Interpretable knowledge discovery from data with DC*

Marco Lucarelli[1] **Ciro Castiello**[1] **Anna M. Fanelli**[1] **Corrado Mencar**[1]

[1]Dept. of Informatics, University of Bari "A. Moro", Italy

Abstract

We present DC* (Double Clustering with A*) as an information granulation method specifically suited for deriving interpretable knowledge from data. DC* is based on two main clustering stages: the first is devoted to compressing multi-dimensional data into few prototypes that grab the main relationships among data; the second is aimed at finding a proper fuzzy granulation of each input feature so that the relations among data can be linguistically described in terms of fuzzy classification rules. We applied DC* as a stage in a knowledge discovery process, aimed at finding interpretable diagnostic rules for sleep-related breathing disorders.

Keywords: DC*, Knowledge discovery from data, Interpretability, sleep-related breathing disorders

1. Introduction

Knowledge discovery is the non-trivial process of identifying patterns in data that are valid, useful and comprehensible [1]. This process is made up of several stages: some of them are aimed at preparing data, others concern the subsequent application of data mining techniques which are mainly responsible for the discovery of patterns exhibiting the above mentioned properties. Eventually, these patterns are interpreted by a human analyst and, possibly, the process is re-iterated until a final objective is met. It is therefore important that the patterns coming from the data-mining stage are truly comprehensible for the human analyst. Such a requisite becomes crucial whenever patterns are intended for final users having little or no knowledge about the technical details behind the knowledge discovery process. In this way, the information provided is useful to support analysts in their activities (especially in decision-making tasks).

A typical case concerns knowledge discovery from medical data. Here, the final users are physicians who, most often than not, do not share a common language with computer scientists or data analysts. As a result, physicians often content themselves with basic statistical methods in their enquires, thus renouncing to exploit the power of modern data mining techniques. The crucial point is the lack of *semantic co-intension* between the medical knowledge of physicians and the semantics of the patterns resulting from data mining.

Semantic co-intension is a property of knowledge-based models. Such models represent systems or phenomena through a knowledge base. This is, in its turn, represented through some kind of symbolic formalization (usually, a rule-based inference engine). Each symbol of the knowledge base represents a part of the modeled system or phenomenon: in this sense, an *explicit* semantics of the symbol can be identified. On the other hand, the very same symbol is used to communicate knowledge in a broader sense. The recipient of the communication is supposed to understand the semantics of the symbol by reading it and framing its meaning in a (shared) background: this is the way the *implicit* semantics of the symbol is conveyed. Semantic co-intension is verified when the explicit semantics almost coincides with the implicit semantics [2].

Semantic co-intension requires that symbols and forms used in a model belong to the vocabulary and the language of the recipient. When the recipient is a human and the model has been automatically generated by some computational process (as in a data mining stage), the problem of co-intension becomes manifest. When not properly faced, the lack of semantic co-intension is a problem which turns into a lack of comprehensibility of the final model. In this case, the final user must blindly “trust” the model, which is unacceptable if the final user must rely on the model to take critical decisions (this is the case of medical diagnosis).

Natural language is the most commonly used channel for communicating information and knowledge. It is more imprecise and ambiguous than the mathematical language, yet it is very effective when the information itself to be communicated is imprecise too. In medical, economic, social sciences (just to mention a few fields of application) the use of natural language for knowledge exchange is a common practice. It is therefore desirable that knowledge-based models that are used in such fields could express a knowledge base in natural language, possibly in a simplified form (especially when such knowledge is automatically acquired from data).

Here fuzzy logic comes into play: most terms in natural language designate perception-based concepts (concepts acquired by people through the act of perception) that are sensory information organized in mind with the purpose of understanding the observed environment [3]. Since we all live in a dynamical environment, it is natural to assume that perception-based concepts are characterized by continuity and lack of boundaries. These concepts can be represented in a machine by fuzzy information granules [4].

Fuzzy information granulation is the process of generating fuzzy information granules from available data in order to capture the relationships embedded in a dataset. In data mining, fuzzy information granules are the patterns identified from data. The extra value of this approach is the possibility of describing the resulting fuzzy information granules by means of natural language terms: their proper choice triggers an overlap between explicit and implicit semantics, thus fulfilling the semantic co-intension requirement.

