology is a simple taxonomy.

Among the existing frameworks, our proposal stands out for two reasons: its dual objective (both knowledge oriented and predictive model design) and the ontology complexity. This dual objective allows for an increased interaction between the methods, but requires more interplay between the domain expert and the analyst (Section 8 discusses this topic). The knowledge oriented part currently aims to integrate relevant features into the ontology, by refining and possibly enriching it. Regarding the complexity of the ontology used in our approach, it is clear that there exist more complex ontologies, however to our knowledge such ontologies have not been used in combination with data-mining techniques.

The proposed approach is generic and could be applied to any learning method that provides interpretable results. In the following, it will be illustrated on decision trees.

6. Decision trees

Decision trees are well established learning methods in supervised data mining and statistical multivariate analysis. They can handle classification cases, where the dependent (output) variable is a class, as well as regression ones, where the dependent variable is continuous. In multidimensional modeling, they perform well in attribute selection and are often used prior to further statistical modeling.

In the kind of problems we aim to tackle, it is quite common to encounter variables with only a few corresponding measurements in the available data sets. However it is important to find out if these variables are important features likely to explain the variability of the dependent variable. Besides that, missing values are one of the curses of statistical models and analysis, and an important asset of decision trees is that any observation with values for the dependent variable and at least one independent (input) variable will participate in the modeling.

This section briefly recalls how decision tree learning methods work, as well as the learning criteria used in the two main families of decision trees (C4.5 [26] and CART [6]). Some elements of comparison between the two families are also given.

6.1. Algorithm description and common features

Decision trees were originally designed to handle classification problems, regression trees having been proposed later. Due to their discrete nature, classification trees tend to be easier to interpret, and as in the present study immediate prediction is not our goal, we will focus on classification trees, leaving aside regression trees.

Input to classification decision trees consists of a collection of training cases, each having a tuple of values for a set of input variables, and a discrete output variable divided into classes: $(\mathbf{x_i}, \mathbf{y_i}) = (x_{1,i}, x_{2,i} \dots x_{K,i}, y_i)$. An attribute X_k is continuous or categorical according to whether its range is numeric or symbolic, and takes its values on a domain $\mathcal{X}_k$. The class attribute Y is discrete and can take M_Y distinct values on a finite domain $\mathcal{Y}$. The goal is to learn from the training cases a recursive structure (taking the shape of a rooted tree) consisting of (i) leaf nodes labeled with a class value, and (ii) test nodes (each one associated to a given variable) that can have two or more outcomes, each of these linked to a subtree.

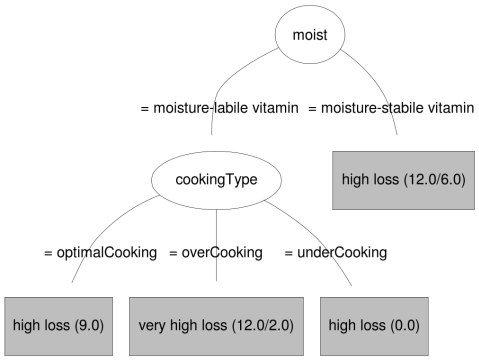

Figure 3: An example of a classification decision tree

Figure 3 displays a simple example of a tree structure. On a given node, the algorithm examines in turn all available variables, and selects the variable that most effectively splits the set of samples into subsets improving the separation between output classes. Once (and if) a variable is selected, a new test node is created that splits on this variable, and the procedure is recursively applied on each (new) node child. At each node, the algorithm stops when no more variables are available, or if there is no improvement by splitting further: the node then becomes a leaf. Due to this graphical nature, decision trees are easily interpretable for a non-expert in statistical or learning methods, and facilitate exchanges with the domain expert. It therefore allows the domain expert to have a qualitative opinion about the tree validity with respect to its own knowledge.

Well-known drawbacks of decision trees are the sensitivity to outliers and the risk of overfitting. To avoid overfitting, cross-validation is included in the procedure and to gain in robustness, a pruning step usually follows the tree growing step (see [25, 6, 26]).

Note that the two kinds of decision trees are seldom used in the same setting: while the CART family [6] (based on binary split) is mostly used by statisticians, the ID3 family [25] (which contains the original idea of C4.5 implementation [26] and is based on n-ary split) is mostly used by artificial intelligence researchers.

6.2. Splitting criteria and comparison

We denote by $p_m(S)$ the proportion of examples at node S that belong to class m.

To select the splitting variable, CART algorithm uses the Gini diversity index I_{Gini}, which is based on squared probabilities and is defined at node S as:

$$I_{Gini}(S) = 1 - \sum_{m=1}^{M_Y} p_m^2(S)$$

On its part, the C4.5 algorithm uses information theory entropy $I_{Entropy}$ as a selection and splitting criterion, which reads, at node S:

$$I_{Entropy}(S) = -\sum_{m=1}^{M_Y} p_m(S) log_2\, p_m(S)$$

The improvement gained by splitting the node S into M_k subsets $S_1, S_2 \ldots S_{M_k}$ according to X_k, is in both cases (CART and C4.5) evaluated as:

$$G(S, X_k) = I(S) - \sum_{i=1}^{M_k} \frac{|S_i|}{|S|} I(S_i)$$

with M_k the number of possible outcomes (2-ary for CART and $|\mathcal{X}_k|$-ary for C4.5, with $|\mathcal{X}_k|$ the cardinality of X_k), and where $I(S)$ is either $I_{Gini}(S)$ or $I_{Entropy}(S)$, respectively when the learning algorithm is CART or C4.5.

Table 3 presents a brief comparison between C4.5 and CART features, to help the reader see the different advantages and drawbacks of each family of decision trees.

	CART	C4.5
Binary splits	Yes	No
Readability	Not so good	High
Variable with multiple modalities	Unbiased	Biased
Pruning	Cost-complexity based	Error based
Compactness	More compact	Less compact
Missing values	Remain unknown	Distributed over known values
Splitting Criterion	Gini Index	Entropy index
Handles regression case	Yes	No

Table 3: C4.5 and CART - a summary of the main differences

6.3. Method evaluation

There are two ways in which the current method can be evaluated:

- subjective human evaluation, performed by experts assessing their confidence in the obtained results, and what are the possible inconsistencies they have detected in the model,
- objective automatic and numerical evaluation, where the results and stability of the predictive models are measured by numerical indices.

