Material classification from imprecise chemical composition : probabilistic vs possibilistic approach

Arnaud Grivet Sébert, Jean-Philippe Poli

Material Classification from Imprecise Chemical Composition : Probabilistic vs Possibilistic Approach

Arnaud Grivet Sébert and Jean-Philippe Poli
CEA, LIST
Data Analysis and System Intelligence Laboratory
Gif-sur-Yvette, France
{arnaud.grivet.sebert,jean-philippe.poli}@cea.fr

2018

Abstract

In this paper we propose a method of explainable material classification from imprecise chemical compositions. The problem of classification from imprecise data is addressed with a fuzzy decision tree whose terms are learned by a clustering algorithm. We deduce fuzzy rules from the tree, which will provide a justification of the result of the classification. Two opposed approaches are compared : the probabilistic approach and the possibilistic approach.

1 Introduction

Customs and ports security is a major issue in Europe. Indeed, many illegal or dangerous substances such as drugs, weapons, explosives pass through customs. Unfortunately, systematic container inspections are impossible in practice because of the cost and time that would be required. The volumes passing daily through the major European ports such as Rotterdam, Antwerp or Hamburg are indeed enormous: for example, 461.2 million tons of goods passed through the port of Rotterdam in 2016.

Our work is part of the C-BORD European project which aims at determining the content of containers using various technologies. One of these technologies uses tagged neutrons, which allows to obtain the chemical composition of a volume of the container. From this chemical composition, we want to determine the materials present in this volume of the container. In order to bring more credibility to the final software used by the customs officers, we will also provide a justification for this classification. Fuzzy rules allow to avoid the "black box" effect since customs officers have access to

a real explanation of the classification made by the software, and close to natural language : "the container may contain drug (confidence degree : x) because the quantity of carbon is high, the quantity of nitrogen is low and the quantity of oxygen is medium" for instance.

The proportions are obtained by different treatments that are beyond our control. Thus, the input data are imprecise and are accompanied by a measure of this inaccuracy. Fuzzy logic thus seems appropriate for the exploitation of such data.

Given the time required and the authorizations needed to use a neutron generator, we will have a small learning data set, even if all classes of relevant materials will obviously be represented. The idea is therefore to use fuzzy decision trees [4], which have been applied successfully on various classification problems [1, 13]. The scarcity of training data and the intrinsic inaccuracy due to the preprocessing preclude conventional statistical learning approaches such as neural networks, SVM, etc. which also do not provide an explanation to users.

In this article, we adapt to imprecise data the classic two-step workflow consisting in using clustering methods to create relevant terms from data and then in building a decision tree to get either a probabilistic or a possibilistic classifier. We propose a comparison of these two approaches, as some authors did for other applied problems [8], [12].

The paper is structured as follows: section 2 describes the context of the application that motivates this work. Then, section 3 describes the method to induce rules from imprecise data. Section 4 presents the results of the different experiments we conducted. Finally, section 5 draws the conclusions of this paper.

2 Application context

2.1 Neutron inspection for container digging

The H2020 project C-BORD (effective Container inspection at BORDer control points) aims at securing borders by exploiting different technologies: e-noses, X-rays, photo-fission and tagged neutrons. The goal is to detect dangerous (explosives, nuclear materials, ...) or illicit (drugs, contraband, ...) substances in containers.

In this paper, we focus on the use of tagged neutron and material classification. As shown in fig. 1, we use a device which produces a neutron beam to focus on a certain voxel (i.e. a volume) of the container. The neutrons interact with the nuclei of the atoms contained in the voxel, producing new particles that can be detected by the matrix sensors which are positioned on the side of the container. These particles are thus characteristic of the atoms encountered in the examined voxel.