Information granules that can be linguistically described are called *interpretable*. However, the passage from a fuzzy information granule to its expression in natural language, far from being straightforward, is almost impossible unless a number of constraints are imposed in the granulation process. Several approaches exist for automatically generating interpretable fuzzy information granules from data, as witnessed by surveys available in literature for the interested reader [5].

In this paper we describe DC* (Double Clustering with A*) as a tool for interpretable fuzzy information granulation in a knowledge discovery process. DC* has several features that distinguish it from other alternative approaches proposed in literature.

First, DC* is able to derive fuzzy information granules from data which can be easily expressed in a simplified natural language. This is accomplished by generating strong fuzzy partitions for each data feature; then fuzzy information granules are defined by combining the fuzzy sets of each partition. Differently from other techniques, the generation of the strong fuzzy partitions is not made sequentially (i.e. one feature at a time) but *simultaneously*, thus ensuring that the multi-dimensional relationships in data are correctly represented. This is achieved by an informed search process based on the well-known A* algorithm.

The second key feature of DC* stands in its ability of automatically finding the best granularity (or specificity) level for all the fuzzy sets in each fuzzy partition. This is achieved by exploiting class information attached to each data sample. As a desirable side-effect, many data features are granulated by a single fuzzy set, i.e. they can be safely removed from the definition of the granules, thus simplifying their representation. Only a limited number of techniques exist in literature that perform fuzzy information granulation with variable granularity based on the multi-dimensional relationships revealed from data [6].

The application of DC* produces fuzzy information granules that are naturally translated into fuzzy classification rules expressed in natural language. This makes DC* a candidate for fulfilling the data mining stage in a knowledge discovery process.

DC* has already be introduced in past works of ours, mainly to show some technical advancements [7]. In this paper we present an application of this method to a real-world problem, within a more general process of knowledge discovery.

The main structure of DC* is described in the next section, while in section 3 we illustrate a medical case study and the related dataset. In section 4 we describe the application of DC* for discovering interpretable fuzzy diagnostic rules. The paper ends with some final notes in section 5.

2. Double Clustering with A* (DC*)

DC* is a method for extracting interpretable fuzzy information granules from numerical, pre-classified data in an automatic fashion. Defining the proper granularity in a data description task is a common problem in knowledge discovery processes. DC* faces such a problem by finding the minimal granularity level for all the involved features simultaneously, being also able to identify a subset of features which is useful for the overall representation purposes. The fuzzy information granules are therefore automatically provided by DC* in terms of number, shape and amplitude. The application of the method requires a single hyper-parameter to be a-priori specified, namely the upper bound for the final number of information granules that are going to be extracted: this enables the user's control on the overall granularity level of the final representation, while granting the possibility to automatically identify the optimal number of fuzzy granules.

Interpretability is a key characteristic of DC* and its acknowledgment is directly embedded in the method without requiring further post-processing. On the one hand, the minimal number of information granules leads to derive more readable partitions. On the other hand, the adoption of strong fuzzy partitions ensures the fulfillment of a number of interpretability constraints. Moreover, relying on the adoption of fuzzy logic enables the representation of perception-based concepts by means of natural language, thus promoting the application of the semantic co-intension property which is essential for knowledge-based models conceived to assist human-centered activities.

The working engine of the DC* method can be schematized into the subsequent implementation of two clustering stages: the first one is devoted to a preliminary compression of multi-dimensional data, aimed at identifying a reduced number of prototypes. These are projected on each dimension to activate the second clustering stage, devoted to identify the granularity of the fuzzy partitions. Finally, feature granules are fuzzified and exploited to describe in linguistic form the information granules over the feature space. The acquired knowledge is represented by IF-THEN fuzzy rules where only linguistic terms appear (one rule for each information granule). The following sections detail the aforementioned working engine.

2.1. Stage 1: Data compression

The first clustering stage of DC*, implementing a compression of data, operates over the whole feature space, taking into account the position of samples and their class information.

In formal terms, let

$$\mathbf{X} = [m_1, M_1] \times ... \times [m_n, M_n] \subseteq \mathbb{R}^n \tag{1}$$

be a n-dimensional Universe of Discourse (UoD),

$$C = \left\{c^1, c^2, \ldots, c^{n_C}\right\}$$

a finite set of classes, and

$$D = \{(\mathbf{x}_i, c_i) \in \mathbf{X} \times C : i = 1, 2, ..., n_D\} \tag{2}$$

the available dataset of n_D classified samples from $\mathbf{X}$. (It is assumed that each class label in C has at least one sample in D associated to it.) The data compression stage is aimed at defining a collection

$$P = \{(\mathbf{p}_j, c_j) \in \mathbf{X} \times C : j = 1, 2, ..., n_P\} \tag{3}$$

of prototypes, which represents aggregate information of the available samples (with $n_P \ll n_D$). The distribution of prototypes for each class is defined to preserve the original distribution of data as much as possible. The number of prototypes n_P is user-defined, being the only hyper-parameter to be set.

