Making Ontology-Based Knowledge and Decision Trees interact: an Approach to enrich knowledge and increase expert confidence in data-driven models

Iyan Johnson[1,3], Joël Abécassis[1], Brigitte Charnomordic[3], Sébastien Destercke[1], and Rallou Thomopoulos[1,2]

[1] IATE Joint Research Unit, UMR1208, CIRAD-INRA-Supagro-Univ. Montpellier II, 2 place P. Viala, F-34060 Montpellier cedex 1
[2] LIRMM, CNRS-Univ. Montpellier II, 161 rue Ada, F-34392 Montpellier cedex 5
[3] INRA, UMR 729 MISTEA, F-34060 Montpellier, France
corresponding author: destercke@supagro.inra.fr

Abstract. When using data-driven models to make simulations and predictions in experimental sciences, it is essential for the domain expert to be confident about the predicted values. Increasing this confidence can be done by using interpretable models, so that the expert can follow the model reasoning pattern, and by integrating expert knowledge to the model itself. New pieces of useful formalised knowledge can then be integrated to an existing corpus while data-driven models are tuned according to the expert advice. In this paper, we propose a generic interactive procedure, relying on an ontology to model qualitative knowledge and on decision trees as a data-driven learning method. A case study based on data issued from multiple scientific papers in the field of cereal transformation illustrates the approach.

1 Introduction

In many domains where extensive mathematical knowledge is not available, sharing expertise and conclusions obtained from data are of great importance for building efficient decision support tools. This is very much the case in Life Sciences [1], owing to the great variability of living organisms and to the difficulty of finding universal deterministic natural laws in biology. Many areas of life science (food processing, cultural practices, transformation processes) rely as much upon expertise and data than upon mathematical models.

For domain experts to use data-driven models (especially in sciences where experiments play a central role), it is necessary for them to be confident in the results. Even if such a confidence can be partially obtained by a numerical validation procedure, an expert (denote by he in the sequel) will always be more confident if he can understand the reasoning followed to make the prediction and if this reasoning coincides with his knowledge of the (natural or industrial) processes and of their interactions. This can be done by using interpretable learning models, such as decision trees, fuzzy rule bases, bayesian networks, ...

Unfortunately, experimental data are seldom collected with a global approach, i.e. with the thought that they are only a part of a more complex system, and are not usually

ideally structured to achieve inductive learning. Learning models from rough experimental data therefore seldom provides models completely meaningful and sensible to the domain expert. Confronting the domain expert to interpretable data-driven models whose descriptive variables do not necessarily exactly coincide with the ones he would have selected has a double benefit. First it can be a means to acquire new items of knowledge from the expert, then it is a good way to design a useful model.

In this paper, we propose an interactive (between AI methods and domain experts) and iterative approach to achieve these two related goals which are usually hard to achieve, i.e. enrich our qualitative knowledge of processes and increase the expert confidence in the data-driven model.

Domain knowledge (coming from expert, literature, ...) is formalised by using an ontology to specify a set of concepts and the relations linking them, which gives a structure that facilitates the interaction with domain experts. Our approach is generic regarding data-driven learning methods, and in the following, we illustrate it with decision trees. 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 domain 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 the conclusions of Miller ([3]) relative to the *magical number* seven.

As far as we know, no interactive approach trying to combine qualitative knowledge (modelled by an ontology) and data-driven learning methods in the field of experimental sciences has been proposed up to now. Indeed, most attempts at such collaborative methods focus on problems where scalability is a main issue, and where method performances can be automatically measured. A few semi-automatic interactive approaches (combining learning and ontology-based knowledge) recently appeared in the literature, in fields where large amounts of data must be treated, such as the Semantic Web ([4],[5]), to deal with multiple ontologies ([6],[7]), or in cases where data are well-structured, such as in image classification ([8]).

The case of inductive learning using ontologies, data and decision trees has been addressed in [9], however it is limited to the specific case of taxonomies[4], whereas in this paper we do not make this restriction. Moreover we consider domain expert knowledge and feedback, while the approach in [9] is more fitted to fully automatic learning (once the ontology is given).