Note that, in the context we are working on, human validation appears as the primary goal. Of course, we should require the numerical evaluation not to be degraded too much during the process.

Objective tree quality evaluation

We use different numerical criteria to make an objective evaluation of the different tree qualities at any given learning step.

- The first and the most classical criterion for classification trees is the misclassification rate, $Ec = \frac{MC}{N}$, where MC is the number of misclassified items and N is the data set size. When enough data are available, a cross validation procedure can be applied, which splits the data between a learning set and a test set. Ec is then computed separately on each of these sets. Otherwise, if the whole data set is used in the learning step, Ec is only calculated on the whole data set.

- The confusion matrix gives complementary information, as it shows the misclassification errors for each class.

- Tree complexity: $Nrules + Nnodes/Nrules$, where $Nrules$ is the number of terminal nodes (leaves), which is equivalent to the number of rules, and $Nnodes$ is the total number of nodes in the tree. This criterion takes into account the complexity of each rule (path) as well as the number of rules.

- Tree stability: one drawback of decision trees is the sensitivity to outliers, which may result in unstable splits, i.e. if there is a slight change in the data, the features selected as *best splits* by the algorithm may be completely different.

  Note that tree stability is not provided within the decision tree algorithms themselves. In [22], the authors propose two numerical measures to evaluate tree stability which both involve a training step and a validation one, and their purpose is that there should not be too much variation in the predictive accuracy rate when validating a trained decision tree on new data sets. The first measure is based on the overall accuracy rates for training and validation, denoted respectively by ACC_T and ACC_V. It is defined by $STAB_c = \min(ACC_T/ACC_V, ACC_V/ACC_T)$. The second one is a finer measure defined as a weighted sum of stabilities of individual leaves: $STAB_F = \sum_{l \in leaves} \phi_{V_l} \sigma_{VT_l}$, where ϕ_{V_l} is the proportion of the validation data set items that are associated with the lth leaf, and $\sigma_{VT_l} = \min\{\rho_{V_l}/\rho_{T_l}, \rho_{T_l}/\rho_{V_l}\}$ is the lth individual leaf stability. ρ_{T_l} and ρ_{T_l} are the proportion of the training and validation data set items associated to the leaf that have the leaf assigned class value.

  These measures of stability involve only the predictive part of the tree, i.e. leaf class information. As our goal is to identify with the expert what inconsistencies are present in the reasoning paths of the tree or to point out the paths of missing knowledge, we need a measure of stability that includes split variables and their position in the tree, hence measures such as $STAB_F$ or $STAB_c$ are not sufficient.

So for a rough estimation of the tree stability in those terms, we propose to use a simple criterion based on bootstrapping. Bagging on decision trees has been introduced for the purpose of compensating tree stability problems [4, 5], but with the perspective of improving prediction. It consists in aggregating multiple trees. Each tree is obtained by drawing a bootstrap sample from the multinomial distribution with parameters n and p1=1/n, ..., pn=1/n. The proposed stability criterion is:

$$St^L = \frac{1}{n_b}\frac{1}{L}\sum_{l=1}^{L}(\frac{1}{n_l}\sum_{m\in nodes_l}\sup_k(n_k^{l,m})) \quad (11)$$

L is the number of split levels considered for the study, the calculation being limited to the first L levels, as the first splits are more discriminant than the next ones.

n_b is the number of bootstrap samples while n_l is the average number of nodes at the lth level, those nodes being denoted by $nodes_l$.

$n_k^{l,m}$ is the number of times the tree algorithm selects the kth variable for a split at the lth level along the experiments with the n_b bootstrap samples. The scores of leaf nodes, for which no split is done, are calculated in the same way.

The criterion value varies between $1/n_b$ (unstable) and 1 (stable). It is applicable to binary (CART) trees or C4.5 trees. However, it is rather gross for CART trees, as the same variable may appear several times along a path to a leaf node.

7. Case study: application to food quality prediction

Until recently, cereal food design relied more on experience than on science [8]. Nevertheless, the last 20 years witnessed a considerable increase in the number of research projects. This resulted in an explosion of scientific papers that (even if they are relevant on their scale of observation) can hardly be used outside their original highly specialized field because they have not been completely integrated into a corpus of knowledge.

At the same time, cereal and pasta industry has developed from a traditional industry relying on experience and having a low rate of innovation, to a dynamic industry geared to follow consumer trends: healthy, safe, easy to prepare, pleasant to eat products [9]. To meet the current challenges, food industry needs modern tools and decision support systems integrating all kinds of available knowledge, i.e., expert know-how or knowledge discovered from data.

Previous systems have been proposed in food science, and more specifically in the field of cereal transformation, in order to help prediction, such as [19, 37]. However none of them takes into account both experimental data and expert knowledge, nor proposes solutions in absence of a predetermined (mathematical or expert) model. All these elements motivated our case study.

7.1. Context and description of the case study

We use a knowledge management system concerning the processing and qualities of cereal food products designed to integrate the information coming from different domains. It is organized according to two axes: the ”technical” axis defined by all the unit operations (29 unit operations), which are involved in transformation from raw materials to end products (e.g. grinding, storage, drying, baking, etc), and the ”quality” axis defined by all the criteria (56 criteria) which are used to represent the end-quality(ies) of food products, according to three aspects: organoleptic, nutritional and safety properties (e.g. colors, vitamins contents, pesticides contents, etc).

For each unit operation composing the transformation process, and for each family of product properties, information has been expressed as a data set. The input variables are the parameters of the unit operation. The output variable is the impact of the unit operation on a property (e.g. the variation of vitamin content).

In this study, we focus on the case of the *Cooking in water* unit operation and the evolution of the*Vitamin content* property, which is of particular importance for respecting consumer needs.

This case study concerns 150 experimental data and the initial ontology includes about 150 concepts. Table 4 shows some input variables and values, and the corresponding output variable and its values. The complete list of variables and their ranges are given in Table 5. The *Cooking in water* operation uses complex underlying biochemical processes with many interactions between them. At an operational scale, it is important to design an explanatory model that highlights the most discriminant features and which is not a *black box* model, high accuracy not being a reasonable objective given the lack of data versus the case complexity.