The data-compression step is performed by the Learning Vector Quantization algorithm LVQ1 proposed by Kohonen [8]. By means of an iterative approach, LVQ1 suitably moves the prototypes in $\mathbf{X}$, with the aim to best represent the underlying class distribution, on the basis of a comparison involving the classes of data and prototypes (see fig. 1).

2.2. Stage 2: Optimal granulation

The second DC* clustering stage is aimed at finding the optimal granulation of the UoD by exploiting the prototype collection P obtained in the previous step. In particular, an informed search procedure based on the well-known A* strategy [9] is performed, which allows to find an optimal solution according to some specific criteria (later described).

First, the prototype collection P is projected onto each dimension $d = 1, 2, \ldots, n$. We denote each projection set as

$$\begin{aligned} P_d = &\{(p_{hd}, c_h) \in [m_d, M_d] \times C : \\ &(\mathbf{p}, c_h) \in P \wedge p_{hd} = \Pi_d(\mathbf{p})\}, \end{aligned} \tag{4}$$

where $\Pi_d(\mathbf{p})$ projects $\mathbf{p}$ to the d-th dimension, and $h = 1, 2, \ldots, n_P$. (We assume that the prototypes are sorted, i.e. $p_{hd} \leq p_{h+1,d}$.) It should be noted that each element inside a projection inherits the class information from the corresponding prototype class.

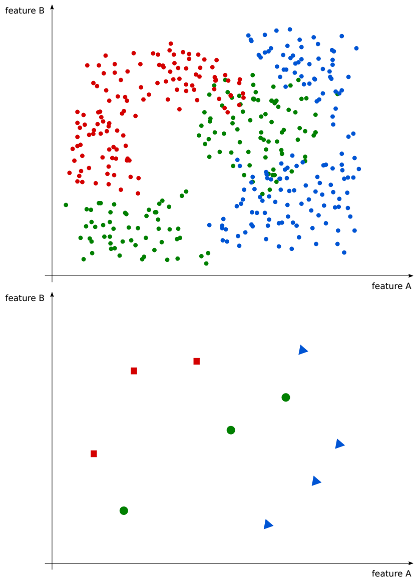

Figure 1: Original data (above) and multi-dimensional prototypes (below) resulting from data compression through LVQ1. Different shapes denote different class labels.

Given a projected prototype collection P_d, the second clustering stage is performed by operating on the related set of *cuts*, which are the boundaries of information granules. Formally, each cut is defined as the midpoint between two prototype projections labeled with different classes (laying on a specific dimension). The set of cuts for the d-th dimension is defined as

$$\begin{aligned} T_d = &\{t_{kd} : (p_{hd}, c_h), (p_{h+1,d}, c_{h+1}) \in P_d \\ &\wedge (c_h \neq c_{h+1}) \wedge t_{kd} = \frac{p_{hd} + p_{h+1,d}}{2}\} \end{aligned} \tag{5}$$

with $k = 1, 2, \ldots, n_{T_d}$ (see fig. 2). Any subset $S_d \subseteq T_d$ of cardinality n_{S_d} defines a clustering of the projections in P_d, where each cluster is defined by all the projections that are not separated by any cut in S_d.

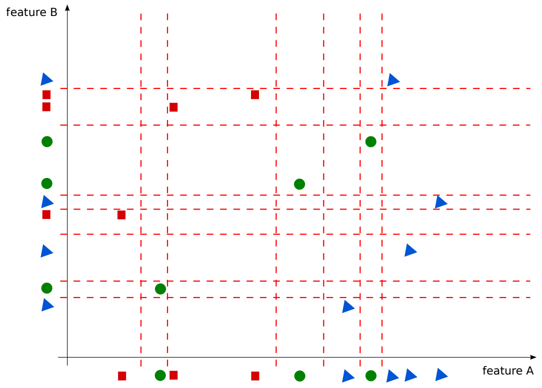

Figure 2: The set of all candidate cuts.

