Explainable artificial intelligence for breast cancer: A visual case-based reasoning approach

Jean-Baptiste Lamy, Boomadevi Sekar, Gilles Guezennec, Jacques Bouaud, Brigitte Séroussi

Artificial Intelligence In Medicine 94 (2019) 42–53

Contents lists available at ScienceDirect

Artificial Intelligence In Medicine

journal homepage: www.elsevier.com/locate/artmed

Explainable artificial intelligence for breast cancer: A visual case-based reasoning approach

Jean-Baptiste Lamy[a,*], Boomadevi Sekar[b], Gilles Guezennec[a], Jacques Bouaud[a,c], Brigitte Séroussi[a,d]

[a] *LIMICS, Université Paris 13, Sorbonne Universités, INSERM UMRS 1142, 93017 Bobigny, France*
[b] *School of Computing and Mathematics, Ulster University, United Kingdom*
[c] *AP-HP, DRCI, Paris, France*
[d] *AP-HP, Hôpital Tenon, Département de Santé Publique, Paris, France*

ARTICLE INFO

Keywords:
Case-based reasoning
Visual explanation
Explainable Artificial Intelligence
Data-driven decision making
Multidimensional Scaling
Breast cancer

ABSTRACT

Case-Based Reasoning (CBR) is a form of analogical reasoning in which the solution for a (new) query case is determined using a database of previous known cases with their solutions. Cases similar to the query are retrieved from the database, and then their solutions are adapted to the query. In medicine, a case usually corresponds to a patient and the problem consists of classifying the patient in a class of diagnostic or therapy. Compared to "black box" algorithms such as deep learning, the responses of CBR systems can be justified easily using the similar cases as examples. However, this possibility is often under-exploited and the explanations provided by most CBR systems are limited to the display of the similar cases.

In this paper, we propose a CBR method that can be both executed automatically as an algorithm and presented visually in a user interface for providing visual explanations or for visual reasoning. After retrieving similar cases, a visual interface displays quantitative and qualitative similarities between the query and the similar cases, so as one can easily classify the query through visual reasoning, in a fully explainable manner. It combines a quantitative approach (visualized by a scatter plot based on Multidimensional Scaling in polar coordinates, preserving distances involving the query) and a qualitative approach (set visualization using rainbow boxes). We applied this method to breast cancer management. We showed on three public datasets that our qualitative method has a classification accuracy comparable to *k*-Nearest Neighbors algorithms, but is better explainable. We also tested the proposed interface during a small user study. Finally, we apply the proposed approach to a real dataset in breast cancer. Medical experts found the visual approach interesting as it explains why cases are similar through the visualization of shared patient characteristics.

1. Introduction

Case-Based Reasoning (CBR) [1] is a form of analogical reasoning based on the memory-centered cognitive model. It came from the field of cognitive science, and is now part of artificial intelligence [2]. In CBR terminology, a case is a problem situation. CBR reuses old cases, which solution is known, to produce a solution for a new case, called the *query case*. A typical CBR system includes a case database, which is a $(p + 1)$-dimensional dataset $X = \{x_1, x_2, \ldots, x_i, \ldots\}$ with $x_i \in \mathbb{A}_1 \times ...\times \mathbb{A}_k \times ...\times \mathbb{A}_p \times Y$ where $\mathbb{A}_k$ are the dimension spaces and Y is the solution space. The query case can be represented as $q \in \mathbb{A}_1 \times ...\times \mathbb{A}_k \times ...\times \mathbb{A}_p$. CBR follows a cycle of four phases: *retrieve* from the case base the old cases that are the most similar to the query, *reuse* the information and knowledge embedded within similar resolved cases to produce a solution for the query case, *revise* the solution to adapt it to the query case, and *retain* the query case with the chosen solution in the case database.

CBR has been applied to many domains, including medicine [3]. In a medical CBR system, a case usually corresponds to a patient and the problem to solve typically consists of classifying a new patient according to various classes. For a diagnostic system targeting a given disorder, there are commonly two classes: healthy *vs* diseased. For a therapeutic system, there are several classes corresponding to the various categories of possible treatments. The case database contains

[*] Corresponding author at: LIMICS, UFR SMBH, Université Paris 13, 74 rue Marcel Cachin, 93017 Bobigny, France.
E-mail addresses: jibalamy@free.fr (J.-B. Lamy), b.sekar@ulster.ac.uk (B. Sekar), gilles.guezennec@univ-paris13.fr (G. Guezennec), jacques.bouaud@sap.ap-hop-paris.fr (J. Bouaud), brigitte.seroussi@tnn.aphp.fr (B. Séroussi).

https://doi.org/10.1016/j.artmed.2019.01.001
Received 10 August 2018; Received in revised form 21 December 2018; Accepted 7 January 2019

previous patients, for which the diagnosis or the treatment is known. Many therapeutic decision support systems implement evidence-based clinical practice guidelines [4]. These systems are therefore knowledge-based rather than case-based. However, CBR is still an interesting approach for patients that are not covered by clinical practice guidelines (they can represent up to 45% of patients [5]), or when guideline's recommendations cannot be applied, *e.g.* due to contraindications or when the patient refuses the recommended therapy. Moreover, CBR and guideline-based approaches can also be combined together [6].