In many cases in Life Sciences, data can be scarce, costly, and not necessarily numerous. Our purpose is to propose a framework to use these data as best as possible. Therefore our primary aim is to not to improve the numerical accuracy of a learnt model (although it is certainly desired), or the fastness with which it detects some features.

We are aware of the challenge to achieve a good balance between the time spent by the domain expert on the learning task and the benefits he can retrieves in terms of model generalization and fiability. Our purpose is to tend towards automated procedures

[4] Ontologies that can be represented as rooted trees in graph theory.

as much as possible, where domain experts, ontology and learning models can interact without the help of AI experts.

The paper is organized as follows: Section 2 provides the necessary background and definitions of ontology and decision trees to understand the paper. Section 3 formally describes the various data processing operations done using the ontology. Section 4 presents the outline of the interactive approach. A case study concerning the impact of agri-food transformation processes on the nutritional quality of wheat-based products is presented in section 5. All along the paper, we illustrate our generic approach by taking examples in the field of expert knowledge, scientific papers and experiments related to cereal product quality.

2 Background

In this section, we briefly recall necessary elements regarding ontology definition and decision trees, which will be used as data-driven inductive learning methods to provide domain expert readable model.

2.1 Ontology definition

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.

Relationship between concepts and variables We consider a data set $\mathbb{D}$ containing K variables and N experiments. Each variable X_k, $k = 1, \ldots, K$, is a concept $c \in \mathcal{C}$ in the ontology Ω. The n^{th} value of the k^{th} variable is denoted $x_{k,n}$.

Concept range A concept c may be associated with a definition domain by the $Range$ function. This definition domain can be: (i) *numeric*, i.e. $Range(c)$ is a closed interval $[min_c, max_c]$; (ii) *'flat' (non hierarchized) symbolic*, i.e. $Range(c)$ is an unordered set of constants, such as a set of scientific papers; (iii) *hierarchized symbolic*, i.e. $Range(c)$ is a set of partially ordered constants, that are themselves concepts belonging to $\mathcal{C}$.

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

1. the *subsumption* or 'kind of' relation denoted by $\preceq$, which defines a partial order over $\mathcal{C}$. Given $c \in \mathcal{C}$, we denote by $\mathcal{C}_c$ the set of sub-concepts of c, such that: $\mathcal{C}_c = \{c' \in \mathcal{C} | c' \preceq c\}$. When c represents a variable with hierarchized symbolic definition domain, we have $Range(c) = \mathcal{C}_c$. For sake of conciseness, we use $\mathcal{C}_c$ in the sequel whenever possible.
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 $\mathcal{C}$. Let $X = \{X_{k_1}, \ldots, X_{k_2}\} \subseteq \mathcal{C}$, $1 \leq k_1 \leq k_2 \leq K$ and $Y = \{Y_{k_3}, \ldots, Y_{k_4}\} \subseteq \mathcal{C}$, $1 \leq k_3 \leq k_4 \leq K$ be two disjoint subsets of concepts. X is said to functionally determine Y if and only if there is a function $DetVal_{FD}$ such that: $DetVal_{FD} : Range(X_{k_1}) \times \ldots \times Range(X_{k_2}) \rightarrow Range(Y_{k_3}) \times \ldots \times Range(Y_{k_4})$. 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 that represent associated properties.

Example 1. $\mathcal{P}(\text{Vitamin}) = \{\text{Thermosensitivity}, \text{Solubility}, \ldots\}$.

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 i^{th} element of $\mathcal{P}(c)$, with $i = 1, \ldots, p_c$. The function $DetVal_{\mathcal{P}}$ will be denoted by $\mathcal{HP}_c$ (for $HasProperty$). It maps a particular value of $Range(c)$ to the particular property values it takes in the ranges of the concepts of $\mathcal{P}(c)$. $\mathcal{HP}_c : Range(c) \rightarrow Range(\mathcal{P}(c)_1) \times \ldots \times Range(\mathcal{P}(c)_{p_c})$. We denote by $\mathcal{HP}_{c\downarrow i} : Range(c) \rightarrow Range(\mathcal{P}(c)_i)$ the restriction of $\mathcal{HP}_c$ to its i^{th} property, that is $\mathcal{HP}_{c\downarrow i} = \mathcal{HP}_c \cap (Range(c) \times Range(\mathcal{P}(c)_i)$.