Next sections detail the results obtained for this case study. The ontology was created using the CoGUI interface (*http://www.lirmm.fr/cogui/*). Decision trees were obtained using the R software [27]. The method proposed in this paper is implemented as an extension of R representing about 2000 lines of code.

Id Result	Vitamin	Cooking temperature (°C)	Cooking time (min)	Water	Vitamin decrease (%)
1	Vitamin B6	100	13	NA	-52
2	Riboflavin	100	12	Tap Water	-53
3	Thiamin	100	10	Deionized water	-37
4	Thiamin	98	15	Distilled water	-47
5	Riboflavin	90	10	NA	-18
6	Thiamin	100	NA	Distilled Deionized	-41

Table 4: Part of the training data set

7.2. Application of the approach to the case study

The approach has been carried out with a strong collaboration between a team of four computer science researchers and two food science researchers[5], with a regular involvement of all participants. The output variable is the *Percentage of vitamin loss* during the process, which is a continuous variable, discretized into four ordered classes *Low loss, Average loss, High loss, Very high loss*.

Cooking in water variables	
Pasta type	Spaghetti, Macaroni, Noodles, Tagliatelle, Unknown
Vitamin	Folic acid, Niacin, Panthothenic acid,Riboflavin, Thiamine, Vitamin A, Vitamin B6
Temperature	[90 ; 100] °C
Time	[10 ;30] mn
Kind of Water	Deionized water, distilled Deionized water, Distilled water, Tap water, Unknown
% of salt in water	0% , Unknown
Other process variables (interaction variables)	
% of vitamin addition [a]	[0 ; 162.50]
Grain Variety	Capeiti, Creso, Unknown
Drying cycle	HT, LT, Unknown
Drying duration	[10 ; 85] hours
Drying maximum temperature	[39 ; 86] °C
Flour storage temperature	[4 ; 40] °C
Flour storage duration	[0 ; 6] months
Output variable	
% of vitamin loss	[-99.5 ; -3.70] %

[a]Percentage of the initial value of the vitamin studied in the experiment

Table 5: Variables of the *Cooking in Water* process

The implementation used for decision trees is the R [27] software with the *rpart* package for CART trees, and the *R-WEKA* package for C4.5 trees. The parameters of the algorithm are (i) for rpart: cross validation = 100 , minimum instances per leaf= 6 (default value), (ii) for C4.5: minimum number of instances per leaf= 6. All trees are pruned. They are to be interpreted as follows:

- **rpart** (i) each test node is labeled by the condition on the selected variable for the split at this node. The displayed condition is always the one yielding the left branch. (ii) each leaf node is labeled by the output class, and the number of observations in each class is specified. (iii) a boxplot with the distribution of examples is displayed below each leaf node. It also gives the number of different

[5]B. Cuq (Prof. in Food Science), J. Abécassis (Research Eng. in Cereal Technology), IATE Joint Research Unit

scientific papers which the observations come from. This could be useful to detect biases.

- **C4.5** (i) each test node is labeled by the splitting variable. (ii) for each leaf node, the number of misclassified observations is specified.

The variables and their ranges are described in Table 5. Our approach follows the collaborative approach introduced in section 5. It will be illustrated by four iterations, the initial one using the initial ontology and the variables described in Table 5, e.g. $(\Omega_0, \mathcal{D}_0)$

7.2.1. Iteration 0: initial state

Figures 4 and 5 show the trees trained on the raw data sample, using the CART and the C4.5 algorithm respectively. For both algorithms, the most discriminant variable is *Ingredient Addition* which corresponds to adding vitamins (or not) for compensating a future loss during the cooking process. Then each method selects different features, *Temperature* for CART and *Kind of Water* for C4.5.

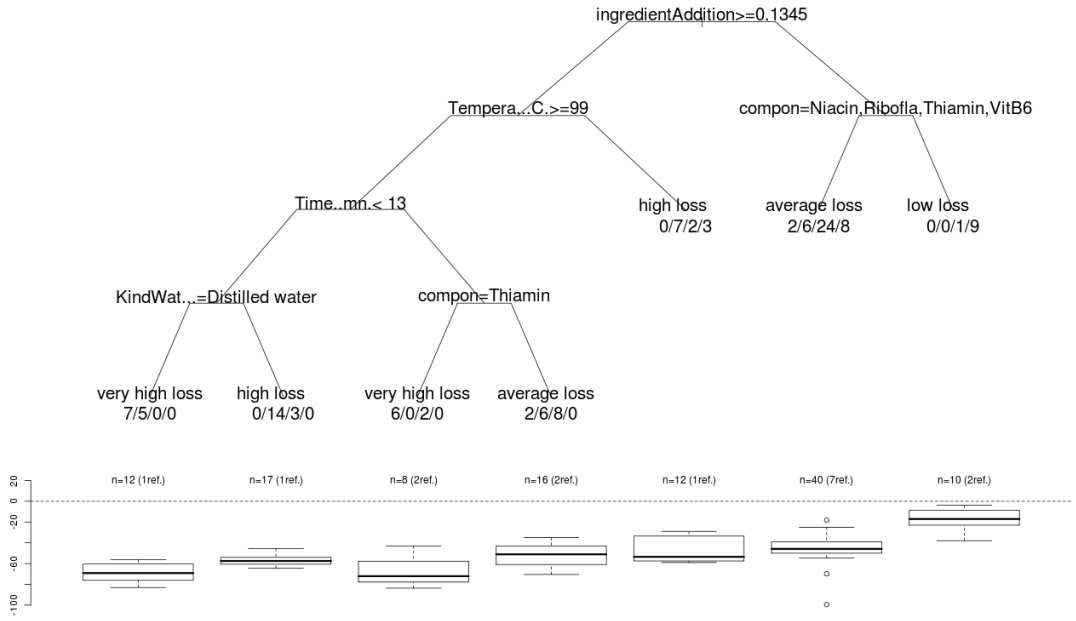

Figure 4: Decision tree generated using the CART algorithm on raw data

7.2.2. Iteration 1: introducing knowledge on Vitamin properties

Discussion with experts The examination of both trees by experts led to the following remarks and adjustments.