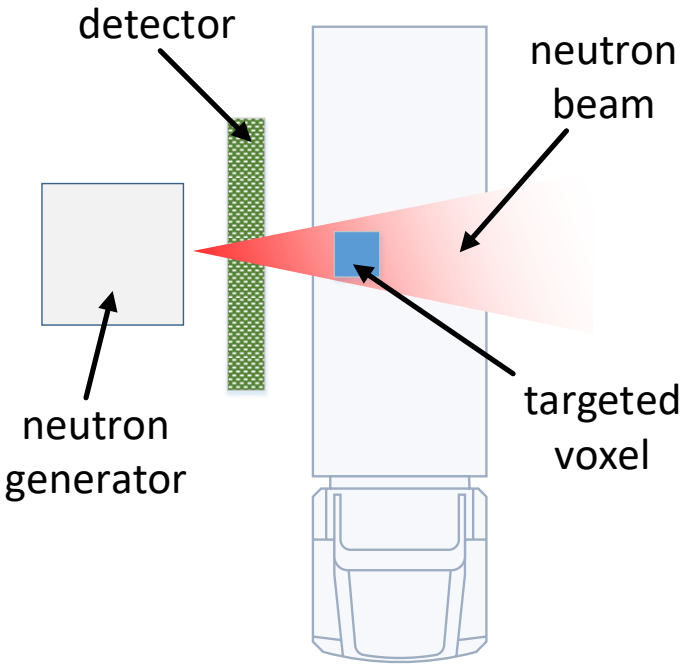

Figure 1: Neutron inspection principle

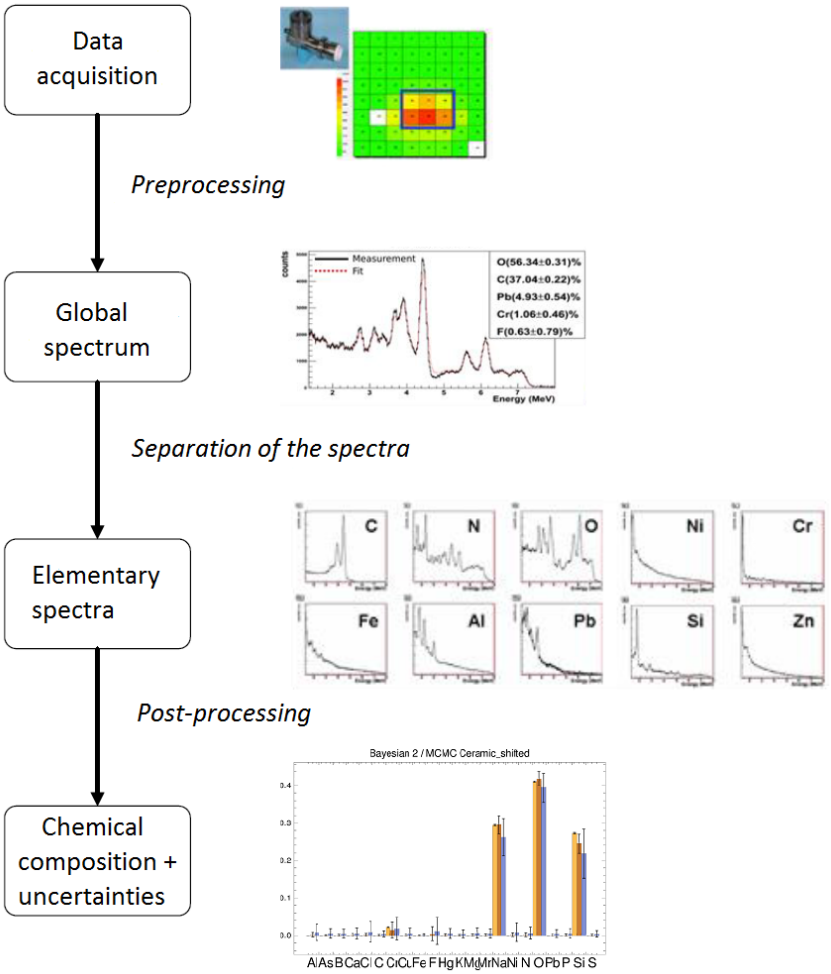

Figure 2: Workflow to get chemical composition assessment from raw data acquisition

The processing of the raw data is not the topic of this article but we quickly describe the principle in fig. 2. After different pretreatments, a global spectrum is obtained. In comparison with the characteristic spectra of each of the studied atoms, this spectrum is decomposed into individual spectra by a Bayesian process which makes it possible to deduce the chemical composition of the voxel, expressed in percentages. This process is based on simulation and we can easily get the mean and the standard deviation for each proportion in order to characterize the inaccuracy of the reconstruction. Fig. 3 shows the result of these treatments for an exposure to ceramics, and in which the inaccuracy is represented by "box plots".

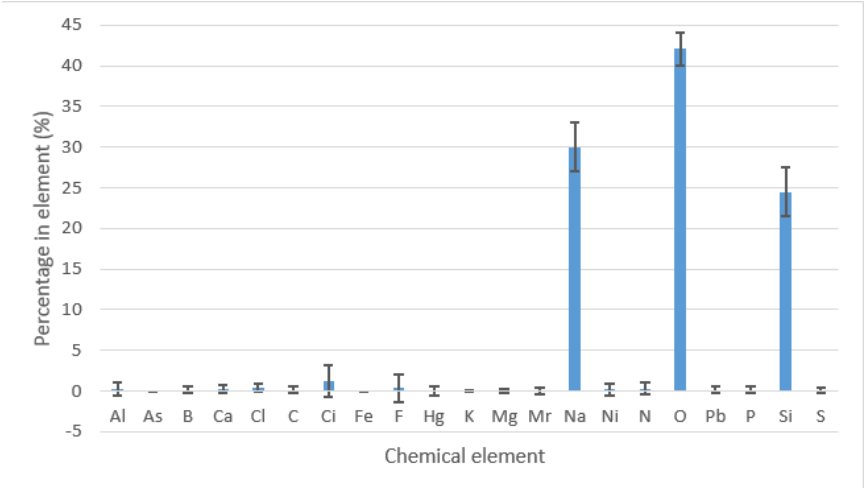

Figure 3: Chemical composition obtained from ceramics simulation

Our work consists in exploiting this information in order to recognize the materials contained in the voxel.

2.2 Previous work

We barely found some articles concerning the use of tagged neutrons to characterize materials. This can be explained by several difficulties.

First, data are scarce because few acquisition campaigns can be conducted. That also explains why few papers address the recognition of materials from their chemical composition. Then, the inaccuracy of the chemical composition makes the task difficult for conventional statistical models. Finally, a not insignificant difficulty is that the device is insensitive to hydrogen atoms, and thus some materials cannot be distinguished.

For these reasons, previous works proposed visual analytics methods to represent the content of the voxels. In [2], the authors proposed a Voronoi diagram to highlight the proximity, in terms of chemical composition, of the current voxel with voxels previously inspected and whose content is known

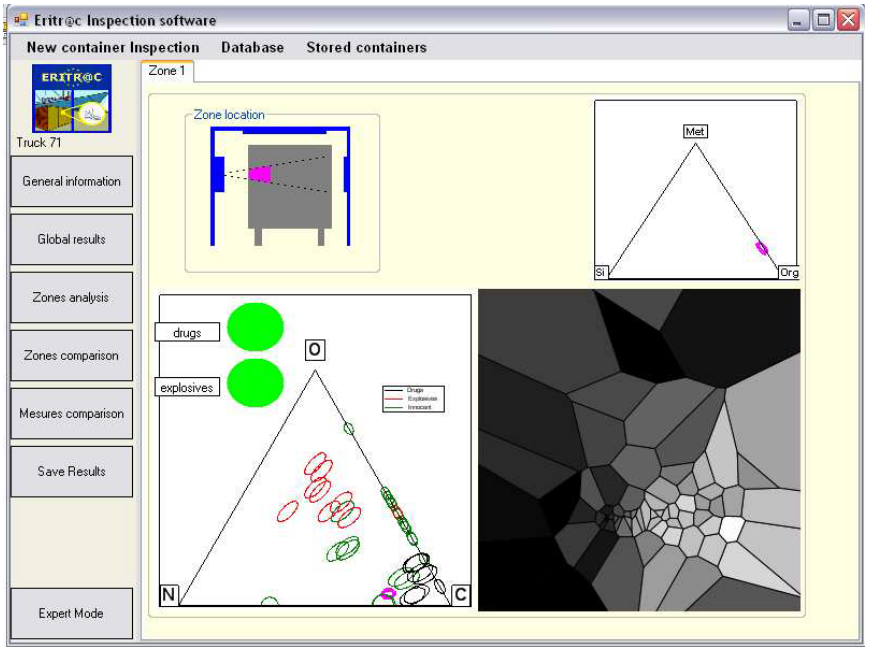

Figure 4: Screenshots of the different visualizations introduced in [2]

(see figure 4). Thus, it is not a question of recognizing the materials inside the container but of displaying visually close and known containers in order to deduce the contents.