Given a subset of n_{S_d} cuts S_d, it is extended to include the boundary points (1):

$$\bar{S}_d = S_d \cup \{m_d, M_d\}\,.$$

Then, the Universe of Discourse $\mathbf{X}$ is partitioned into *hyper-boxes* defined as:

$$\begin{aligned}\mathbf{B}_{k_1,k_2,\dots,k_n} =& \{(x_1,\dots,x_n) \in \mathbf{X} : s_{k_d}, s_{k_d+1} \in \bar{S}_d \\ & \wedge s_{k_d} \le x_d \le s_{k_d+1}\}\end{aligned} \tag{6}$$

(it is assumed that $s_{k_d} < s_{k_d+1}$; also $s_0 = m_d$ and $s_{n_{S_d}} = M_d$). A hyper-box contains zero or more multi-dimensional prototypes of P and it is said *pure* if it is empty or if all its prototypes belong to the same class; otherwise it is said *impure* (see fig. 3). An *optimal* solution in DC* is intended as a set

$$\sigma = S_1 \times \cdots \times S_n \tag{7}$$

with *minimal* cardinality, that partitions the feature space into *pure* hyper-boxes only. Pure (non-empty) hyper-boxes are surrogates for information granules: data samples surrounding prototypes are mostly contained in the same hyper-box. The requisite of minimal cardinality yields the coarsest granulation of the feature space.

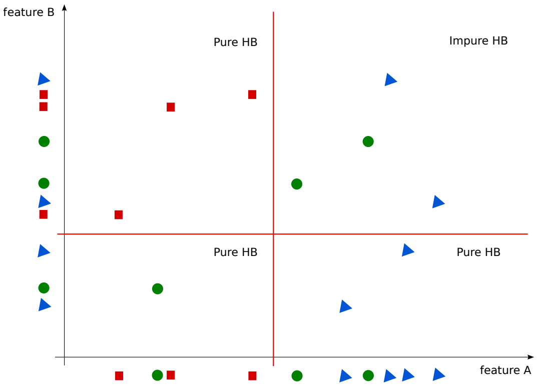

Figure 3: The hyper-box on the top-right is impure, all the others are pure.

The search for an optimal solution is a problem characterized by exponential time complexity, and a strategy based on the A* algorithm is adopted to face it. A* operates an informed search whose computational cost (which is still exponential in the worst case) can be reduced in most practical cases by properly designing its working set, as described in the following.

Search space. The search space $\mathbf{\Sigma}$ is defined as the set of all possible partitions of $\mathbf{X}$ as defined in (7). Hence, any state $\sigma \in \mathbf{\Sigma}$ is defined by a possible sequence of valid cuts on the n-dimensions. The initial node σ_0 to start the search is the trivial one, where $S_d = \emptyset$ for all $d = 1, 2, \dots, n$.

Test function. The test function recognizes if a given node σ is a goal (i.e. an optimal solution), hence it can be expressed as a binary truth function $\tau : \mathbf{\Sigma} \to \{\mathrm{T}, \mathrm{F}\}$. A node is *final* if and only if the resulting hyper-boxes (6) are all pure; because of the operating mode of A*, the first goal node is always optimal, hence the search can be terminated.

Successor operator. Given a node, the successor operator generates a set of new nodes. Each generated node differs from the original node by one additional cut on one dimension. A proper generation strategy avoids possible duplicates, i.e. same nodes generated by different nodes. This additive process eventually leads to a *terminal node* (the node where all cuts are used). By construction, the cuts of the terminal state generate pure hyper-boxes only: the termination of A* is guaranteed.

Cost function. In A* a cost function is required, which is defined as:

$$f(\sigma) = g(\sigma) + h(\sigma), \tag{8}$$

where $g(\sigma)$ is the *path-cost* which, in our settings, simply counts the number of cuts in a node, while $h(\sigma)$ is the heuristic function that *estimates* the minimum number of cuts needed to produce a final node passing from σ (i.e. to generate pure hyper-boxes from all impure hyper-boxes present in σ).

It is well known that the A* computational performance heavily relies on the heuristic function. In particular, the heuristic function has to be *admissible*: $h(\sigma)$ must never overestimate the true cost of the optimal solution. On the other hand, the closest is $f(\sigma)$ to the true cost, the fastest is the convergence of A* to the optimal solution. Finding the right trade-off between admissibility and informativeness is a challenging task and the DC* heuristic function exploits both the class-information contained in the impure hyper-boxes and the spatial relations of these hyper-boxes in the feature space [7].