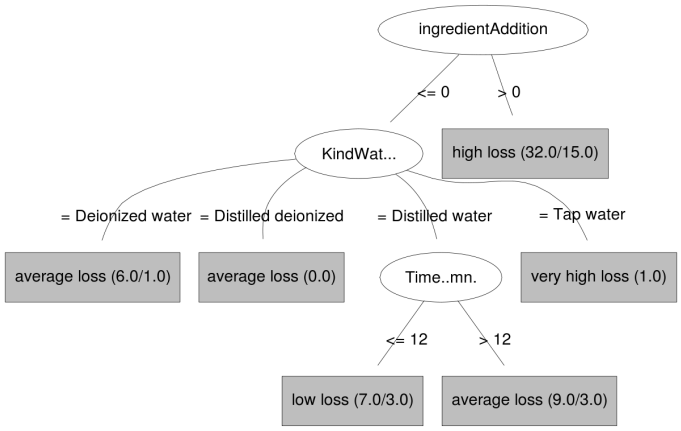

Figure 5: Decision tree generated using the C4.5 algorithm on raw data

The first two splits were not discussed at that stage. What puzzled the experts is the vitamin type only appearing in the CART tree, the corresponding split separating the vitamins into two groups. The experts noticed the difficulty to interpret such a result, as they expected the loss to be explained by vitamin properties rather than by some particular subgroups. Thus they suggested to *enrich the ontology* by characterizing the vitamins by their properties. This required first to perform a change in the ontology, since physical properties of vitamins were not initially separated from the vitamins (see for instance the hydrosoluble vitamin concept in Figure 2).

Ontology completion Compared to the ontology of Figure 2, some concepts ($Thermolabile\ vitamin$, $Liposoluble\ vitamin$, $\ldots$) had first to be transformed by separating properties from vitamin names, designing a property taxonomy in the process. Figure 6 provides an excerpt of the newly built properties taxonomy, together with an illustration of the addition of some relations of the type $\mathcal{P}$ and $\mathcal{HP}$ for Vitamins and Vitamin A concepts, respectively. Some of the added elements are:

$$\mathcal{P}(Vitamin) = \{Solubility, Thermosensitivity, Photosensitivity, \ldots\},$$

$$Range(Thermosensitivity) = \{Thermostability, Thermolability\},$$

$$\mathcal{HP}_{Vitamin}(VitaminA) = \{Liposolubility, Thermolability, Photostability\}.$$

Data transformation The next step was to instantiate the links between the vitamins and their properties (see Equation (2)).

In this case, $X_k = Vitamin$ is the (non relevant) variable to be replaced, and we select the properties $\mathcal{P}(Vitamin) = \{Photosensitivity, Thermosensitivity\}$, as they are the only ones that can have impact on vitamin content during the cooking operation. The two new variables created from Vitamin are $X_{K+1} = Photosensitivity$ and $X_{K+2} = Thermosensitivity$ (the column $X_k = Vitamin$ is removed from

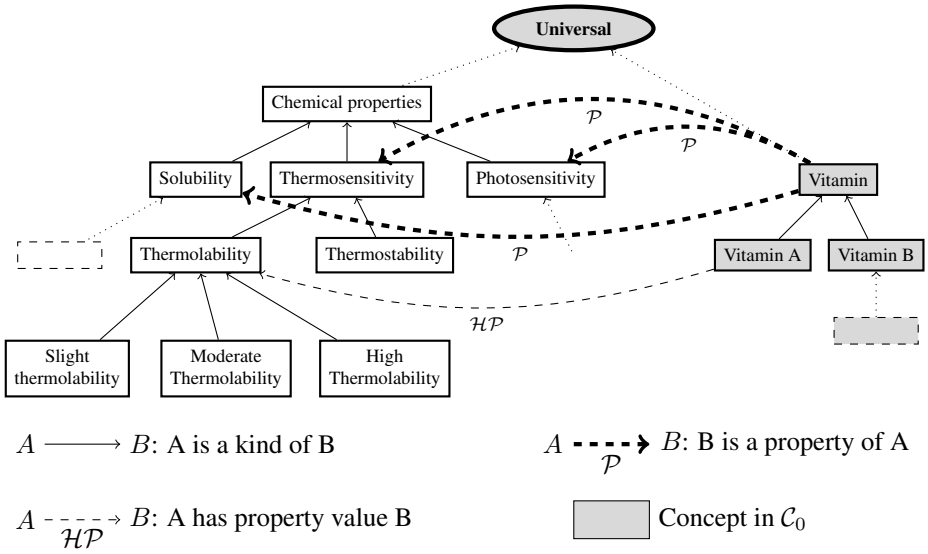

Figure 6: Excerpt of concept and relation additions to Ω_0.

the data set). Now, if for the nth experiment $x_{k,n} = VitaminA$, then the new values are $x_{K+1,n} = \mathcal{HP}_{X_k}(x_{k,n})_1 = Photolabile$ and $x_{K+2,n} = \mathcal{HP}_{X_k}(x_{k,n})_2 = Thermolabile$. All vitamins are replaced by their corresponding properties (see Section 4.1).

Resulting model

The outcome of the learning procedure with $(\Omega_1, \mathcal{D}_1)$ is a new tree model $\mathcal{M}_1$ described by Figure 7. To simplify the reading, we only present the C4.5 tree in this iteration and the next ones.

One newly introduced vitamin property: *Photosensitivity* is selected as a discriminant variable by the algorithm, thus comforting the expert intuition.

7.2.3. Iteration 2: introducing Cooking type and Water properties

During this iteration, expert noticed that the *Kind of water* and the *Cooking time* variables were highlighted by the new tree.

Discussion with experts brought out the fact that the *Cooking time* variable is not relevant by itself, and has significance only if considered together with the *Pasta type*. Indeed, the same *Cooking time* for different types of pasta can lead to different pasta consistencies. When cooking pasta, it is common to differentiate between *Undercooked*, *Over-cooked*, and *Optimally cooked* products. These states depend both on the type of pasta and on the cooking-time, which are usually the available variables in experiments.