Example 2. We have $\mathcal{P}(\text{Vitamin})_1 = \text{Thermosensitivity}$.

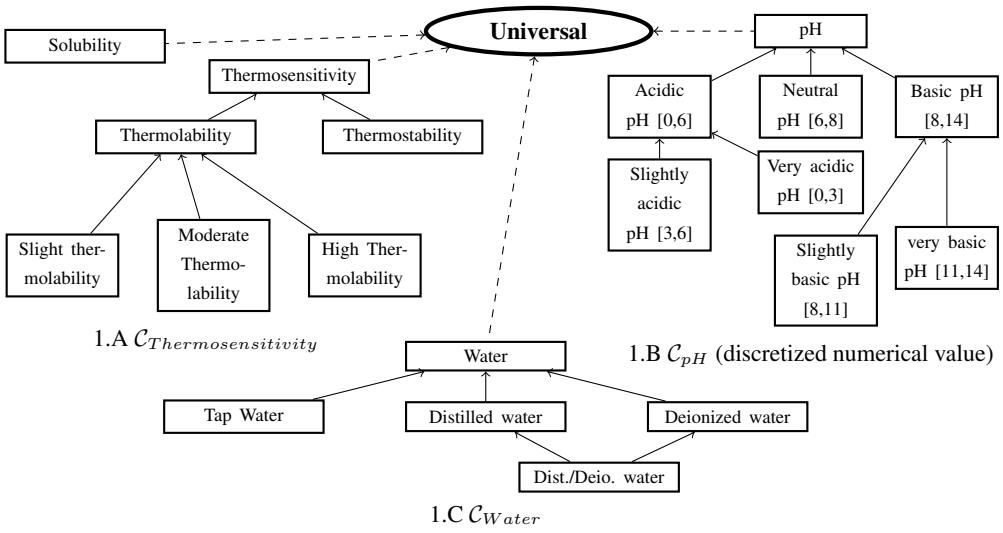

Fig. 1. Some variables and related ontology parts where $A \rightarrow B$ means that A is a *kind of* B

– a *determines* relation $\mathcal{D} : 2^{|\mathcal{C}|} \rightarrow \mathcal{C}$ which specifies a subset of concepts whose values entirely determine the value taken by another concept.

Example 3. $\mathcal{D}(\{$*Pastatype,Cookingtime*$\}) =$ *Cookingtype* models the fact that the *Cooking type* is a function depending on the values of *Pasta type* and of *Cooking time*.

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 $\mathcal{HD}_C : Range(c_1) \times \ldots \times Range(c_{|C|}) \rightarrow Range(\mathcal{D}(C))$, with c_i and $|C|$ being respectively the i^{th} element and the number of elements of C. The function $\mathcal{HD}$ simply gives the values of $\mathcal{D}(C)$, given the values of the determinant variables.

Example 4. $\mathcal{HD}(\{Short, 18min\}) = Overcooking$.

Figure 1 is an example of three categorical variables: ***pH***, ***Water*** and ***Thermosensitivity***, together with the sub-ontologies induced by the order $\preceq$. pH is an example of a continuous variable discretized into a categorical variable. Note that $\mathcal{C}_{Water}$ is *not* a simple taxonomy. We will repeatedly refer to this figure in our forthcoming examples.

2.2 Decision trees

Decision trees are well established learning methods in supervised data mining. They can handle both classification and regression tasks. In multidimensional modeling, they perform well in attribute selection and are often used prior to further statistical modeling. Also note that decision trees algorithms include methods to deal with missing data, meaning that every experiment (or data), even the one with lacking values for some variables, is used in the process. In this paper, due to lack of space, we focus on the C4.5 [10] family of decision trees, and we use them for classification. In the present study, another main interest of decision trees are their interpretability by domain experts, due to their graphical nature.