This approach has the advantage of not requiring learning or parametrization since it relies on the manual selection of a neighborhood. In figure 4, we can also see two classical representations that have been used in conjunction with this method. These are projections of the current voxel onto two triangles:

- the "materials triangle" indicates the proximity of the voxel with metals, ceramics and organic materials;
- the "alert triangle" presents the ratios between carbon, nitrogen and oxygen and is useful to recognize organic materials (even if the hydrogen is not detected).

The latter triangle helps to distinguish between drugs and explosives. The main drawback of the visual analytics approach is that the operator must be able to interpret the different representations himself.

In practice, the mastering of these representations, particularly the Voronoi graph, can be difficult for operators who are not familiar with these visualization techniques. As part of the C-BORD project, we want to go further and propose a list of materials, and an explanation of the decision. It is

to overcome these different difficulties that we want to use a fuzzy expert system.

3 Fuzzy material classification

To solve the classification problem we build a fuzzy decision tree [4] from which fuzzy rules are extracted to provide the end-user with a justification of the result. Each node of the tree is split into child nodes by an attribute, i.e. a chemical element. The child nodes correspond to fuzzy terms induced by a strong partition of the domain of the chemical ratios, namely [0, 100] since they are percentages. If no pruning is performed, the tree has a maximal depth equal to the number of chemical elements used as attributes for the classification. An example of a tree is shown in fig. 5.

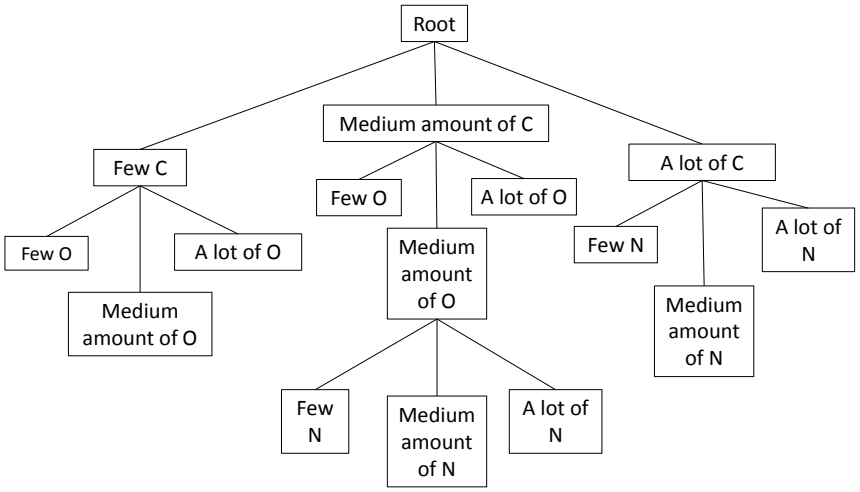

Figure 5: Fuzzy tree for material classification

The imprecise inputs are modeled as either probability or possibility distributions. In the probabilistic case, the distributions are Gaussian distributions normalized so that their integral on the domain of the input data, namely $[0, 100]$, equals 1. In the possibilistic case, we tested triangular and Gaussian distributions normalized so that their maximum value is 1.

We chose to learn fuzzy terms and to build the decision tree in two separate phases.

3.1 Fuzzy terms learning

Among the various methods we tested, the best methods to learn linguistic terms turned out to be clustering techniques. We describe hereafter the adaptations of two clustering algorithms.

3.1.1 Measures of dissimilarity

We modified these algorithms using a dissimilarity adapted to imprecise data. We consider the following dissimilarity for the probabilistic case :

$$d(x, y) = 1 - \int_a^b min(f_x(t), f_y(t))dt \quad (1)$$

where f_x and f_y are the densities of the distributions representing the data x and y, a and b are the bounds of the definition domain of the data, 0 and 100 in our case. This dissimilarity is actually a distance for continuous probability distributions but we will not prove it here for reasons of space.

In the possibilistic case, the integral is replaced by the *max*.

Fig. 6 shows the area corresponding to the associated similarity $\int_a^b min(f_x(t), f_y(t))dt$ between two Gaussian probability distributions, drawn in red.

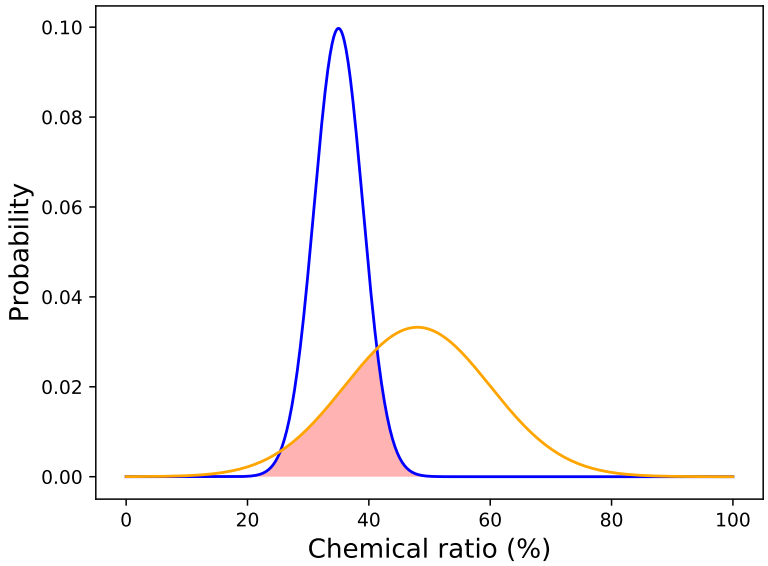

Figure 6: Similarity between two imprecise data - probabilistic modeling

We tested several algorithms and decided to study more thoroughly k-medoids and affinity propagation because they gave better results than k-means and DBSCAN in particular. K-means implies to compute the distance between a data input and a centroid, which is a single point randomly chosen and hence cannot be modeled as an imprecise datum. Unable to use the previously defined dissimilarity, we used asymmetric dissimilarities (in the probabilistic case for example, the integral of the distance between the centroid and the moving point of the imprecise distribution). This incompatibility between the imprecise inputs and the single-point centroids might

explain the poorer results. Using DBSCAN, we often obtained a very small number of clusters because the data distribution is quite continuous along the domain $[0, 100]$. Since DBSCAN works by analyzing the neighborhood of each data point, the absence of clear gaps in the data distribution prevents DBSCAN from splitting the data set.