Experts also made the remark that the *kind of water* gave only indirect information about the process. They suggested that water can be better characterized in terms of *pH* and of *Hardness*. This better translates the fact that a high content of minerals in

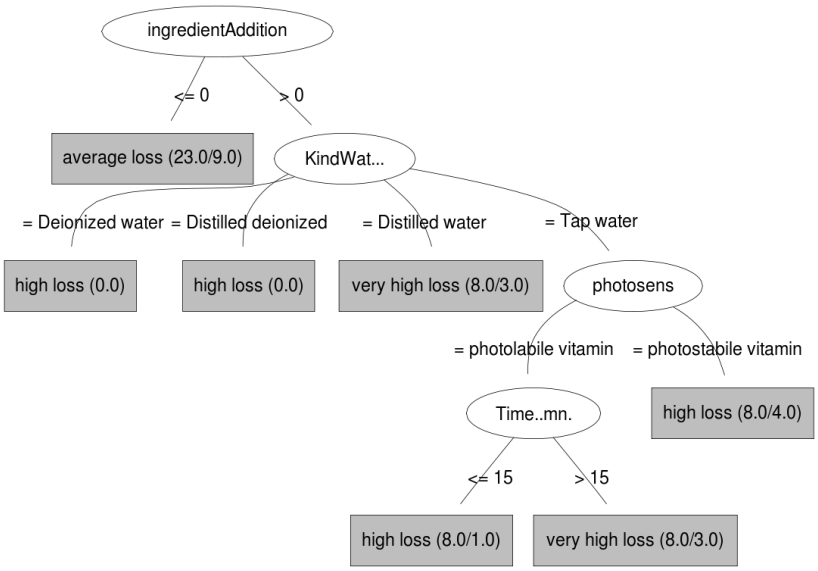

Figure 7: Decision tree (C4.5 algorithm) obtained after Iteration 1

an acid solution increases the amount of possible chemical reactions with vitamins.

Ontology completion The *Cookingtype* concept was added to the ontology, with the three modalities mentioned above.

In the available experiments, the water *pH* and *Hardness* were not measured. However they can be reconstructed from the water types. We first added the *Hardness* and *pH* concepts (Figure 8 illustrates the pH part of the ontology together with the corresponding numerical ranges). The cut chosen for the pH values is $\{AcidpH, NeutralpH, BasicpH\}$ (see Section 4.3). *pH* is also a good example of a numerical variable that has been discretized (see Section 4.5). The following relations (see Equation (3)) were also added.

$$\mathcal{D}(\{Pastatype, Cookingtime\}) = Cookingtype,$$

$$\mathcal{HD}(\{short, 18min\}) = Overcooking,$$

along with other combinations of $Pastatype$ and $Cookingtime$. Note that the $\mathcal{HD}_C$ function takes here both a numeric and a symbolic variable as arguments.

Similarly, the following relations were added to the ontology (see Equation (2)).

$$\mathcal{P}(Water) = \{pH, Hardness\}$$

$$Range(ph) = \{AcidpH, NeutralpH, BasicpH\}$$

$$\mathcal{HP}_{water}(Tapwater) = \{NeutralpH, Hard\}$$

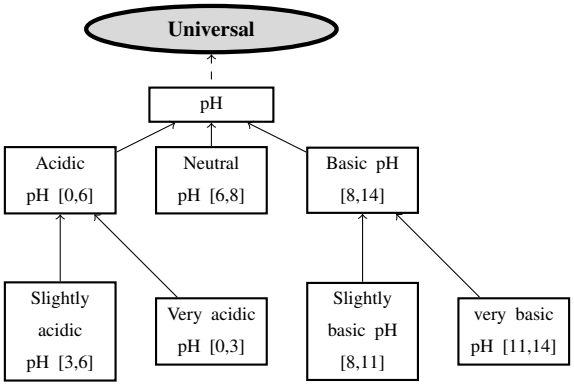

Figure 8: Addition of pH concepts to the ontology. $A \rightarrow B$: A is a kind of B.

Data transformation Two operations were performed: instantiating the variable *Cookingtype*, for instance every experiment where *Cooking time*=18 and *Pasta type*=*Short* was replaced by *Over-cooked* (see Section 4.4), and replacing the kind of water variable by both its *Hardness* and *pH* (see Section 4.1).

Resulting model

Figure 9 shows the $\mathcal{M}_2$ tree obtained using $(\Omega_2, \mathcal{D}_2)$. The *Hardness* variable is now selected for the second split, and the *Cookingtype* one appears further down the tree.

7.2.4. Iteration 3 (final): introducing the Cooking pH

Discussion with experts While experts expressed their feeling that the model was more readable, they highlighted the existence of a link between *Water hardness* and *pH* evolution. Namely, the water pH evolution, which plays an important role in vitamin degradation, depends both on the *Cooking temperature* and on the *Water hardness* (boiling water activating some specific chemical reactions).

Ontology completion To reflect this phenomenon, a new variable was created, as described in Equation (3), and according to a few expert rules not detailed here.

$$\mathcal{D}(\{pH, Temperature\}) = CookingpH$$

Data transformation was done accordingly.

Resulting model Figure 10 displays the final $\mathcal{M}_3$ tree model, issued from learning with $(\Omega_3, \mathcal{D}_3)$. When comparing this tree with the original one, we note that some continuous variables which had been measured through the experiments, such as *Cooking time*, are now replaced by more meaningful ones, such as *Cooking type*, which is obtained by conjunction with one more concept introduced in the ontology, i.e. *Pasta type*. At the end, experts were rather satisfied with this tree, and more willing to trust it than the previous ones.