Algorithm description 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 Y divided into M_Y classes: $(\mathbf{x_n}, \mathbf{y_n}) = (x_{1,n}, x_{2,n} \dots x_{K,n}, y_n)$. An attribute X_k can be continuous or categorical. 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.

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 [11, 10]).

Splitting criteria We denote by $p_m(S)$ the proportion of examples at node S that belong to class m. To select the splitting variable, the C4.5 algorithm uses information theory entropy $I_{Entropy}$ as a selection and splitting criterion, whose value at node S is $I_{Entropy}(S) = -\sum_{m=1}^{M_Y} p_m(S) log_2\, p_m(S)$.

If we denote by M_k the number of modalities of X_k, the improvement gained by splitting the node S into M_k subsets $S_1, S_2 \dots S_{M_k}$ according to X_k, is 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.

3 Data processing using ontologies

When automatically treating data to perform knowledge discovery or classification, some input variables and/or their modalities may be irrelevant to the problem at hand. Indeed, experimental data reported in papers, reports, etc., are usually collected for specific research objectives and may not entirely fit in a global knowledge engineering

approach. In some cases, a particular variable may be decomposed into some properties more significant for the expert. For instance, to appreciate the degradation of vitamin component during the *Cooking in water* operation, it is more interesting to consider the thermosensitivities and reactivities to carbonate of the vitamins rather than the vitamin types themselves. Also, the variable modalities may be too numerous, creating noise. For example, a pH value may be divided into *slightly*, *moderately*, *very* acid and basic, whereas separating between acid and basic pH is sufficient.

This section details various data transformations exploiting both the ontology defined in Sect. 2.1 and domain expert feedbacks to build more significant variables from the original ones. These transformations are performed automatically, according to the used ontological knowledge (note that this ontological knowledge may not be available from the start). Transformed data can then be re-used in the laerning process, thus providing a new model. Feedbacks may be stimulated by a third-party data treatment method, i.e., decision trees in the present paper. Appropriate transformations are selected by an expert evaluation of learning results.

3.1 Replacement of a variable by new ones

This process consists of substituting a variable by some of its (more relevant) properties, which then become new variables. Let X_k be a variable such that $\forall n \in [1; N], \mathcal{P}(X_k) \neq \emptyset$. 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: $\forall n \in [1; N] \quad x_{K+i,n} = \mathcal{HP}_{X_k}(x_{k,n})_{\downarrow i}$, with $\mathcal{HP}_{X_k}(x_{k,n})_{\downarrow i}$ the projection of $\mathcal{HP}_{X_k}(x_{k,n})$ on $Range(\mathcal{P}(X_k)_i)$ and $\mathcal{P}(X_k)_i$ the i^{th} element of $\mathcal{P}(X_k)$. Indeed, a given variable may summarise many aspects of a process, and it is sometimes desirable to decompose this variables into properties to better understand the process and the properties that most influence it (for example, the "year effect" often considered in crop management summarise information related to temperatures, climatic conditions, presence of diseases, ...).

Example 5. Let $X_k = vitamin$ be the (non relevant) variable to be replaced and $\mathcal{P}(vitamin) = \{solubility, thermosensitivity\}$ its properties. We have $X_{K+1} = solubility$ and $X_{K+2} = thermosensitivity$. The new variables are *solubility* and *thermosensitivity*. Now, if for the n^{th} experiment, $x_{k,n} = VitaminA$, the two new values for the n^{th} experiment are $x_{K+1,n} = \mathcal{HP}_{X_k}(x_{k,n})_1 = Liposoluble$ and $x_{K+2,n} = \mathcal{HP}_{X_k}(x_{k,n})_2 = Thermolability$. The initial variable $X_k = Vitamin$ is removed.

3.2 Grouping the modalities of a variable using common properties

In some cases, it may be useful to consider subsets of modalities corresponding to a particular feature rather than the modalities themselves. 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 $n \in [1; N]$:

$z_n = \mathcal{HP}_{X_k}(x_{k,n})_{\downarrow i}, z_n \in Range(\mathcal{P}(X_k)_i)$ and $x'_{k,n} = \mathcal{HP}_{X_k \downarrow i}^{-1}(z_n)$.