Priority queue. A* uses a priority queue to store the nodes to be evaluated: at each step the first node in the queue is extracted and tested; if it is not a goal, the successor nodes are generated and put in the queue according to a priority level. In DC*, nodes are arranged in the queue according to a three-level priority aimed at privileging nodes with the potentiality of being goal (or ancestors for it). The highest priority is represented by the cost value $f(\sigma)$ (this is required by A*). The second priority level is intended to promote nodes corresponding to coarse-grained information granules. Finally, the third priority level is intended to promote nodes requiring a few features to represent information granules. (Features with no cuts are automatically removed.) Whenever nodes exhibit the same priority at a given level, they are ranked on the basis of the next priority level.

According to this priority schema, the second stage of DC* operates a simultaneous search of cuts on each dimension, such that: (i) the number of cuts is minimized, (ii) cuts on the same feature are well spaced; (iii) the number of features involved in granulation is small.

2.3. Granule fuzzification

In order to design a fuzzy partition of the feature space (which is ultimately oriented to express the relations among data in linguistic terms), the sets of cuts resulting from the second stage of DC* are properly translated into interpretable fuzzy information granules. To this pursuit, Strong Fuzzy Partitions (SFPs) are defined for each feature, since their use conjugates the fulfillment of a number of interpretability constraints with an easiness of application. Since DC* is a granulation method that makes use of cuts, the use of trapezoidal fuzzy sets is required to define SFPs [10]. In particular, we use the Variable Fuzziness (VF) approach, which moves from the assumption that fuzzy sets with larger support are more imprecise than fuzzy sets with smaller support. Therefore, the slopes of the trapezoidal membership functions are defined on the basis of the distance between adjacent cuts, so that closer cuts correspond to sharper fuzzy sets.

3. Case study: sleep-related breathing disorders

In this paper we present an application of DC* for knowledge discovery in a medical context, namely in the diagnosis of sleep-related breathing disorders (SRBDs) [11, 12]. In essence, SRBDs are induced by compromised blood oxygenation which translates into reiterated stress conditions for heart and brain. The consequences range from daily sleepiness (with possible danger of work or drive accidents) to heart attack or cerebral stroke.

In our analysis, three specific disorders are considered: the obstructive sleep apnea syndrome (OSAS), the obesity hypo-ventilation syndrome (OHS) and the overlap syndrome (OS). OSAS is characterized by pauses in breathing during sleep, due to the partial or total obstruction of the human airways. When breathing is temporary interrupted, the oxygen level drops, an occurrence that may cause angina, arrhythmia or heart attack. OSAS is manifested when a number of apnea events occur, together with other clinical symptoms affecting also daily activities. OHS is the combination of a severe over-weighting condition with hypoxia (deprivation of oxygen in some regions of the body) and hypercapnia (increased levels of carbon dioxide in the blood). People affected by OHS happen to stop breathing during sleep and are induced to continual sleepiness during the day. OHS negatively affects the quality of life and has been experimentally associated with an increased risk of death. OS is a common combination of OSAS and chronic obstructive pulmonary disease (COPD), which are a couple of very frequent pulmonary diseases affecting at the same time a great number of patients [13]. The negative effects related to OS (including hypoxemia during sleep) are greater than those produced by the single diseases considered alone.

The diagnosis of these types of SRBDs is performed by means of a multi-parametric sleep study known as polysomnography (PSG), which collects several data during the sleep of the patients. Records concern a number of biophysiological parameters deriving from monitoring processes including EEG, ECG, and many others. Even if PSG stands as the principal test for diagnosing SRBDs, it is affected by several drawbacks, mainly related to the comfort of the patient (who is supposed to be "wired" for a number of hours while sleeping in a sleep lab) and to the great costs connected with such a procedure. For those reasons, physicians ultimately welcome the realization of alternative methods of diagnosing which can be simpler and characterized by reduced expense.

In this context, we carried on a knowledge discovery process that applies DC* for discovering diagnostic rules in terms of interpretable fuzzy information granules. The following subsection reports the description of the dataset used for knowledge discovery and its pre-processing.