3.1.2 K-medoids

We adapted k-medoids algorithm [11] to cluster the imprecise data by substituting the traditional Euclidean distance by the dissimilarity defined above. This well-known algorithm takes the number n of clusters as a parameter. It randomly chooses n objects from the dataset to be the initial centers of the clusters and the other objects are assigned to the closest center. In each cluster, the object minimizing the sum of the distances to the other objects is set as the new center. Then the algorithm alternatively updates the sets belonging to each cluster and the centers until convergence. The output is the best result over several initial random configurations.

3.1.3 Affinity propagation

We also tested the affinity propagation algorithm [6]. In this algorithm, every data point is a potential cluster exemplar. A distribution of similarity for each pair of points is given as an input. Another parameter, a real called preference, is used to control the number of clusters.

The data points send "messages" to each other. A data point i sends to a data point k the responsibility $r(i, k)$ which represents how much i is likely to be in a cluster whose exemplar is k. k sends to i the availability $a(i, k)$ which represents how much k is available for i to be in its cluster. The exemplars are the points i such that $r(i, i) + a(i, i) > 0$ and a data point is assigned to the closest exemplar according to the given similarity. The responsibilities depend on the similarity distribution. Besides, the responsibilities are function of the availabilities and conversely. The algorithm successively updates the responsibilities and availabilities until the clusters converge. We implemented this algorithm using the similarity associated with the dissimilarity defined in 1.

3.1.4 From clusters to membership functions

From each cluster we deduce a triangular fuzzy term whose top, of value 1, is set at the center of the cluster. The slopes are induced by the other terms' tops under the strong partition constraint. The extreme sets are slightly different : they are right-angled trapezoids whose flat part ends at the bound of the domain. We can see an example of a five-set partition in fig. 7. More complex term shapes (trapezoids, pentagons) reduce the performance and even the accuracy in some cases.

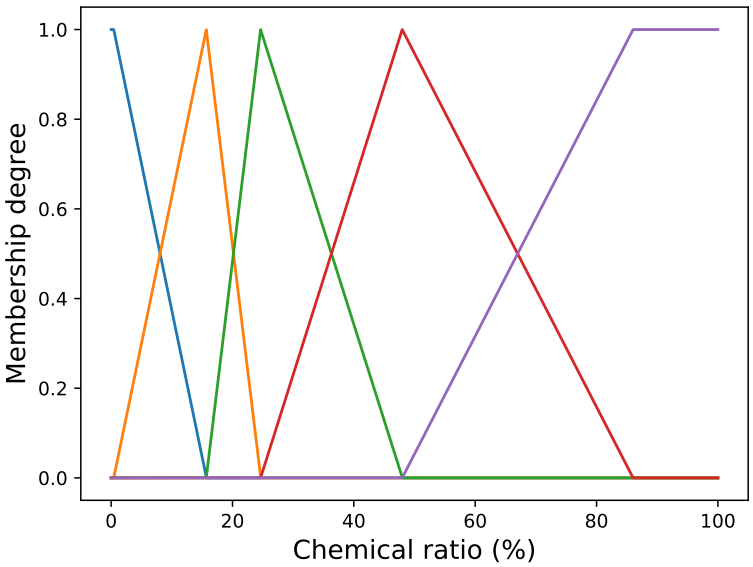

Figure 7: Five learned fuzzy sets

3.2 Construction of the fuzzy decision tree

3.2.1 Probabilistic approach

To build the fuzzy decision tree, we also need to adapt to imprecise data the definition of the membership degrees to the linguistic terms. In [5] and [15], the authors proposed integration techniques to deal with relations between uncertain data and crisp terms. We generalized these techniques to imprecise data and fuzzy terms.

Given f the density of the probability distribution representing the imprecision associated with the proportion of an element e in an example x, and $\mu_v(t)$ the membership degree of the value t to the fuzzy term v, the membership degree of the imprecise example x to the fuzzy term v is :

$$\tilde{\mu}_v(x) = \int_0^{100} f(t)\mu_v(t)dt \tag{2}$$

This definition is illustrated in fig. 8 where the imprecision distribution is dilated 10 times for a better visibility.

The membership degree of an imprecise input x to a node n is defined as the product of the membership degrees to the linguistic terms associated with the ascendants of n, including n :

$$d_n(x) = \prod_{n' \in A(n)} \tilde{\mu}_{v_{n'}}(x) \tag{3}$$

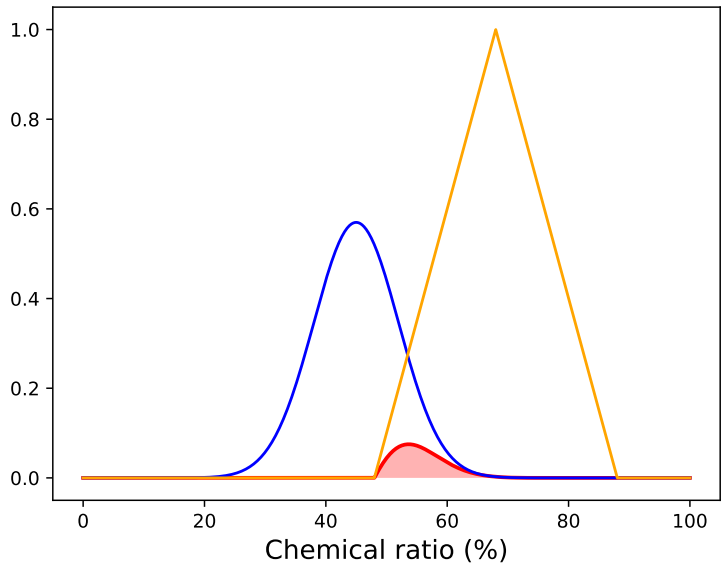

Figure 8: Membership degree of an imprecise datum to a linguistic term - probabilistic modeling

where, for all node n' , $v_{n'}$ is the linguistic term associated with n', and $A(n)$ is the set of the ascendants of n, including n. By convention, the degree of any input to the root is equal to 1 [1].