The first equation expresses that we first get z_n, the ith property value associated with $x_{k,n}$. The second equation expresses that we then search for all the antecedents of

this value, i.e. all the $x_{k,l}$ ($l \in [1; N]$) whose i^{th} property value is equal to z_n, which includes $x_{k,n}$ but may also include other values.

Example 6. 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 experiments, but that it would be desirable to group them by pH. We have $\mathcal{HP}_{Water}(Tap\ water)_{\downarrow pH} = Basic\ pH$, and $\mathcal{HP}_{Water}(c)_{\downarrow pH} = Neutral\ pH$ for any other $c \in \mathcal{C}_{Water}$. The new variable X'_k thus has the following two modalities: {*Tap Water*} and {*Deionized water, Distilled water, Distilled deionized water*}. Since the second modality is multi-valued, it can then be replaced by a new concept *Ion-poor water* in $\mathcal{C}$, added as a sub-concept of *Water* and a super-concept of *Distilled water* and *Deionized water* (see Fig. 1).

3.3 Merging of variables in order to create a new one

It may be relevant 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 to consider as significant a single variable that is only significant (at least from an expert standpoint) in conjunction with other variables. Let $C = \{X_{k_1}, \ldots, X_{k_{|C|}}\} \in 2^{\mathcal{X}}$ such that $\mathcal{D}(C) \neq \emptyset$. Then we define a new variable: $\mathcal{X}_{K+1} = \mathcal{D}(\{X_{k_1}, \ldots, X_{k_{|C|}}\})$ such that for n $\in [1; N]$: $x_{K+1,n} = \mathcal{HD}_C(\{x_{k_1,n}, \ldots, x_{k_{|C|},n}\})$.

Example 7. When cooking pasta, domain expert differentiate between *Under-cooked*, *Over-cooked*, and *Optimally cooked* products. However, these states depend on the type of pasta and on the cooking-time, which are usually the specified variables in experiments. Therefore, it makes sense to replace *Cooking time* and *Pasta type* by a new variable *Cooking type*. For example, $\mathcal{HD}_C(\{18min, Short\}) =$ *Over-cooked*, replacing in every experiment where *Cooking time*=18 and *Pasta type*=*Short* with *Over-cooked*.

4 Interactive approach: principles and evaluation

In this section, we first present the principles of our interactive approach. Then we detail the way we evaluated the approach and its results.

4.1 Principles

We assume that we start from an initial domain ontology $\Omega_0 = \{\mathcal{C}_0, \mathcal{R}_0\}$, that can be obtained from semi-automated methods [12], domain expert elicitation or that is readily available. We also assume that an initial learning data set $\mathbb{D}_0$ is available, whose variables coincide with the ontology concepts (they may have to be added before starting).

How learning methods and ontology-based knowledge are combined through an interactive and iterative process is summarized in Figure 2. At a step i, It can be summarised as follows:

1. Induce model $\mathbb{M}_i$ from data, using the data set $\mathbb{D}_i$ (starting with $\mathbb{D}_0$);
2. Assess numerical accuracy of $\mathbb{M}_i$ and discuss its significance with domain expert;

3. If domain expert is satisfied, stop the process, if not, elicit from domain expert the transformations to be done on variables, as well as the modalities,properties or functional dependencies used in this transformation. Add newly identified concepts and relations to the ontology Ω_i, obtaining Ω_{i+1} (starting with Ω_0) ;
4. Using Ω_{i+1} and domain expert opinion, transform data (using methods from Sect. 3) to obtain $\mathbb{D}_{i+1}$ from $\mathbb{D}_i$;
5. Set $i = i + 1$ and go back to step 1;