3.1. Data description and pre-processing

The global dataset D involved in the experimentation has been provided by the Department of Medical and Surgical Sciences[1] of the University of Foggia, Italy. It is composed of 186 samples related to patients affected by SRBDs and the information reported concerns 18 features: all of them pertain to basic parameters which include personal data, anamnesis and medical reports. Each sample is associated to a class label describing the diagnosis (OSAS or OS or OHS).

The nature of the dataset suggested the application of a pre-processing phase to face a number of problems:

1. the dataset is imbalanced (81.17% OSAS, 11.30% OS, 7.53% OHS);
2. the number of samples (186) is limited with respect to the number of involved features (18), thus triggering the well-known curse of dimensionality problem;
3. the presence of missing values (0.67% of the entire dataset).

To tackle problem 1, an oversampling technique has been adopted, namely the Synthetic Minority Over-sampling TEchnique (SMOTE) [14]. SMOTE allows the creation of synthetic samples to be added to the under-represented classes of examples. Such

[1] http://www.medicina.unifg.it

Table 1: The features selected for the dataset D. The last column specifies the % of samples with a value for each feature.

#	Feature	Range	Completeness
1	Age	[25, 82]	92.08%
2	ESS	[0, 29]	97.52%
3	TIFF	[41, 113]	96.04%
4	pO2	[32, 105]	100.00%
5	pCO2	[29.7, 108]	100.00%

synthetic samples are created by randomly selecting an arbitrary point along the line connecting one real sample and the next nearest one. In this way, the involved dataset has been filled up with additional samples, so that the representativeness of classes has been paired. (To avoid possible bias, SMOTE has been applied to the subset of samples without missing values.)

Problem 2 induced the necessity to reduce the number of involved features by selecting the most discriminating of them for the problem at hand. A Backward Feature Elimination (BFE) process has been applied on the datasets augmented by the SMOTE contribution. BFE is based on the iterated training of classifiers on the dataset, intended to identify, at each step, the feature that influenced the prediction results the least, in order to discard it (thus allowing each successive training to consider a reduced set of features) [1]. In our experimentation, for efficiency reasons, decision trees and neural networks have been adopted as classifiers and the BFE process ultimately allowed to reduce the dataset D to a number of 5 features (reported in table 1).

Finally, due to the limited number of samples at our disposal, we rejected to tackle problem 3 by simply discarding the examples troubled by the presence of missing values. Instead, missing values have been filled by predicting their values with Machine Learning tools[2].

4. Discovery of diagnostic rules with DC*

As previously asserted, DC* requires the specification of just one hyper-parameter (n_P), which specifies the number of prototypes derived in the first clustering stage, as defined in (3). Such hyper-parameter regulates the accuracy/interpretability trade-off of the final model: roughly speaking, higher values of n_P correspond to increased accuracy and reduced interpretability (and vice-versa). In order to investigate the performance trend with respect to the number of prototypes, we started by setting n_P equal to the number of classes in the dataset – i.e. one prototype for each class – and then doubling its value until the classification error became stable. DC* models have been imported into the FisPro tool [15], which provides a suite of functions that are useful for testing the derived fuzzy rule base.

Table 2: Accuracy and complexity results of DC* (avg ±std. dev.). In bold the best DC* model.

n_P	err (%)	# rules	# features
3	50.7±10.6	3±0	2±0
6	22.6±10.2	3.1±0.3	1.9±0.2
12	19.3±10.7	4.5±1.5	2.4±0.6
24	**10.0**±7.1	**4.6**±1.1	**2.3**±0.4
48	11.1±7.2	7.0±2.6	2.8±0.8

In order to validate the DC* results, five different[3] 10-fold Cross-Validations have been performed leading to 50 different training-test sets. Values reported in tables 2 and 3 are meant as the mean values on the test sets of all the folds.

Table 2 shows the obtained results in terms of both accuracy (classification error) and interpretability of the models with respect to the number of prototypes set in DC*. We compared the obtained models with standard C4.5 Decision Trees (no pruning), Multi-Layer Perceptrons (10 hidden neurons) and the Naive Bayes classifiers, which carried out, respectively, the following mean classification errors: 6.4% (DT), 8.2% (MLP) and 12.2% (NB). We observe that DC* produces models with slightly higher classification errors (this is due to the bias introduced by the interpretability constraints), but it requires a small number of features and only a few rules to describe the classification function (the other techniques, on the other hand, require all the features).