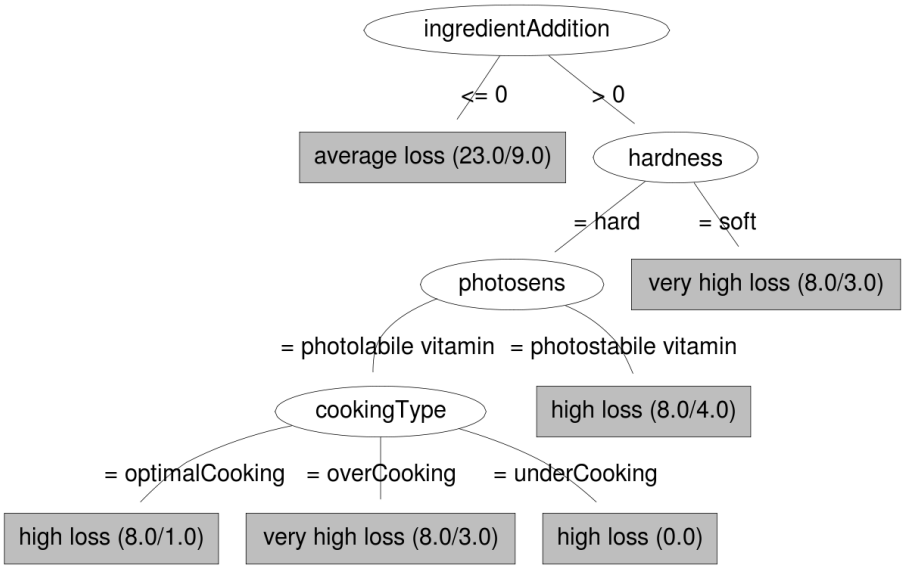

Figure 9: Decision tree (C4.5 algorithm) obtained after Iteration 2

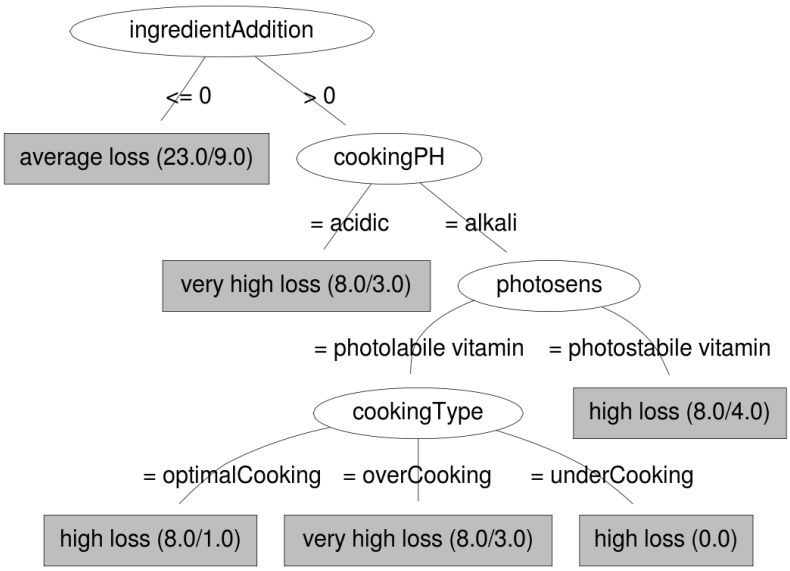

Figure 10: Final decision tree

7.2.5. Result discussion and evaluation

As the method makes ontological knowledge and learning methods interact, we can assess the results obtained for both parts.

Concerning ontological knowledge, the method has resulted in a significant change of the ontology, these changes better reflecting expert knowledge about various points. These changes are illustrated all along the iterations. Among noticeable changes, we can cite the separation of chemical properties from the vitamin sub-ontology, the characterization of water types in term of chemical properties or the addition of new significant variables to estimate the impact of water cooking.

The final ontology includes 220 concepts. Obviously, it could not be easy for experts to point out the useful ones from the start, and guidance saves a lot of time.

Most of these changes can be useful for other case studies in the application domain, such as the characterisation of some elements in terms of their properties, or the description of different pH levels. Results concerning knowledge collection and integration were therefore very promising.

Table 6 gives some quantitative results about the C4.5 trees obtained after each iteration. For each iteration, the second and third column show respectively the misclassification rate and tree complexity. The last column displays the value of the stability index proposed in Equation 11, for $L = 2$, corresponding to the two first tree levels, and calculated using 7 bootstrap samples. Not only did the trees seem more meaningful to the experts, they also make more accurate prediction, as shown by the decreasing misclassification rate. Complexity remained roughly the same after every iteration, therefore not making the tree more difficult to read (on the contrary, since appearing variables were more meaningful to the experts). This shows that the method has greatly improved the learned model quality, both in terms of numerical accuracy and of understandability. The stability values show that the first and the last trees are the most stables of all. The good stability of the first tree is essentially due to the stability of the first split, the second level being more unstable.

Iteration #	MC rate (%)	Complexity	Stability St^2 ($n_b = 7$)
1	44	7.3	0.75
2	48	8.4	0.61
3	35	7.5	0.54
4	35	7.5	0.75

Table 6: C4.5 tree objective evaluations

A thorough examination of the confusion matrices (not given here) also showed that most classification errors were caused by the prediction of a label close to the observed one.

8. Practical steps towards an automation of the proposed method

Using the successive trees and confronting them to the expert opinions allowed us to enrich our ontology with several new concepts and relations that could eventually prove

useful in the future. Some of them were very generic (e.g., the addition of vitamins properties rather than the mere name of vitamins), while others were more specific to the problem at hand (e.g., the characterisation of the *cooking type* depending of both the *pasta type* and the *cooking time*).

Surprisingly, little attention has been paid to the interest of a collaborative framework in the literature, perhaps because the main motivation to develop automated methods is to do without expert knowledge, arguing that expert knowledge elicitation is a hard task. So there is a trend to design more and more automated learning methods.

With regard to this concern, a limitation of the proposed method is its relative lack of automation. Indeed, in order to validate the various changes and analyze them, frequent meetings between domain experts and analysts are necessary. Finding time slots for such meetings can be hard, hence it would be desirable to develop solutions that would lower this frequency, for example by automatically proposing interesting data transformations to the experts or by detecting some possible inconsistencies in the used variables. Although there are many fields (such as experimental sciences) where suppressing all interactions between the expert and the analyst by using a fully automated method is clearly undesirable, advancing towards semi-automated methods would allow discussions to converge more quickly to core problems. Another limitation of the current work is that imperfect (e.g., imprecise, missing) data are not handled. Some future directions to tackle these problems are proposed. They are four-fold:

- the first step consists in ensuring that the inefficiency of the used variables is indeed due to a lack of knowledge, i.e. excluding other possible reasons which could justify that the used variables are not explanatory enough;
- the second step relies on the analysis of the obtained decision trees, in order to identify missing knowledge;
- the third step consists in enriching the ontology, so that transformations of current variables into more meaningful ones for the studied phenomenon are proposed;
- the fourth consists in defining methods that can handle imperfect knowledge, such as missing data or multi-valued functions.