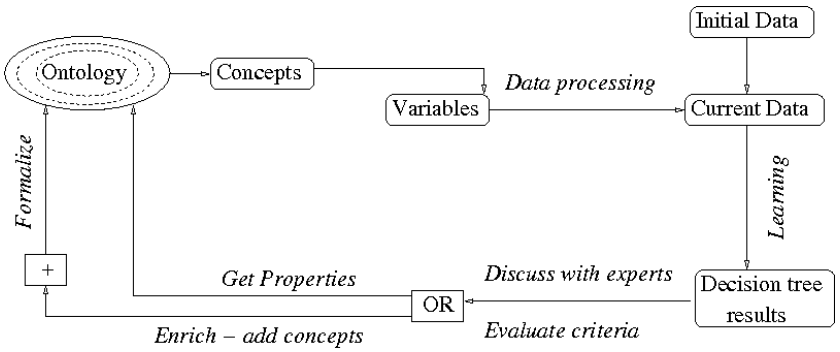

Fig. 2. Collaborative method scheme

4.2 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 numerical evaluation, where the results and stability of the predictive models are measured by numerical indices.
  - The most classical criterion for classification trees is the miclassification rate, $Ec = \frac{MC}{N}$, where MC is the number of misclassified items and N is the data set size, computed with a cross validation procedure or on the whole data set.
  - 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.

5 Case study: application to food quality prediction

Cereal and pasta industry has developed from traditional companies 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 [13].

Previous systems have been proposed in food science, and more specifically in the field of cereal transformation, in order to help prediction [14]. 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.

5.1 Context and description of the case study

For each unit operation of the transformation process, and for each family of product properties, information is given as a data set. The input variables are the operation parameters. The output variable is the operation impact on a property (e.g. the variation of vitamin content). Here, we study the case of the *Cooking in water* unit operation and the *Vitamin content* property. This case concerns 150 experimental data and involves 60 of the ontology concepts. Table 1(a) shows some values of the input variables and of the output variable. The ontology was created using CoGUI (*http://www.lirmm.fr/cogui/*). Data transformation and decision trees were obtained using the R software [15] (use of *R-WEKA* package and about 2000 lines of developed code).

Id	Vitamin	Cooking temp. (C)	Cooking time (min)	Water	Vitamin loss (%)
1	B6	100	13	NA	-52
2	B2	100	12	Tap	-53
3	B1	98	15	Distilled	-47
4	B2	90	10	NA	-18
5	B1	100	NA	Dist./Deio.	-41

Iteration number	MC rate (%)	Complexity
1	44	7.3
2	48	8.4
3	35	7.5
4	35	7.5

Table 1. (a) Part of the training data set (b) Tree evaluation

5.2 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*.

The implementation used for decision trees is the R software with the *R-WEKA* package. The parameters of the algorithm are: minimum number of instances per leaf= 6. All trees are pruned. They are to be interpreted as follows:

1. Each test node is labeled by the splitting variable.
2. for each leaf node, the number of misclassified observations is specified.

Our approach will be conforming to the collaborative approach outlined in section 4. It will be illustrated by four iterations.

Iteration 1: initial state Figure 3 shows the tree trained on the raw data sample ($\mathbb{D}_0$). As mentioned in Section 2.2, the complexity for C4.5 decision trees increases with the number of modalities, which is the case for the *Kind of water* variable. The purpose of our approach is also to reduce that complexity by identifying the relevant underlying properties hidden behind these modalities.