Many data-driven classification approaches in artificial intelligence suffer from a lack of explainability; this is particularly true for "black box" approaches like deep learning, *e.g.* IBM Watson, which has been recently applied to breast cancer therapy [7]. However, in medical systems, black boxes are usually not well-appreciated by physicians since they prefer to understand how the system produces a recommendation [8], and automatic decision-support systems are often perceived as a threat and a loss of control [9]. Indeed, years ago, explainability was ranked by physicians as the most desirable feature of a clinical decision support system [10]. Today, in France, the recent Villani report [11] on artificial intelligence recommends to "open the black-box of artificial intelligence", with a special focus on medicine.

Explainable Artificial Intelligence (XAI) is a field that focuses on designing intelligent systems that are able to explain their recommendations to a human being. Biran et al. [12] distinguish two approaches: (a) interpretable models, which rely on non-black box systems such as rule-based ones, and (b) prediction interpretation and justification, which aim at generating explanations for a black box algorithm. The same authors mention a third approach, visualization. XAI has been particularly studied in the military domain [13,14]. With regard to XAI, CBR is particularly interesting because the similar cases can be used as examples for justifying the response of the system. This can be considered as an interpretable model. However, in terms of explanations, most CBR systems are limited to the display of the similar cases.

The presented work is part of the DESIREE (Decision Support and Information Management System for Breast Cancer) European project,[1] aimed at developing web-based services for the management of primary breast cancer. In this context, we propose a CBR system able to classify a query case using an automatic algorithm (displayed as "1)" in the general overview shown in Fig. 1), but also through visual reasoning ("2)" in Fig. 1). It includes a visual interface displaying quantitative and qualitative similarities between the query and the similar cases. This visual approach can be used independently, or for explaining the results of the automatic algorithm ("3)" in Fig. 1). We combined a quantitative approach, corresponding to a scatter plot produced using Multidimensional Scaling (MDS) in polar coordinates, and a qualitative approach, based on rainbow boxes [15], a recent technique for overlapping set visualization. We translated the visual reasoning permitted by the interface into algorithms, in order to formalize it and to evaluate whether the similarities displayed can be used for classification with a good accuracy. We applied our method to breast cancer, and we evaluated it on three public datasets. We also presented the interface to 11 medical experts for usability and acceptability validation.

Scatter plots have already been widely used for visual classification and CBR, although rarely in polar coordinates that allow preserving all distances involving the query. On the contrary, rainbow boxes have currently been only used for presenting medical knowledge, such as drug properties [16,17], but their application to CBR and their association with a scatter plot is new. A preliminary version of this work was presented in a French workshop [18], proposing the general approach but without detailed methods or experiments.

Since it is easier to formalize a visual approach than to translate an algorithm visually, we initially developed our method from the visual side, and here, we will present it in that way.

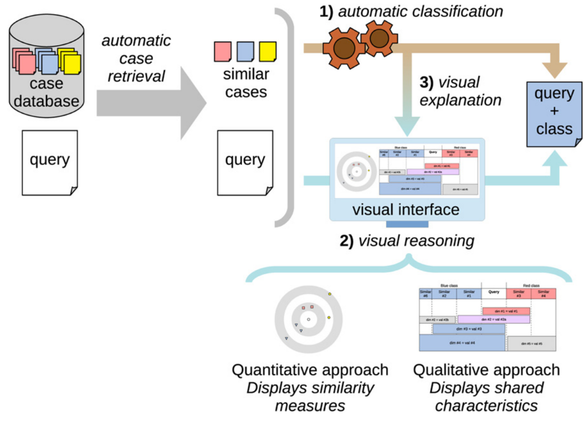


Fig. 1. General overview of the proposed CBR approach.

The rest of the paper is organized as follows. Section 2 presents some related works on CBR and high-dimensional multivariate data visualization, and also introduces the optimization algorithm that we used. Section 3 presents the proposed visual interface. Section 4 describes its translation into classification algorithms. Section 5 presents an application to breast cancer, including experimental results on machine learning datasets. Section 6 presents an application to real data for therapeutic decision support in breast cancer. Section 7 discusses the methods and the results. Finally, Section 8 concludes.

2. Related works

2.1. Case-based reasoning

kNN is a well-known algorithm for automatic CBR [19]. It consists of a "majority vote" between the similar cases: the query case is classified in the class that is the most frequently observed in the similar cases. Distance-weighted kNN (WkNN) [20] is a variant, in which the weight of a similar case is inversely proportional to the distance between the similar case and the query case. WkNN is known to deliver better accuracy than kNN, but with a modest improvement (about 1–2% [21]), and often requires higher value of *k*. In WkNN, the weight of a similar case *i* is:

$$w_i = \begin{cases} 1, & \text{if } d_{\max} = d_{\min} \\ \frac{d_{\max} - d_i}{d_{\max} - d_{\min}}, & \text{otherwise} \end{cases} \quad (1)$$

where d_i is the distance between the similar case *i* and the query case, and d_{min} and d_{max} are the minimum and maximum values of d_i.