Using this definition, we can compute the fuzzy entropy, inspired from [14], which is the criterion to select which attribute will be used to split a node. We consider the "fuzzy frequency" of a class c in a node n :

$$fr_{c/n} = \frac{\sum_{x \in Tr \cap c} d_n(x)}{\sum_{x \in Tr} d_n(x)}$$

where Tr is the training set. We also define the "membership frequency" of the examples to the node n :

$$fr_n = \frac{\sum_{x \in Tr} d_n(x)}{\sum_{m \in S} \sum_{x \in Tr} d_m(x)}$$

where S is the set of the sibling nodes of n, including n.

The fuzzy entropy, according to an element e, used at a node N whose set of children is noted $Chil(N)$, these nodes corresponding to each fuzzy term relative to e, is then defined as :

$$E(N) = - \sum_{n \in Chil(N)} fr_n \sum_{c \in C} fr_{c/n} \times log_2(fr_{c/n})$$

[1] Any other non negative real d_0 would lead to the same results since it would only multiply the membership degrees of all examples to all nodes by d_0.

where C is the set of the classes of the problem.

Each node n of the tree is split in child nodes regarding the linguistic terms corresponding to the chemical element which minimizes the entropy, thus maximizing the entropy gain $G(n) = E(N) - E(n)$, N being the father of n. This node splitting is processed until one of the following stopping criteria is reached :

- All the attributes have been used splitting the ascendants of the current node.
- The sum of the membership degrees to the current node is less than a threshold fixed in advance.
- The entropy gain is less than another threshold fixed in advance.

3.2.2 Possibilistic approach

The definition 2 is replaced by :

$$\tilde{\mu}_v(x) = \underset{0 \leqslant t \leqslant 100}{max}(min(f(t), \mu_v(t))) \tag{4}$$

where the possibilistic norms *max* and *min* are used. Fig. 9 illustrates this definition in the case of a Gaussian modeling of the input datum. Note that this definition coincides with the possibilistic similarity defined in 3.1.1, applied to a datum and a linguistic term.

The membership degree of an imprecise input x to a node n is defined as the minimum of the membership degrees to the linguistic terms associated with the ascendants of n, including n :

$$d_n(x) = \underset{N \in A(n)}{min} \tilde{\mu}_{v_N}(x) \tag{5}$$

where, for all node N , v_N is the linguistic term associated with N, and $A(n)$ is the set of the ascendants of n, including n.

We replaced the entropy by another splitting criterion, the non-specificity, proposed in [7]. We consider an ordered discrete possibility distribution $(\pi_i)_{1 \leqslant i \leqslant m}$ such that $\forall i \in [|1, m-1|], \pi_i \geqslant \pi_{i+1}$ and $\pi_1 = 1$. We also note $\pi_{m+1} = 0$. The measure of non-specificity, called U-uncertainty is defined as follows :

$$U(\pi) = \sum_{i=2}^{m} (\pi_i - \pi_{i+1}) log_2(i)$$

In our case, π is the possibility distribution of the classes in the current node : for every class c in C we note

$$\pi_{c/n} = \frac{\sum_{x \in T \cap c} d_n(x)}{max_{c' \in C} \sum_{x \in T \cap c'} d_n(x)}$$

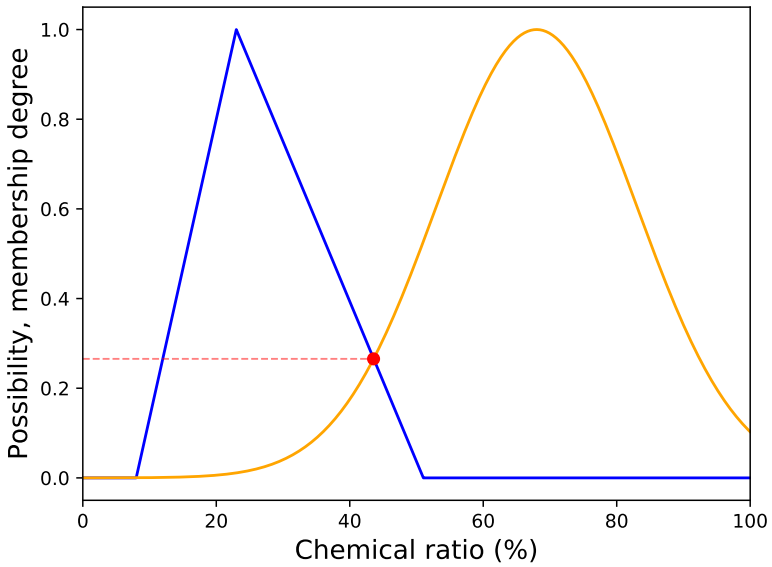

Figure 9: Membership degree of an imprecise datum to a linguistic term - possibilistic modeling

We then sort $(\pi_{c/n})_{c \in C}$ in the decreasing order and compute the non-specificity using the formula above.

The stopping criteria are the same as in the probabilistic framework, the entropy gain being replaced by the non-specificity gain $NSG(n) = U(N) - U(n)$, where N is the father of the node n.

To avoid the bias due to the varying number of linguistic terms of the attributes, we decided, inspiring from [10], to normalize the non-specificity gain by the $SplitInfo(n) = -\sum_{n' \in Chil(n)} fr_{n'} \times log_2(fr_{n'})$ which is actually the potential fuzzy entropy generated by splitting the node n, ignoring the classes of the samples.