8.1. Excluding external bias that would lead to variable inefficiency

The first guarantee for the quality of the obtained results is the relevance of the chosen learning method. A classification of existing learning methods has been proposed in [3]. Such a systematic view can be used as a basis for the analysis of method relevance.

Another element that may lead to inefficiency is the prominence of missing data. A perspective to the present work is thus to adequately deal with missing data, by making assumptions about the possible and plausible values of missing data. In particular, recent approaches involving imprecise probabilities [32] allow to handle imprecise or missing data with a clear interpretation and method, and seem of interest for the present topic.

Finally, the quality of class definition, for numerical discretized variables, has to be checked to avoid inoperable results.

8.2. Identifying missing knowledge through tree analysis

Beside tree quality evaluation through numerical criteria, as presented in section 6.3, a complementary approach could be based on the tree structure analysis. The objective is to detect stable or unstable structures in the obtained decision trees, in order to identify the parts where variables are inadequate, before submitting them to expert opinion.

Conversely, stable structures could be proposed to the experts as potential knowledge or reasoning patterns to integrate in the ontology.

Such objectives can be reached by using graph operations, since we want to detect similar structures or important differences in bootstrapped trees, or by data mining techniques, as the identification of frequent patterns in bootstrapped trees can be a way to identify stable knowledge.

8.3. Automatic propositions of variable transformations

All along the case study, an important amount of time was spent in the identification of relevant data transformations. A way to reduce expert time consumption would then be to directly propose some data transformations.

This requires to identify:

1. which variables play an important role in the studied phenomenon;
2. whether initial experimental data can be transformed into such variables.

The first task could be achieved by adding to the ontology some concepts or relations of the kind "Variable X in phenomenon Y has a Z influence", where Z is some ordinal variable quantifying variable X influence. Another solution is to use comparative assessments of the kind "Variable X_1 is more influent/more meaningful than variable X_2 on phenomenon Y". Such comparative assessments would provide a partial order on the possible influencing variables.

The second task can be achieved by considering the most influential variables on a given phenomenon, and then to search if there exists a sequence of data transformation that would lead from initial experimental variables to these most influential variables.

For example, in the cooking in water operation, experts could have pinpointed (when initially asked about the process) water pH as an important variable in the process. Consequently, if only the *type of water* is known in the experiment, the method could automatically propose to transform it into the corresponding pH.

8.4. Handling imperfect knowledge

In practice, there will be many cases when a variable has uncertain or missing values. It may also happen that the description granularity will not allow to make exact data transformations. For example, it may only be known for an experiment that the *Vitamin* was a *B-type vitamin* (rather than knowing the exact kind, as in Table 1). In such a case, as not all *B-type vitamins* share the same properties (e.g., some are hydrosoluble, others are not), some properties linked to this experiment will only be known with imprecision, and possible data transformations need to take it into account (in this case, methods of Section 4.1). Similarly, value of *Cooking time* in the last row

of Table 1 or the *kind of water* values (two rows in Table 1) are totally unknown, i.e. fully imprecise.

To allow for such situations in an automated process, there is a need to define proper methods to handle them. This could be done, for example, by using recent uncertainty theories such as evidence theory [29] that permits the use of multi-valued mapping (i.e., functional dependencies) and includes many practical uncertainty models (intervals, probabilities, fuzzy sets) as special cases.

9. Conclusion

Acquiring and formalising new expert knowledge, as well as building reliable models, are two important aspects of artificial intelligence research in experimental sciences. Of particular importance is the confidence that domain experts grant to statistically learnt models. As in other domains (e.g., the semantic web), both data-driven and ontological knowledge can help each other in their respective tasks.

The approach proposed in this paper is a collaborative and iterative method, where expert knowledge and opinion issued from learnt models were integrated to the ontology describing the domain knowledge. This formalisation is then re-used to transform available data and learn new models from these transformed data, these new models being again the source of additional expert opinions, and so on until experts are satisfied with the results. This allows both to enrich the ontological knowledge and to increase expert confidence in the results delivered by learning methods.

The proposed method is applied to a case study in the field of cereal transformation. This case study was undertaken iteratively, in tight collaboration with domain experts. It demonstrates the added value of taking into account ontology-based knowledge, by providing a gain in interpretability and relevance of the results obtained by the learning method. It also aims to extract, by confronting experts with data-driven models, ontological knowledge that may be useful in other applications.

The present work is a first step to meet the difficult challenge of building semi-automated methods, where possible data-transformation derived from the ontological knowledge could be proposed to experts in order to design more significant data-driven models, or alternatively where experts could easily add some knowledge from the results of data-driven methods, without the help of the analyst.

Acknowledgments

The authors wish to thank P. Buche (INRA) for his detailed criticism and comments that have helped to improve the paper presentation.

References

[1] Adomavicius, G., Tuzhilin, A., 2001. Expert-driven validation of rule-based user models in personalization applications. Data Mining and Knowledge Discovery 5, 33–58.

[2] Ben-David, A., Sterling, L., 2006. Generating rules from examples of human multiattribute decision making should be simple. Expert Syst. Appl. 31, 390–396.

[3] Bernstein, A., Provost, F., 2002. An intelligent assistant for the knowledge discovery process, in: In Proc. of the IJCAI-01 Workshop on Wrappers for Performance Enhancement in KDD, Morgan Kaufmann.

[4] Breiman, L., 1996. Bagging predictors. Machine Learning 24, 123–140.

[5] Breiman, L., 1998. Arcing classifiers. The Annals of Statistics 26, 801–824.

[6] Breiman, L., Friedman, J., Olshen, R., Stone, C., . Classification and regression trees. 1984. Wadsworth, Belmont, CA 1.