It has been shown that explanations based on similar cases are more convincing than explanations based on rules [22], although this possibility has not been widely used and many explanation systems are not visual and rely on a single similar case. Few visual approaches have been proposed for CBR, and most of them do not target the CBR final users. They rather focus on helping developers with the design of CBR systems. For example, CTBV (Case Base Topology Viewer) [23] uses scatter plots to visualize the topology of a case base and compares the impact of various similarity measures on CBR, and Zhu et al. [24] used treemaps for visualizing the result of case clustering. On the contrary, Massie et al. [25] proposed a visual approach for providing explanations of CBR to final users, using parallel coordinates. However, the authors demonstrate their system with three similar cases and it may not scale well with more.

In the literature, most of medical CBR systems were aimed at helping clinicians to diagnose a given disorder, or a small number of close disorders [3], and these systems were limited to quantitative aspects [26,27].

[1] H2020 PHC-30-2015 #690238.

2.2. High-dimensional multivariate data visualization

Many techniques exist for visualizing high-dimensional or multivariate data; we will focus on the two approaches relevant for the presented work. First, *dimension reduction* approaches reduce the number of dimensions to 2 or 3, at the price of an information loss, and visualize the results using a scatter plot. The main techniques for dimension reduction are Principal Component Analysis (PCA), Multidimensional Scaling (MDS) and Self-Organizing Map (SOM) [28]. PCA is a statistical method. It generates new dimensions that are linear combinations of the original ones, and that maximize the variance on the first new dimensions. An example of PCA-based visualization is iPCA [29].

MDS [30] performs non-linear dimension reduction. MDS works on a distance matrix $d = (d_{i,j})_{1 \leq i,j \leq n} \in \mathbb{R}$, where $d_{i,j}$ is the distance between the elements i and j. It minimizes the stress function, which takes into account the distances in the original distance matrix $d_{i,j}$ and the distances in the generated scatter plot $\delta_{i,j}$. Various quality metrics have been proposed for high-dimensional data visualization [31]. Two major types of MDS exist: *metric* MDS tries to preserve distances while *non-metric* MDS tries to preserve the ordering between distances but not their absolute values. Many variants exist, such as relational perspective map (RPM) [32], which is similar to MDS but optimizes an energy function instead of a stress function. However, when using scatter plots in CBR, all points are not equivalent: one represents the query while the others represent the similar cases. Thus, all distances are not equivalent: since the objective is to classify the query, distances between the query and similar cases are more important than distances between two similar cases. Klawonn et al. [26] proposed *case-centered MDS* for medical diagnostic, a 3-dimensional MDS approach based on polar coordinates. It preserves the distances involving one point (in CBR, the query), to the detriment of the other distances. Rehm et al. [33] also proposed an MDS approach in polar coordinates in POLARMAP, aimed at adding new points without high computational costs, but not targeting CBR.

Second, when considering qualitative multivariate data (or quantitative data that has been discretized), *set visualization* techniques can be applied: sets of elements having a given value in a given dimension can be visualized. Many techniques exist for set visualization [34]. Recently, we introduced a new technique, called *rainbow boxes* [15], which has been initially applied to the comparison of drug properties [16]. In rainbow boxes, the elements (*e.g.* cases) are shown in columns, and the sets are represented by rectangular boxes placed below column headers (see example in Fig. 2, on the right). Each box covers the columns corresponding to the elements belonging to the set. The column order is computed using a heuristic optimization algorithm, which tries to order the columns so as the elements belonging to similar sets are contiguous. When it is not possible to have them contiguous for a given set, holes are present in the set's box. Boxes can be colored according to various schemes. Finally, boxes are stacked vertically. Two boxes can be next to each other as long as they do not occupy the same columns. We proposed a proportional version of rainbow boxes [17], in which the height of the boxes is an additional visual variable.

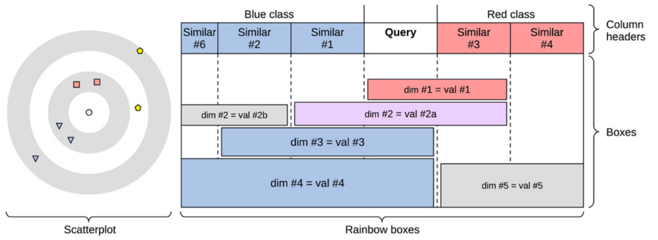


Fig. 2. General organization of the proposed interface for visual case-based reasoning ("dim" stands for dimension and "val" for value).

In this work, we chose polar MDS for its ability to produce a scatter plot from a wide range of data, to cope with non-linearity, and to preserve distances involving the query. Rainbow boxes were chosen for their ability to display qualitative information, which is lacking in scatter plot.

2.3. Artificial Feeding Birds (AFB) metaheuristic

Several visualization techniques require to solve optimization problems, including MDS and rainbow boxes. Nature-inspired metaheuristics [35] are simple, efficient and adaptable optimization algorithms. Here, we chose Artificial Feeding Birds (AFB) [36], a recent metaheuristic inspired by the behavior of pigeons.