3.3 Rule generation

While the tree is being built, the fuzzy frequencies $fr_{c/l}$ of every class c in every leaf l are computed and stored in memory. We then obtain, for each leaf, $|C|$ rules - $|C|$ being the number of classes of the problem -, weighted by the associated fuzzy frequency which corresponds to the "certainty factor", as defined in [3]. For instance, in fig. 5, the leaf on the extreme right will induce the rules "if there is a lot of carbon and a lot of nitrogen, then the object is in class c", for all c in C.

3.4 Recognition of input test data

As both training and test data are imprecise, for every new test sample x, we compute the membership degree of x to every leaf of the tree using the very same definition as in the training phase (3 or 5). Given a class c, this membership degree is then conjugated by the appropriate T-norm with the certainty factor of each rule whose consequent is c and the results are summed up using the weighted voting method [9] to obtain the confidence degree of the proposition $x \in c$:

$$conf(x \in c) = \sum_{l \in L} T(fr_{c/l}, d_l(x))$$

where L is the set of the leaves of the tree and T is the probabilistic (resp. possibilistic) T-norm $\times$ (resp. min). We finally deduce the distribution of the classes for the input x normalizing the confidence degrees. In the probability case, we then have :

$$\forall c \in C, p_x(c) = \frac{conf(x \in c)}{\sum_{c' \in C} conf(x \in c')} \tag{6}$$

and in the possibilitic case :

$$\forall c \in C, \pi_x(c) = \frac{conf(x \in c)}{max_{c' \in C} conf(x \in c')} \tag{7}$$

4 Experiments

4.1 Data simulation

We decided to test several methods on simulated data while waiting for real data. To do so, we refer to the molecular formulæ of the different materials - the classes of the problem. For each material m and each element e, we deduce from the formula of m the stoichiometric percentage of e in m. This percentage p is used as a reference around which we randomly and in an uniform way pick a value v in the interval $[p - p \times \frac{DI}{100}, p + p \times \frac{DI}{100}]$. The span of this interval is proportional to p and to a parameter DI we call "degree of imprecision", used to control the level of imprecision of the generated data. We then generate a random standard deviation σ via a normal distribution of mean $\frac{p}{2} \times \frac{DI}{100}$ and standard deviation $\frac{p}{4} \times \frac{DI}{100}$. The tuple (v, σ) constitutes an imprecise datum.

We ran the experiments with data generated from the molecular formulæ of seventeen explosives and nine drugs, with a degree of imprecision of 15. Since the real data will be few because of the financial and temporal costs of the experiments, we decided to use training data sets with only ten samples per class, on which we performed five-fold cross validation. We finally averaged the results on five data sets.

In the probabilistic approach, we represented our data as Gaussian distributions. In the possibilistic case, we tested our method with both triangular and Gaussian distributions ; the following results are obtained with Gaussian distributions, with which the performance turned out to be much better.

4.2 Fuzzy partitioning

One of the parameters that affect the most the accuracy of our classification method is the number of linguistic fuzzy terms used to define the partition of each chemical ratio domain.

4.2.1 Comparison of two clustering algorithms

We tested our method, using the adapted k-medoids algorithm, over all the combinations of numbers of fuzzy terms from 2 to 14 fuzzy terms, for the elements carbon (C), nitrogen (N) and oxygen (O). Since the other elements are far less discriminative for the classification of the materials we focus on, the number of fuzzy terms for these elements is set to 5 for the moment to run quicker tests. The best combination of numbers of fuzzy terms in the probabilistic framework is 14 for carbon, 14 for nitrogen and 13 for oxygen. In the possibilistic framework, it is respectively 5, 13 and 14 terms.

We then tested affinity propagation algorithm. In the probabilistic case, in the light of k-medoids test, we set the same preference to the clustering for the partitions of C, N and O ratios. It gave slightly worse results (88%) than the best obtained with k-medoids (90%). In the possibilistic case, we also obtained slightly worse results than k-medoids ones (79% against 81%).

We can conclude that k-medoids algorithm is preferable for this problem since it gives results at least as good as affinity propagation but the parameter to be optimized - the number of fuzzy terms - is discrete whereas affinity propagation preference is continuous.

To improve the robustness of our method, it would have been interesting to split the training set in two parts in order to learn the fuzzy terms from different data from the ones used to build the tree. But since we have very few data, we decided to learn both the fuzzy terms and the tree from the whole training set.

4.2.2 Probability vs possibility

Fig. 10 shows the dependence of the correct classification rate on the number of fuzzy terms for both probability and possibility approaches. Here, the results are averaged on five data sets. We can clearly see that the probability framework (in dark green) gives better results than the possibilistic one (in light blue). We merged the numbers of terms for N and O in one axis using a technique thoroughly described below, in the explanation of fig. 11.

To avoid bias due to the differences in the sizes of the trees, we ran the test setting the entropy and non-specificity gain thresholds so that the number of leaves of the trees are similar in the two approaches : the average number of leaves is close to 500 for a threshold of 0.1 in the probabilistic case, and 0.15 in the possibilistic case.

We can then conclude that the probabilistic approach is better than the possibilistic approach for our problem of material classification from imprecise chemical data. One reason might be that the possibilistic T-conorm, *max*, does not take into account the whole distribution which represents the input datum, and only considers a single value. On the contrary, the integral used in the probabilistic model grasps the sense of the whole distribution by computing the average of all the contributions.