[7] Caragea, D., Zhang, J., Bao, J., Pathak, J., Honavar, V., 2005. Algorithms and software for collaborative discovery from autonomous, semantically heterogeneous, distributed information sources, in: Jain, S., Simon, H.U., Tomita, E. (Eds.), ALT, Springer. pp. 13–44.

[8] Charalampopoulos, D., Wang, R., Pandiella, S.S., Webb, C., 2002. Application of cereals and cereal components in functional foods: a review. International Journal of Food Microbiology 79, 131 – 141.

[9] Dalbon, G., Grivon, D., Pagnani, M., 1996. Continuous manufacturing process, in: Kruger, J., Matsuo, R., Dick, J. (Eds.), Pasta and noodle technology. AACC, St Paul (MN-USA).

[10] Davesne, E., Casanova, P., Chojnacki, E., Paquet, F., Blanchardon, E., 2011. Optimisation of internal contamination monitoring programme by integration of uncertainties. Radiation Protection Dosimetry 144, 361–366.

[11] Gaines, B.R., Shaw, M.L.G., 1993. Eliciting knowledge and transferring it effectively to a knowledge-based system. IEEE Transactions on Knowledge and Data Engineering 5, 4–14.

[12] Guarino, N., Oberle, D., Staab, S., 2009. Handbook on Ontologies. Springer. chapter What is an Ontology? 2nd edition.

[13] Guillaume, S., Charnomordic, B., 2011. Learning interpretable fuzzy inference systems with fispro. Information Sciences , In Press,.

[14] Hartigan, J.A., Wong, M.A., 1979. Algorithm AS 136: A k-means clustering algorithm. Applied Statistics 28, 100–108.

[15] Ling, T., Kang, B.H., Johns, D.P., Walls, J., Bindoff, I., 2008. Expert-driven knowledge discovery, in: Latifi, S. (Ed.), Proceedings of the fifth international conference on information technology: new generations, pp. 174–178.

[16] Maillot, N., Thonnat, M., 2008. Ontology based complex object recognition. Image and Vision Computing 26, 102–113.

[17] Mansingh, G., Osei-Bryson, K.M., Reichgelt, H., 2011. Using ontologies to facilitate post-processing of association rules by domain experts. Information Sciences 181, 419 – 434.

[18] Miller, G.A., 1956. The magical number seven, plus or minus two: Some limits on our capacity for processing information. Psychological Review 63, 81–97.

[19] Mueangdee, N., Mabille, F., Thomopoulos, R., Abecassis, J., 2006. Virtual grain: a data warehouse for mesh grid representation of cereal grain properties, in: Proceedings of the 9th European Conference on Food Industry and Statistics, Montpellier, France. pp. 291–299.

[20] Musen, M., 1992. Dimensions of knowledge sharing and reuse. Computers and Biomedical Research 25, 435–467.

[21] Noy, N., McGuinness, D., . Ontology Development 101: AGuide to Creating Your First Ontology. Disponible en http://www. Ksl. stanford. edu/people/dim/papers/ontology-tutorial-noy-mcguinessabstract. html [consulta: diciembre de 2005] .

[22] Osei-Bryson, K.M., 2004. Evaluation of decision trees: a multi-criteria approach. Comput. Oper. Res. 31, 1933–1945.

[23] Parekh, V., Gwo, J.P.J., 2004. Mining Domain Specific Texts and Glossaries to Evaluate and Enrich Domain Ontologies, in: International Conference of Information and Knowledge Engineering, The International MultiConference in Computer Science and Computer Engineering, Las Vegas, NV.

[24] Popescu, M., Xu, D., 2009. Data Mining in Biomedicine Using Ontologies. Artech House, Inc., Norwood, MA, USA. 1st edition.

[25] Quinlan, J., 1986. Induction of decision trees. Machine learning 1, 81–106.

[26] Quinlan, J., 1993. C4. 5: programs for machine learning. Morgan Kaufmann.

[27] R Development Core Team, 2009. R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing. Vienna, Austria. ISBN 3-900051-07-0.

[28] Seising, R., 2007. Soft computing and the life science-philosophical remarks, in: IEEE International Conference on Fuzzy Systems, IEEE. pp. 798–803.

[29] Shafer, G., 1976. A mathematical Theory of Evidence. Princeton University Press, New Jersey.

[30] Solomatine, D., Ostfeld, A., 2008. Data-driven modeling: some past experiences and new approaches. Journal of Hydroinformatics 10, 3–22.

[31] Sousa, R.L., Einstein, H.H., 2012. Risk analysis during tunnel construction using bayesian networks: Porto metro case study. Tunnelling and Underground Space Technology 27, 86–100.

[32] Strobl, C., 2008. Statistical Issues in Machine Learning - Towards Reliable Split Selection and Variable Importance Measures. Ph.D. thesis. Ludwig-Maximilians-University Munich, Germany.

[33] Stumme, G., Hotho, A., Berendt, B., 2006. Semantic web mining: State of the art and future directions. J. of Web Semantics 4, 124–143.

[34] Thomopoulos, R., Baget, J., Haemmerle, O., 2007. Conceptual graphs as cooperative formalism to build and validate a domain expertise. Lecture Notes in Computer Science 4604, 112.

[35] Vialette, M., Pinon, A., Leporq, B., Dervin, C., Membré, J.M., 2005. Meta-analysis of food safety information based on a combination of a relational database and a predictive modeling tool. Risk Analysis 25, 75–83.

[36] Villanueva-Rosales, N., Dumontier, M., 2008. Modeling life science knowledge with owl 1.1, in: Proceedings of OWL'08.

[37] Young, L., 2007. Application of Baking Knowledge in Software Systems, in: Technology of Breadmaking - 2nd edition. Springer, US, pp. 207–222.

[38] Zhang, J., Kang, D.K., Silvescu, A., Honavar, V., 2006. Learning accurate and concise naïve bayes classifiers from attribute value taxonomies and data. Knowledge and Information Systems 9, 157–179.

[39] Zhang, J., Silvescu, A., Honavar, V., 2002. Ontology-driven induction of decision trees at multiple levels of abstraction. Lecture Notes in Computer Science , 316–323.