AFB considers a population of artificial birds (usually, 20 birds). The position of each bird represents a candidate solution for the optimization problem. The algorithm performs several cycles; in each cycle, each bird performs one move. Four moves are possible: (1) walk to a random position close to the actual one, (2) fly to a random position, (3) fly to the best position found by the same bird yet, and (4) fly to join the position of another random bird. Move #4 is allowed only for large birds, which represent 25% of the bird population. Moves #3 and #4 are totally independent from the optimization problem. On the contrary, moves #1 and #2 depend on the types of optimization problem. Consequently, AFB can be applied to any optimization problem that is defined by a triplet of functions (*cost*, *fly*, *walk*), where *cost* is the cost function to minimize, *fly* is a function that returns a totally random solution and *walk* is a function that returns a random solution close to another previous solution.

AFB was chosen because it is very easy to adapt to new problems, including constrained combinatorial optimization, and we successfully used it previously for optimizing rainbow boxes [36] but also for quantitative preference learning [37].

3. Visual interface

3.1. General principles

In this work, we consider a CBR problem defined by a $(p + 1)$-dimensional database X and a query case q. For each dimension, the corresponding values can be real numbers ($\mathbb{R}$), integer numbers ($\mathbb{N}$), booleans ($\{True, False\}$) or a set of nominals ($\{a, b, c \ldots\}$). Y is the solution space, *i.e.* the set of possible solutions; here we consider a finite set of classes $Y = \{y_1, y_2, \ldots\}$ with $2 \leq |Y| \leq 10$. The visual interface aims at helping a user to find to which class q belongs. It takes 2 parameters: n, the total number of cases displayed in the interface (the first one being q, thus $n - 1$ similar cases are retrieved from X), and m, the maximum number of qualitative boxes to display in rainbow boxes.

A specific color is associated with each class in Y. The interface aims at translating the problem "which class y does q belong to?" into a visual problem: "what is the dominant color?". Fig. 2 shows the general organization of the visual interface. The interface is divided in two parts: scatter plot (left) and rainbow boxes (right).

The scatter plot is a representation in two (arbitrary) dimensions of the distance matrix between the cases. q is displayed as a white point in the center, and similar cases are represented by points colored according to their class. In addition to colors, shapes (*e.g.* circle, square, diamond, *etc.*) are used to encode the class, for helping color-blind users. The scatter plot aims at preserving distances, *i.e.* the closer two points are, the more similar the two cases they represent. Axes are not shown because they are meaningless in MDS. Instead, the background of the scatter plot displays a target centered on q, for facilitating the appreciation of distances (especially when they are not in the same direction, *e.g.* a vertical distance *vs* a diagonal one).

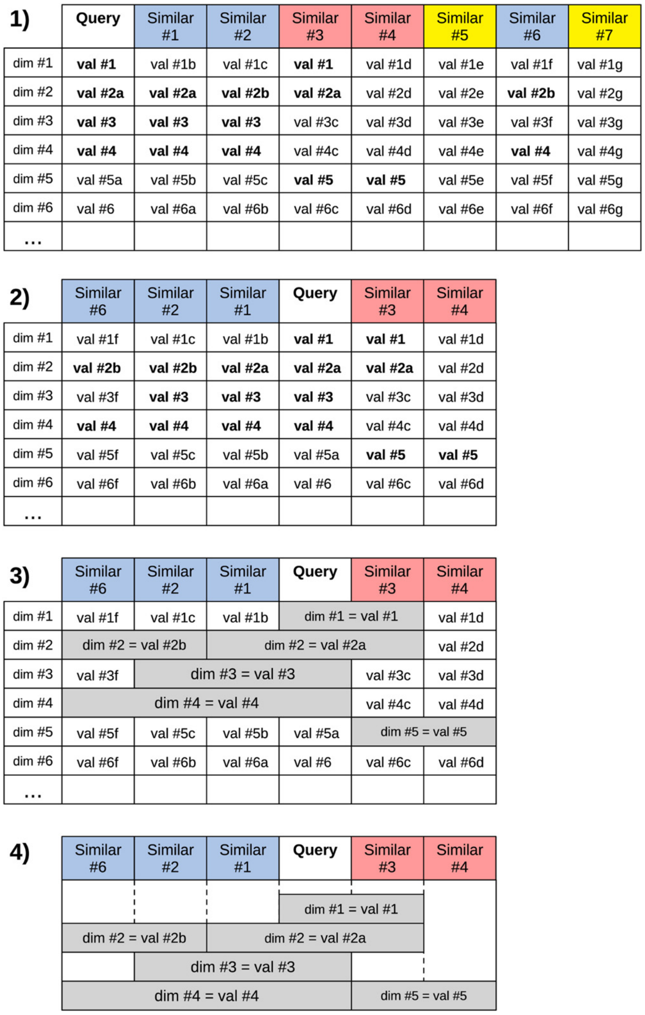
On the scatter plot, one can determine the number of similar cases in each class. One can also identify the class to which belong the cases that are the most similar to q, *i.e.* the color of the points closest to the white point. In Fig. 2, there are 3 similar cases belonging to the blue class *vs* only 2 for the red and yellow classes, and the most similar cases belong to the red and blue classes. In addition, one can verify that the colored points are well-separated (*e.g.* in Fig. 2, red points are at the top of the scatter plot, yellow points in the right and blue points in the bottom left). When the colored points are not well-separated but mixed together, it may indicate that the attributes in the database or the current CBR design do not allow classifying the cases correctly. Therefore, in this situation, the results of the CBR approach must be interpreted cautiously.