The best accuracy results of DC* have been obtained with 24 prototypes. In this case the method built a model with 4-5 rules, where only 2-3 out of 5 features have been involved. Thus, DC* has automatically found a subset of significant features in order to describe the problem, hence operating a feature-selection process directly dictated by data. Moreover, such a result proves how the number of prototypes represents just an upper bound of the maximum number of possible rules, without being a strict constraint for the optimization process. This is due to the summarization effects of prototypes belonging to the same information granule, which are represented by a single rule. It is interesting to observe that the number of rules is quite stable on the different runs. This is an indication that DC* is capable of finding an intrinsic granularity of data that is only loosely related with the hyper-parameter n_P which, however, affects the accuracy of the final model.

Table 3 summarizes the DC* classification performances in a confusion matrix. As shown by the

[2] More specifically, we used the Google Prediction API to perform data filling through the Smart Autofill Spreadsheets Add On. See https://cloud.google.com/prediction/docs/smart_autofill_add_on for further information.

[3] Practically, it means five different seeds in the random fold generation.

Figure 4: An example of interpretable fuzzy rule base derived by DC*.

if TIFF is low	and pCO2 is low	then OS
if TIFF is high	and pCO2 is low	then OSAS
if TIFF is high	and pCO2 is medium	then OHS
if TIFF is high	and pCO2 is high	then OSAS

reported values, when considering the single diagnoses separately, OSAS is characterized by a more accurate prediction (4.35% of misclassifications over the total number of samples belonging to that class), followed OS (21.79% of misclassifications) and OHS (54.27% of misclassifications).

Table 3: Average confusion-matrix of the classification operated by DC* with 24 prototypes.

obs/inf	OSAS	OS	OHS	tot
OSAS	**77.64%**	1.50%	2.04%	81.17%
OS	1.61%	**8.84%**	0.85%	11.30%
OHS	4.09%	0.00%	**3.44%**	7.53%
tot	83.34%	10.33%	6.33%	**89.92%**

Misclassified cases have been mostly labeled as OSAS (particularly, all the wrongly classified OHS samples have been labeled as OSAS). Such a behavior may be explained by referring to the intrinsic imbalance characterizing the original dataset. However, to tackle the problem of OHS class misclassification, suited strategies should be adopted in future works.

Figure 4 and 5 offer an insight into a specific predictive model derived by DC* (for representative purpose we chose a model exhibiting performance values fitting in with the average obtained results). Particularly, a fuzzy rule base expressed in natural language is illustrated in figure 4 and figure 5 depicts the fuzzy partitioning of the corresponding feature space (reduced to a couple of features).

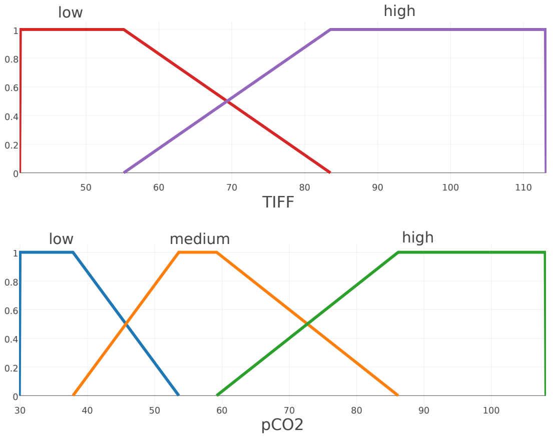

Figure 5: Fuzzy sets defined for the features involved in the rule base illustrated in figure 4.

The fuzzy rule base is particularly remarkable for its simplicity and readability: the adoption of linguistic terms and the compactness of the information provided allow physicians to easily understand patterns underlying data, thus favoring the validation of the extracted knowledge. Actually, only two out of five features have been involved in the predictive inference engine.

Fuzzy set labeling is a key point to allow a thorough rule base comprehension (this is where the semantic co-intension property comes into play for the benefit of physicians' trusting in the derived knowledge representation). Indeed, the linguistic terms adopted to define the variable values (*low*, *medium*, *high*) have been associated to the automatically identified fuzzy sets in a straightforward way. The intrinsic capabilities of DC* are fairly appreciated while looking at both the fuzzy set shapes and arrangement: the proposed data mining method is able to automatically identify the suitable number of information granules, together with their configuration and position over the feature dimensions, so that the problem at hand can be effectively described. As previously observed, wider granule amplitudes are associated with greater fuzziness expression.