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

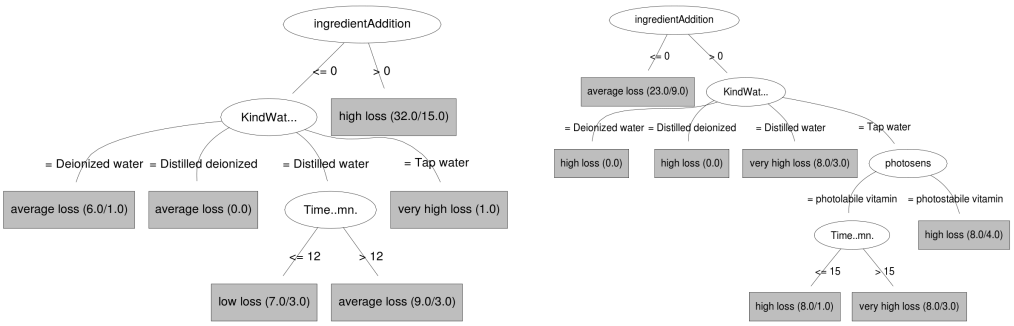

Fig. 3. Decision trees on - (a) raw data - (b) data with vitamin properties

Expert examination of the tree led to the following remarks and adjustments. First, the most discriminant variable is *Ingredient Addition*. Indeed, it corresponds to adding vitamins for compensating a loss during the cooking process. The experts suggested to *enrich the ontology* by characterizing the vitamins by their properties. The following elements were added to the ontology (obtaining Ω_1), and data were transformed to obtain $\mathbb{D}_1$.

$$\mathcal{P}(Vitamin)=\{Solubility,Thermosensitivity,Photosensitivity,...\}$$
$$Range(Photosensitivity)=\{Photolabile,Photostabile\}$$
$$\mathcal{HP}_{Vitamin}(VitaminA)=\{Liposoluble,Thermolabile,Photostabile\}$$

Iteration 2: introducing knowledge on Vitamin properties The model $\mathbb{M}_1$ is a new tree illustrated by Figure 3(b). The *Kind of water* and the *Cooking time* variables are emphasized by this tree. And yet, discussion with experts brought out the fact that the *Cooking time* variable is relevant only if considered with the *Pasta type*. Experts also suggested that water can be better characterized in terms of *pH* and of *Hardness*. In the available experiments the water *pH* and *Hardness* were not measured. However they can be reconstructed from the water types. The following elements were added to Ω_1 to obtain Ω_2 and used to transform $\mathbb{D}_1$ in $\mathbb{D}_2$ (see section 2.1):

$$\mathcal{P}(Water)=\{pH,Hardness\}, \qquad Range(ph)=\{AcidpH,NeutralpH,BasicpH\}$$
$$\mathcal{HP}_{water}(Tapwater)=\{NeutralpH,Hard\}$$
$$\mathcal{D}(\{Pastatype,Cookingtime\})=Cookingtype, \qquad \mathcal{HD}(\{short,18min\})=Overcooking$$

Iteration 3: introducing Cooking type and Water properties Figure 4(a) shows $\mathbb{M}_2$, the tree obtained with the previous modifications. We can see on this tree that the *Hardness*, the newly built variable, is now selected for the second split. The discussion with experts highligted the existence of a link between *Water hardness* and *pH* evolution. The water pH evolution depends both on the *Cooking temperature* and on the *Water hardness*. A new variable will then be created according to a few expert rules not detailed here, obtaining Ω_3 and $\mathbb{D}_3$.

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

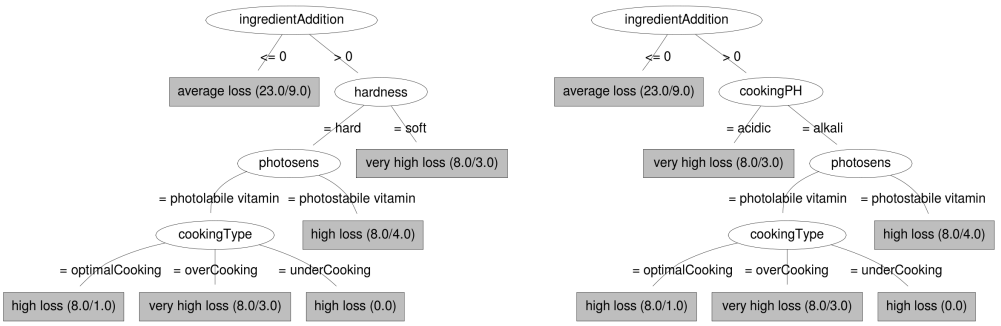

Fig. 4. Decision tree - (a) includingCooking Type and Water Properties - (b) at the final step

Iteration 4: introducing the Cooking pH Figure 4(b) displays the final C4.5 tree (model $\mathbb{M}_3$). When comparing this tree with the initial ones, we can see that relevant variables are now selected by the learning algorithm. In particular, some continuous variables which had been measured through the experiments, such as *Cooking time*, are now replaced by meaningful ones, such as *Cooking type*, which is obtained by conjunction with one more concept introduced in the ontology, i.e. *Pasta type*.