Rainbow boxes display a qualitative comparison on a subset of n' cases (with $n' \leq n$), containing q and the similar cases belonging to the two best candidate classes (in Fig. 2, they are the blue and the red classes). Each case is displayed as a column, and cases are grouped by class, with q in the middle. The column header color indicates the class of similar cases. The column width is proportional to the similarity of the similar case, *e.g.* here one of the blue column is smaller because the corresponding case is less similar to the query (as shown in the scatter plot). Some (dimension, value) pairs are displayed below in boxes, each box occupying the columns of the cases for which the (dimension, value) pair holds. Mutual Information (MI, detailed later in Section 3.5) is used to select the most interesting (dimension, value) pairs with regard to the class distribution, and also to compute the height of the boxes: taller boxes correspond to higher MI. Finally, the color of a box is gray if it does not contain q, and otherwise the color is a mix of the two colors associated with the two classes, in proportion equivalent to the ratio of the two classes in the box (each case being weighted by its similarity), *e.g.* in Fig. 2, the box "dim #2 = val #2a" is 50% blue and 50% red, thus violet.

Column headers allow the quick identification of the two main candidate classes for q and their weighted ratio over the similar cases. In Fig. 2, there are 3 similar cases belonging to the blue class, and only 2 belonging to the red one. Colored boxes show the characteristics, *i.e.* (dimension, value) pairs, shared between q and the similar cases. Box color indicates which class they orientate to. Notice how the column width weights each similar case according to its similarity. In Fig. 2, there are two tall blue boxes, one small violet box and one small red box. Blue boxes are taller and more numerous: this argues in favor of the blue class. The user can adapt or mitigate his choice by taking into account the pertinence of the dimensions shown in the boxes. Finally, gray boxes might suggest arguments for *not* choosing a class, *e.g.* the tall gray box labeled "dim #5 = val #5″ could be an argument for not choosing the red class, because all cases belonging to this class have the given value in dimension #5, and q has not. However, being different according to a dimension does not necessarily imply that there is no similarity according to others.

Fig. 3 shows the various steps explaining how rainbow boxes are built from a plain table. Step 1 is the data table, with cases in columns. Colors identify classes, and values in bold correspond to those shared by several cases and retained later in rainbow boxes. In step 2, only the two best candidate classes are kept, and columns (*i.e.* cases) have been reordered by similarity, with the query in the middle. In step 3, boxes are created for values shared between several cases. Notice how shared values are much easier to identify at this step, compared to step 1. In step 4, only the boxes are kept. Rows are removed, and boxes are stacked at bottom. The final step (corresponding to the one shown in Fig. 2, right) consists in adding box colors, box heights and column widths.

To sum up, the visual interface supports the visual classification of the query case by three means: (1) the scatter plot shows the class associated with the closest similar cases, (2) the scatter plot and rainbow boxes headers show the number of similar cases belonging to each class, and their similarity level, and (3) the boxes indicate *how* the similar cases are similar to the query, and toward which class each characteristic value orientates. (1) and (2) allow a *quantitative* approach for classification, while (3) allows a *qualitative* approach based on specific (dimension, value) pairs.



1)

	Query	Similar #1	Similar #2	Similar #3	Similar #4	Similar #5	Similar #6	Similar #7
dim #1	**val #1**	val #1b	val #1c	**val #1**	val #1d	val #1e	val #1f	val #1g
dim #2	**val #2a**	**val #2a**	**val #2b**	**val #2a**	val #2d	val #2e	**val #2b**	val #2g
dim #3	**val #3**	**val #3**	**val #3**	val #3c	val #3d	val #3e	val #3f	val #3g
dim #4	**val #4**	**val #4**	**val #4**	val #4c	val #4d	val #4e	**val #4**	val #4g
dim #5	val #5a	val #5b	val #5c	**val #5**	**val #5**	val #5e	val #5f	val #5g
dim #6	val #6	val #6a	val #6b	val #6c	val #6d	val #6e	val #6f	val #6g
…								

2)

	Similar #6	Similar #2	Similar #1	Query	Similar #3	Similar #4
dim #1	val #1f	val #1c	val #1b	**val #1**	**val #1**	val #1d
dim #2	**val #2b**	**val #2b**	**val #2a**	**val #2a**	**val #2a**	val #2d
dim #3	val #3f	**val #3**	**val #3**	**val #3**	val #3c	val #3d
dim #4	**val #4**	**val #4**	**val #4**	**val #4**	val #4c	val #4d
dim #5	val #5f	val #5c	val #5b	val #5a	**val #5**	**val #5**
dim #6	val #6f	val #6b	val #6a	val #6	val #6c	val #6d
…						

3)

	Similar #6	Similar #2	Similar #1	Query	Similar #3	Similar #4
dim #1	val #1f	val #1c	val #1b	dim #1 = val #1		val #1d
dim #2	dim #2 = val #2b		dim #2 = val #2a			val #2d
dim #3	val #3f	dim #3 = val #3			val #3c	val #3d
dim #4	dim #4 = val #4				val #4c	val #4d
dim #5	val #5f	val #5c	val #5b	val #5a	dim #5 = val #5	
dim #6	val #6f	val #6b	val #6a	val #6	val #6c	val #6d
…						

4)

Similar #6	Similar #2	Similar #1	Query	Similar #3	Similar #4
			dim #1 = val #1		
dim #2 = val #2b		dim #2 = val #2a			
	dim #3 = val #3				
dim #4 = val #4				dim #5 = val #5	

Fig. 3. The various steps for building rainbow boxes, starting from a data table.