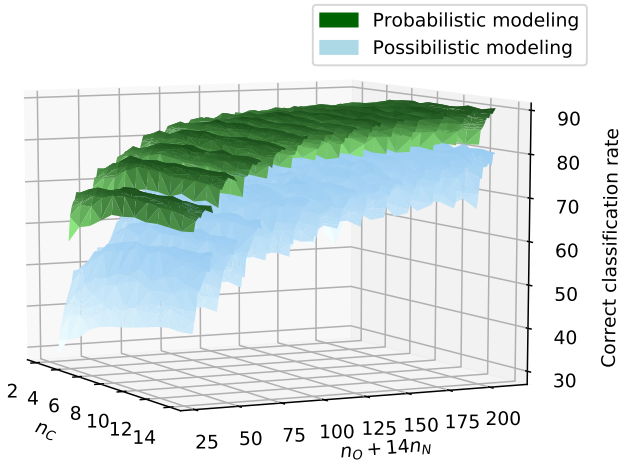

Figure 10: Correct classification rate against number of fuzzy terms for both approaches

Fig. 11 illustrates the grid-based tests we performed with the probabilistic framework for numbers of terms varying from 2 to 14 for C, N and O, and set to 5 for the other elements. The graph displays the accuracy as a function of the numbers of terms for C, N and O. In order to represent this dependence in a 3D graph, we merged the N and O numbers of terms on one axis. Every test is represented as a point of coordinates $(n_C, n_O + 14n_N, r)$, where n_C, n_N and n_O are the numbers of terms for C, N and O respectively, and r is the correct classification rate. The points corresponding to a same value of n_N are linked into a surface. Multiplying n_N by 13 would have been

sufficient to guarantee the injectivity of the representation but we multiplied by 14 to let a gap between two consecutive surfaces for the sake of visibility. We can see that higher numbers of terms for N and O increase the accuracy but that a relatively low number of terms for C is enough to obtain good correct classification rates. The best combination turned out to be 5 terms for C, 14 for N and 12 for O. In the following, we will present further results for the probabilistic framework, with this optimal combination.

4.3 Results analysis

In the probabilistic approach, our treatment of the imprecision of the data does improve the accuracy compared to the same method without taking into account this imprecision, i.e. considering the data as discrete values - the means of the imprecision distributions. Fig. 12 represents the difference of correct classification rate with and without imprecision treatment. The gain of the imprecision treatment increases with the degree of imprecision of the data (see the definition in 4.1).

Fig. 13 is a confusion matrix which summarizes the results of the probabilistic method on the 26-class drug/explosive problem with the optimal combination of 14 fuzzy terms for carbon, 14 for nitrogen and 13 for oxygen. We can see that the algorithm classes quite correctly both the explosives (seventeen first classes from the top in ordinate) and the drugs (six last classes). The correct classification rate is 88.08%.

We also tested the method with the same parameters on a more complex problem with 38 classes of drugs, explosives and benign materials. The accuracy obtained is a bit lower : 79% of correct classification (see confusion matrix in fig. 14).

5 Conclusion

We presented a workflow for explainable material recognition from chemical compositions, based on linguistic variables learning and fuzzy decision tree induction. We implemented both probabilistic and possibilistic approaches. The probabilistic framework is actually an hybrid probabilistic/fuzzy approach since it takes a Gaussian probability distribution as an input and apply a fuzzy inference process to it. The comparison between the two approaches showed that the probabilistic one provides better results.

The method takes into account the imprecision of the data in both training and evaluation phases. In the probabilistic case, it outperforms the method without imprecision treatment and this gain increases with the degree of imprecision of the input data. We obtained correct classification rates up to 85%.

Several refinements may allow us to enhance our method's accuracy in further work. We might improve the linguistic terms learning using other

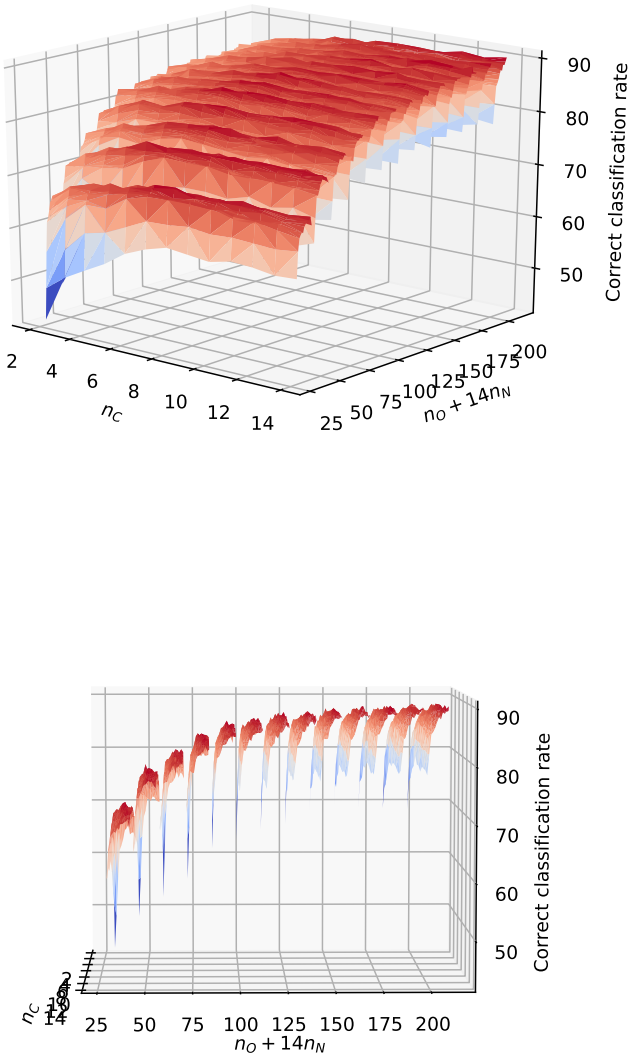

Figure 11: Correct classification rate against number of fuzzy terms - two views of the same graph