Table 1(b) presents the evolution of the criteria defined in section 4.2. Though the misclassification rate remains high, essentially due to the data scarity, it is better for the last two iterations, while the complexity remains low. Further investigation, through the examination of the confusion matrix, showed that almost all prediction errors are due to the assignment of a label *close* to the *right* one, for instance *High Loss* instead of *Very High Loss*.

6 Conclusion

Formalising and acquiring new expert knowledge, as well as the construction of 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.

In this paper, we proposed a collaborative and iterative approach, where expert knowledge and opinion issued from learnt models was integrated to the ontology describing the domain knowledge. This formalisation is then re-used to transform available data and to learn new models from them, 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 approach 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 expert to 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. There are several perspectives for future work in that direction: to handle missing (or imprecisely defined) items in a more appropriate way (for instance using imprecise probabilities as in recent approaches, see [16]); to consider instances whose possible properties are only partially known; to define new tree evaluation criteria regarding the stability of the selected variables; to automatise the whole process so that AI expert are not needed to perform the analysis.

References

1. Seising, R.: Soft computing and the life science-philosophical remarks. In: IEEE International Conference on Fuzzy Systems, IEEE (July 2007) 798–803
2. Ben-David, A., Sterling, L.: Generating rules from examples of human multiattribute decision making should be simple. Expert Syst. Appl. **31**(2) (2006) 390–396
3. Miller, G.A.: The magical number seven, plus or minus two: Some limits on our capacity for processing information. Psychological Review **63** (1956) 81–97
4. Stumme, G., Hotho, A., Berendt, B.: Semantic web mining: State of the art and future directions. J. of Web Semantics **4** (2006) 124–143
5. Adomavicius, G., Tuzhilin, A.: Expert-driven validation of rule-based user models in personalization applications. Data Mining and Knowledge Discovery **5**(1-2) (2001) 33–58
6. Parekh, V., Gwo, J.P.J.: Mining Domain Specific Texts and Glossaries to Evaluate and Enrich Domain Ontologies. In: International Conference of Information and Knowledge Engineering, Las Vegas, NV, The International MultiConference in Computer Science and Computer Engineering (June 2004)
7. Ling, T., Kang, B.H., Johns, D.P., Walls, J., Bindoff, I.: Expert-driven knowledge discovery. In Latifi, S., ed.: Proceedings of the fifth international conference on information technology: new generations. (2008) 174–178
8. Maillot, N., Thonnat, M.: Ontology based complex object recognition. Image and Vision Computing **26** (2008) 102–113
9. Zhang, J., Silvescu, A., Honavar, V.: Ontology-driven induction of decision trees at multiple levels of abstraction. Lecture Notes in Computer Science (2002) 316–323
10. Quinlan, J.: C4. 5: programs for machine learning. Morgan Kaufmann (1993)
11. Quinlan, J.: Induction of decision trees. Machine learning **1**(1) (1986) 81–106
12. Thomopoulos, R., Baget, J., Haemmerle, O.: Conceptual graphs as cooperative formalism to build and validate a domain expertise. Lecture Notes in Computer Science **4604** (2007) 112
13. Dalbon, G., Grivon, D., Pagnani, M.: Continuous manufacturing process. In Kruger, J., Matsuo, R., Dick, J., eds.: Pasta and noodle technology. AACC, St Paul (MN-USA) (1996)
14. Young, L.: Application of Baking Knowledge in Software Systems. In: Technology of Breadmaking - 2nd edition. Springer, US (2007) 207–222
15. R Development Core Team: R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing, Vienna, Austria. (2009) ISBN 3-900051-07-0.
16. Strobl, C.: Statistical Issues in Machine Learning - Towards Reliable Split Selection and Variable Importance Measures. PhD thesis, Ludwig-Maximilians-University Munich, Germany (2008)