The interface is generated in 6 steps, detailed in the following subsections.

3.2. Selecting similar cases

The $n - 1$ cases most similar to q are selected from database X, using a common CBR technique. This is the retrieval phase, which is not the focus of the current paper. Many methods have been proposed, and any could be used here. They typically consider a dissimilarity measure s that quantifies the dissimilarity (or distance) between two cases. The most similar cases can be found by computing the dissimilarity $s(q, X_i)$ between q and each case X_i in the case base. A common example of dissimilarity measure is the Euclidean distance. For nominal dimensions, one can simply consider a dissimilarity of 0 if the values are equal, and a dissimilarity of 1 if they differ (as we will do in Section 5), or use a more complex semantic distance, *e.g.* based on a formal ontology (as we will do in Section 6).

We name X' the set of cases obtained; X'_1 is q and X'_2 to X'_n are the cases similar to q. In addition, we build a symmetric distance matrix $d = (d_{i,j})_{1 \leq i,j \leq n}$ using the same dissimilarity measure, and we compute similar case weights w_i using equation (1).

Algorithm 1. *fly* and *walk* functions for optimizing angles.

```
Function fly():
    $x' \in \mathbb{R}^d$
    For $1 \le k \le d$:
        $x'_k$= random real number between 0 and $2\pi$
    Return $x'$
Function walk($i$):
    $x' \in \mathbb{R}^d$, $x'_k = x_{ik}$ for $1 \le k \le d$
    $j$ = random integer number between 1 and $n$, $j \ne i$
    $k$ = random integer number between 1 and $d$
    $\Delta = |x_{ik} - x_{jk}|$
    If $\Delta = 0$: $\Delta = 0.001$
    $r$ = random real number between $-1$ and 1
    $x'_k = x'_k + r \times \Delta$
    If not $0 \le x'_k < 2\pi$: $x'_k = x'_k \bmod 2\pi$
    Return $x'$
```

3.3. Generating the polar MDS scatter plot

As explained in Section 2.2, when using MDS to produce scatter plot for CBR, polar coordinates are a better choice than Cartesian coordinates, because they allow preserving the distances between the query and the similar cases. Here, we propose a 2-dimensional approach for MDS in polar coordinates. q is placed at the origin (0, 0). Then, for each other case $i \ge 2$, we need to compute its polar coordinates (l_i, θ_i), where l_i is the distance from the origin and θ_i is the angle to the polar axis. Because the origin is also q, l_i is already available in the distance matrix: $l_i = d_{1,i}$. For angles, we use AFB to find the values of θ_i that minimize the stress function:

$$S_p(d) = \sum_{2<i<j} \frac{(d_{\mathrm{ij}} - \delta_{\mathrm{ij}})^2}{d_{\mathrm{ij}}}$$

where d_{ij} is the distance between point i and j in the n-dimensional distance matrix, and δ_{ij} is the distance between point i and j in the resulting 2-dimensional scatter plot, computed as follows:

$$\delta_{\mathrm{ij}} = \sqrt{(d_{1,i}\cos(\theta_i) - d_{1,j}\cos(\theta_j))^2 + (d_{1,i}\sin(\theta_i) - d_{1,j}\sin(\theta_j))^2}$$

i and j start at 2, because index 1 is q, and distances involving q are fixed. We run AFB using the stress function $S_p(d)$ as the cost function, and the *fly* and *walk* functions given in Algorithm 1. These functions correspond to the one proposed previously for global non-linear optimization [36], but we modified the *walk* functions to take into account the cyclic nature of angles: when the new value is outside the expected range [0, 2π[, we apply a modulo 2π. We test 3000 candidate solutions and we keep the best solution found.

Fig. 4 shows 4 scatter plots generated from the same distance matrix, using PCA, metric MDS, non-metric MDS and polar MDS (the focus being the center white point). In particular, the distance between the white point and the closest red and green points were equal in the distance matrix, and they remain equal on the polar MDS scatter plot, unlike with the three other methods. However, other distances are less well preserved. For example, we see in the first three scatter plots that one of the green points is distant from the other ones. In the polar MDS scatter plot, this information is lost. Consequently, using polar MDS instead of traditional MDS is a trade-off between favoring the distance involving one case and not favoring any distance. In CBR, polar MDS is very interesting because it preserves perfectly the distances between the query and similar cases, allowing a better accuracy when performing visually a WkNN. On the contrary, distances between two similar cases are less interesting in that context, although they can be used to determine whether the similar cases belonging to a given class are close together or not.