Finally, it is noteworthy to underline how those kinds of results are obtained by DC* through a totally data-driven process, without requiring any user intervention nor a predefined number of granules. The derived fuzzy sets adapt to the data distribution in the multi-dimensional space and fulfill a number of constraints ensuring the interpretability of the final model.

Acknowledgement

This work was partially supported by "National Group of Computing Science, Italian Institute of Higher Mathematics (GNCS-INDAM).

5. Conclusion

The paper shows the employment of DC* to tackle a real-world problem. Due to the particular context of application, pertaining to the field of medical diagnosis, a predictive model merely relying on satisfactory accuracy results is not sufficient and the production of highly interpretable classification rules is notably welcomed. The available real data underwent a thorough process of knowledge discovery from data, including a pre-processing stage (addressed to face a number of issues related to the particular dataset configuration) and a data mining stage (properly performed by means of DC*).

The obtained results demonstrate that DC* is capable to derive very simple rules which are effective in capturing the most relevant relations among data. As a confirmation, the accuracy results proved to be not far from those obtained by other standard techniques which, on the other hand, produce more complex predictive models. As concerning the specific medical diagnosis problem, the obtained rule base has been confronted with expert knowledge, showing interesting overlaps that stimulate further insights for future research.

References

[1] O. Maimon and L. Rokach. Introduction to knowledge discovery and data mining. In O. Maimon and L. Rokach, editors, *Data Mining and Knowledge Discovery Handbook*, pages 1–15. Springer US, 2010.

[2] J. M. Alonso, C. Castiello, and C. Mencar. Interpretability of fuzzy systems: Current research trends and prospects. In J. Kacprzyk and W. Pedrycz, editors, *Springer Handbook of Computational Intelligence*. Springer, 2015.

[3] L. J. West. Perception. Entry of *Encyclopædia Britannica Online*, 2015.

[4] L. Zadeh. Toward a theory of fuzzy information granulation and its centrality in human reasoning and fuzzy logic. *Fuzzy Sets and Systems*, 90(2):111–127, 9 1997.

[5] C. Mencar and A. M. Fanelli. Interpretability constraints for fuzzy information granulation. *Information Sciences*, 178(24):4585–4618, 2008.

[6] M. Zeinalkhani and M. Eftekhari. Fuzzy partitioning of continuous attributes through discretization methods to construct fuzzy decision tree classifiers. *Information Sciences*, 278:715–735, 9 2014.

[7] M. Lucarelli and C. Mencar. A new heuristic function for DC*. In *Fuzzy Logic and Applications (10th International Workshop, WILF 2013)*, Lecture Notes in Artficial Intelligence, pages 44–51, 2013.

[8] T. Kohonen. *Self-Organizing Maps*, volume 30 of *Springer Series in Information Sciences*. Springer-Verlag, 2001.

[9] S. Edelkamp and S. Schrödl. *Heuristic Search: Theory and Applications*. Morgan Kaufmann, 2012.

[10] C. Mencar, M. Lucarelli, C. Castiello, and F. Anna Maria. Design of strong fuzzy partitions from cuts. In *Proceedings of the 8th conference of the European Society for Fuzzy Logic and Technology*, Advances in Intelligent Systems Research, pages 424–431, 2013.

[11] V. Tsara, A. Amfilochiou, M. J. Papagrigorakis, D. Georgopoulos, and E. Liolios. Guidelines for diagnosis and treatment of sleep-related breathing disorders in adults and children. Definition and classification of sleep related breathing disorders in adults: different types and indications for sleep studies (part 1). *Hippokratia*, 13(3):187–91, 7 2009.

[12] D. Lacedonia, G. E. Carpagnano, M. Aliani, R. Sabato, M. P. Foschino Barbaro, A. Spanevello, M. Carone, and F. Fanfulla. Daytime PaO2 in OSAS, COPD and the combination of the two (overlap syndrome). *Respiratory medicine*, 107(2):310–6, 3 2013.

[13] D. Lacedonia, R. Tamisier, F. Roche, D. Monneret, J. P. Baguet, P. Lévy, and J. L. Pépin. Respective effects of osa treatment and angiotensin receptor blocker on aldosterone in hypertensive osa patients: a randomized cross-over controlled trial. *International journal of cardiology*, 177(2):629–31, 12 2014.

[14] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer. Smote: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research*, 16:321–357, 2002.

[15] S. Guillaume and B. Charnomordic. Learning interpretable fuzzy inference systems with fispro. *Information Sciences*, 181(20):4409–4427, 10 2011.