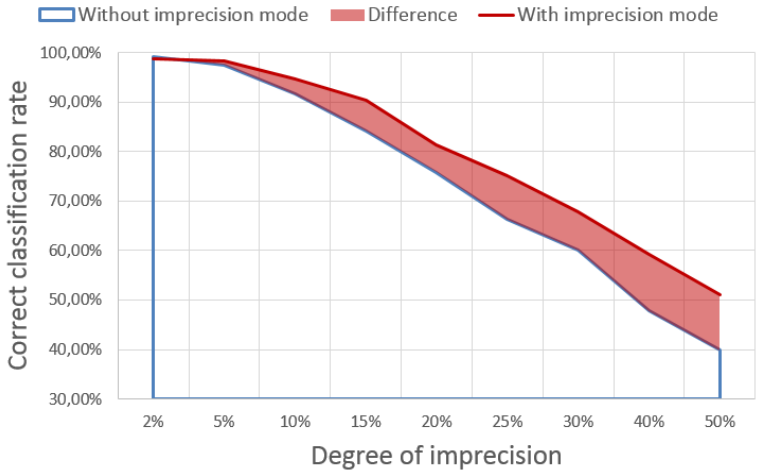

Figure 12: Gain obtained when taking the imprecision of the data into account

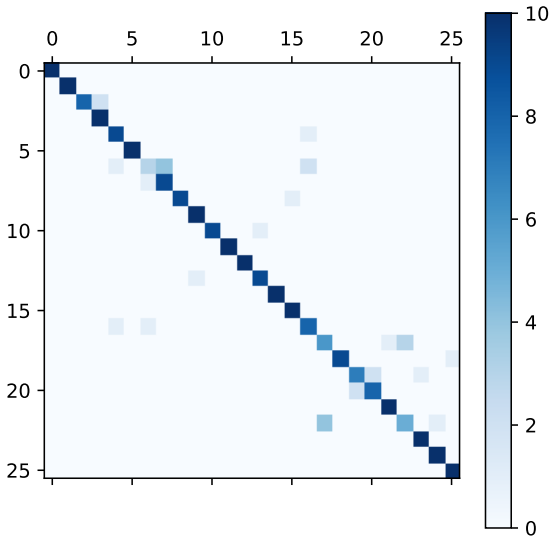

Figure 13: Confusion matrix for drugs and explosives

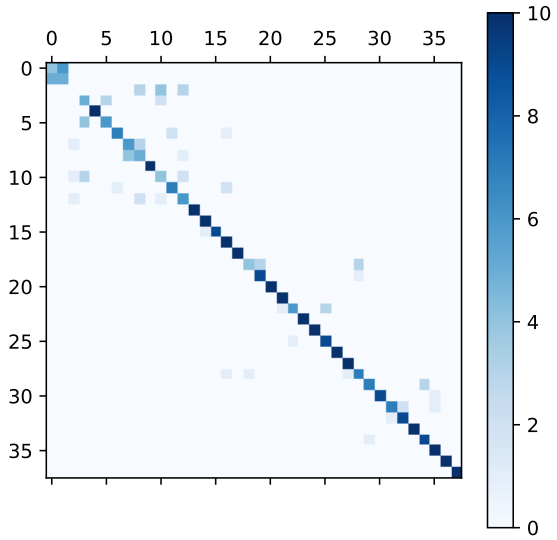

Figure 14: Confusion matrix for drugs, explosives and benigns

dissimilarities, and taking the samples' labels into account. We shall also experiment more sophisticated aggregation methods than the mere addition of the results associated to each leaf. Some post-pruning techniques could be used along with the pre-pruning criteria to avoid trees with too many rules. Simpler rules, corresponding to the intern nodes of the tree, might also be useful in the classification.

We will soon be able to evaluate our method on real data sets with high imprecision. Data acquisition on pure elements, chemical products, drug and explosive simulants are being performed. This will be a first step since we will eventually have to address the problem of recognition of mixtures and materials into matrices, that are very frequent in real life container voxels.

Acknowledgment

This project has received funding from the European Union's Horizon 2020 research and innovation program under grant agreement No 653323. We warmly thank S. Moretto, C. Fontana, F. Pino, A. Sardet, C. Carasco, B. Pérot and V. Picaud for their contributions before our work, for their availability and their expertise.

References

[1] T. Akiyama and H. Inokuchi, "Application of fuzzy decision tree to analyze the attitude of citizens for wellness city development," in *2016 17th International Symposium on Advanced Intelligent Systems*, 2016.

[2] M. Aupetit, L. Allano, I. Espagnon, and G. Sannié, "Visual analytics to check marine containers in the eritr@c project," in *Proceedings of International Symposium on visual analytics science and technology*, 2010, pp. 3 – 4.

[3] M. Bounhas, H. Prade, M. Serrurier, and K. Mellouli, "A possibilistic rule-based classifier," in *IPMU 2012 : Advances on computational intelligence*, 2012.

[4] R. L. P. Chang and T. Pavlidis, "Fuzzy decision tree algorithms," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 7, no. 1, pp. 28 – 35, 1977.

[5] W. Duch, "Uncertainty of data, fuzzy membership functions, and multilayer perceptrons," *IEEE Transactions on Neural Networks*, vol. 16, no. 1, pp. 10 – 16, 2005.

[6] B. J. Frey and D. Dueck, "Clustering by passing messages between data points," *Science*, 2007.

[7] M. Higashi and G. J. Klir, "Measures of uncertainty and information based on possibility distributions," *International journal of general systems*, 1982.

[8] A. G. Huizing, A. Theil, and F. C. A. Groen, "A comparison of a possibilistic and a probabilistic classifier in a multitarget tracking environment," in *Proccedings of IEEE 1997*, 1997.

[9] H. Ishibuchi, T. Nakasima, and T. Morisawa, *Fuzzy sets and Systems*. Elsevier, 1999, ch. Voting in fuzzy rule-based systems for pattern classification problems, pp. 223–238.

[10] I. Jenhani, N. B. Amor, and Z. Elouedi, "Decision trees as possibilistic classifiers," *International journal of approximate reasoning*, 2008.

[11] L. Kaufman and P. J. Rousseeuw, *Statistical data analysis based on the L_1 norm and related methods*. Y. Dodge, 1987, ch. Clustering by means of medoids, pp. 405–416.

[12] E. Nikolaidis, S. Chen, H. Cudney, R. T. Haftka, and R. Rosca, "Comparison of probability and possibility for design against catastrophic failure under uncertainty," *Transactions of the ASME*, 2004.

[13] C. Olaru and L. Wehenkel, "A complete fuzzy decision tree technique," *Fuzzy sets and systems 138*, pp. 221–254, 2003.

[14] Y. Peng and P. A. Flach, "Soft discretization to enhance the continuous decision tree induction," *Integrating Aspects of Data Mining, Decision Support and Meta-Learning*, 2001.

[15] S. Tsang, B. Kao, K. Y. Yip, Wai-Shing, and S. D. Lee, "Decision trees for uncertain data," *IEEE Transactions on Knowledge and Data Engineering*, 2009.