3.4. Retaining the two main candidate classes

Only the two best candidate classes are retained for rainbow boxes, to limit visual complexity. We perform a majority voting between similar cases, weighted by case weights w_i. We arbitrarily name these two

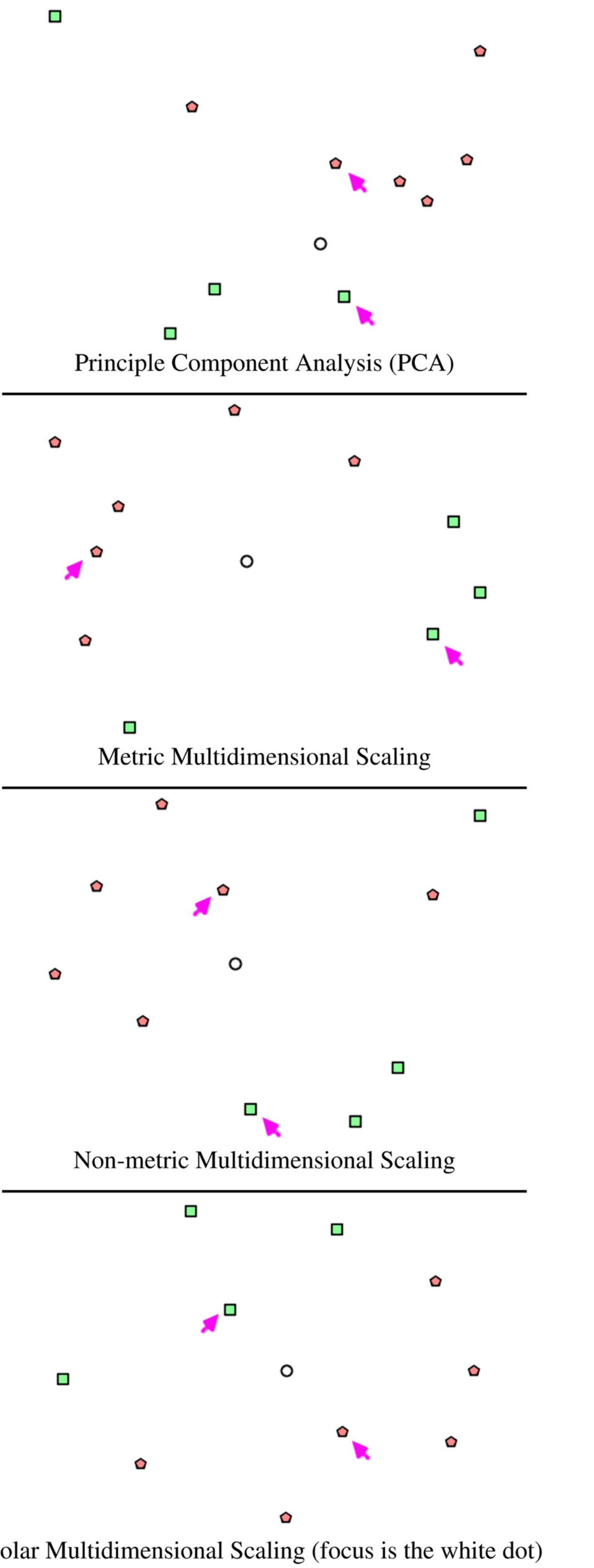


Fig. 4. Scatter plots generated by 4 different methods, for the same dataset. In the original distance matrix, the distances between the white dot and the closest red and green dots (identified by pink arrows) are actually equal. Notice that only polar MDS maintains this equality properly in the scatter plot. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

classes y_1 and y_2, and we extract $X'' = X' \cap (\{q\} \cup y_1 \cup y_2)$, the subsets of cases belonging to one of the retained classes (still including q).

3.5. Generating and selecting boxes

First, numeric values are discretized, using the MDLP (Minimum Description Length Principle) algorithm [38].

A box is defined by a (dimension, value) pair and can be noted Z_v, where Z is the dimension and v the value. It can be formalized as a subset of the cases in the visualization: $Z_v \subseteq X''$. For each boolean dimension Z, one box Z_{True} is generated for the *True* value. For each nominal dimension Z (including discretized integer or real dimensions), one box Z_v is generated for each possible nominal value v.

Many boxes can be generated, if the number of dimension p is high. To avoid overloading the visualization, we select the most interesting boxes, according to Mutual Information (MI). We compute per-box MI (not per-dimension):

$$\mathrm{MI}(Z_v Y) = \sum_{z \in \{Z_v, \bar{Z}_v\}} \sum_{y \in \{y_1, y_2\}} p(z, y) \log\left(\frac{p(z, y)}{p(z)p(y)}\right)$$

where $\bar{Z}_v$ is the subset of cases not belonging to Z_v (and excluding q), i.e. $\bar{Z}_v = X'' \backslash (\{q\} \cup Z_v)$.

Probability $p(\ldots)$ are computed over $X'' \setminus \{q\}$, *i.e.*:

$$p(y_1) = \frac{|y_1|}{|X'' \backslash \{q\}|} \quad \text{and} \quad p(Z_v, y_1) = \frac{|Z_v \cap y_1|}{|X'' \backslash \{q\}|}$$

Finally, we kept the m boxes with the highest MI (m being a parameter of the visualization). Box color C_{Z_v} is the mean of the two classes colors C_{y_1} and C_{y_2}, weighted by the weight w_i of the similar cases involved:

$$C_{Z_v} = C_{y_1} \times \frac{\sum \{w_{2 \le i \le n} | x_i \in Z_v \cap y_1\}}{\sum \{w_{2 \le i \le n} | x_i \in Z_v\}} + C_{y_2} \times \frac{\sum \{w_{2 \le i \le n} | x_i \in Z_v \cap y_2\}}{\sum \{w_{2 \le i \le n} | x_i \in Z_v\}}$$

Boxes are arranged vertically according to their colors.

Algorithm 2. AFB *fly* and *walk* functions for optimizing rainbow boxes column order while grouping similar cases by class. q is the query case, C_1 and C_2 are the sets of similar cases associated with each of the two classes.

```
Function fly():
  $x_1'$ = sequence of the elements in $C_1$, in a random order
  $x_2'$ = sequence of the elements in $C_2$, in a random order
  Return the concatenation of $x_1'$, $q$, $x_2'$

Function walk(i):
  $x_i$ is the position of the bird $i$, i.e. a candidate column order
  $p$ = position of $q$ in sequence $x_i$
  $x_1'$ = elements 1 to $p - 1$ in sequence $x_i$
  $x_2'$ = elements $p + 1$ to $|x_i|$ in sequence $x_i$
  $k$ = random integer number between 0 and 1
  If $k < \frac{|C_1|^2}{|C_1|^2 + |C_2|^2}$:
    $x_1' = \mathrm{2op}(x_1')$
  Else:
    $x_2' = \mathrm{2op}(x_2')$
  Return the concatenation of $x_1'$, $q$, $x_2'$

Function 2op(L):
  $i$ = random integer number between 1 and $|L| - 1$
  $j$ = random integer number between $i + 1$ and $|L|$
  $L'$ = clone of sequence $L$
  Reverse the order of elements between $L_i'$ and $L_j'$
  Return $L'$
```

3.6. Optimizing rainbow boxes

The optimization of rainbow boxes consists of finding the column order that minimizes the number of holes in the boxes. We proposed the AFB metaheuristic [36], able to optimize more than 30 columns in a satisfying time. Here, we add a constraint: similar cases must be grouped by class and q must be in the middle (as shown in Fig. 2). Since some boxes may span across the two groups of similar cases, each group cannot be optimized separately. Therefore, this is a *constrained* combinatorial optimization problem, of complexity $O(n_1! n_2!)$, where n_1 and n_2 are the numbers of similar cases associated with each of the two classes.

We solved this problem using the AFB metaheuristic, with new dedicated *fly* and *walk* functions (Algorithm 2). The *fly* function creates a random column order, made of three parts: the similar cases associated with the first class, the query case, and the similar cases associated with the second class. The two parts with similar cases are randomly shuffled.

The *walk* function is based on the 2-opt local search heuristic [39]. It consists in opening the sequence at two points, and reconnecting it after reversing one of the two parts. The *walk* function includes three steps. First, it extracts from the column order the two subsequences with similar cases. Second, it determines which one of the two subsequences will be modified. The choice is performed randomly, with a relative probability equal to the squared number of cases in each subsequence. This gives a higher chance to modify the longer subsequence, which has more possible orders and thus requires more optimization effort. Third, the chosen subsequence is modified using the 2-opt local search heuristic.

The *cost* function computes the number of holes present in the rainbow boxes, for the given candidate column order. Each hole has a cost that is proportional to the height of the holed box. This favors holes in small boxes, and lets the most important and tallest boxes without holes. The AFB metaheuristic is run until 3000 candidate solutions have been tested, and the best solution found is retained.

3.7. Adding interactivity

We add interactivity for displaying additional information on demand and for connecting together the two parts of the visual interface. Interactivity was added at three levels: (1) when the mouse cursor is over a point in the scatter plot, the corresponding column in rainbow boxes is highlighted by fading other columns, (2) when the mouse cursor is over a column header, the corresponding point in the scatter plot is highlighted by fading other points, (3) when the mouse cursor is over a box, all the scatter plot's points associated with the columns covered by the box are highlighted in a similar way, and a popup label displays the full box label and the exact value of the box dimension for q.

4. Automatic algorithms

In this section, we translate the visual reasoning permitted by the proposed interface into automatic classification algorithms. It provides a better understanding of the interface, although it does not necessarily model the way a human user will use the information and it does not consider the medical knowledge that an expert might use. Finally, it permits determining the best parameter values, and evaluating whether the information displayed could be used for accurate classification.

The interface includes two parts. The scatter plot visualizes the classes the similar cases belong to, and the distances between the similar cases and q. This is very similar to WkNN, so the scatter plot can be formalized by WkNN.

For rainbow boxes, the visual search of the dominant color can be formalized by a rainbow boxes-inspired algorithm (RBIA). It computes a score S_y for each of the two colors (representing the two retained classes y_1 and y_2), and then it classifies q in the class that obtains the highest score. S_y measures the "quantity" of each of the two colors associated with the classes in the visualization. For each box, S_y must take into

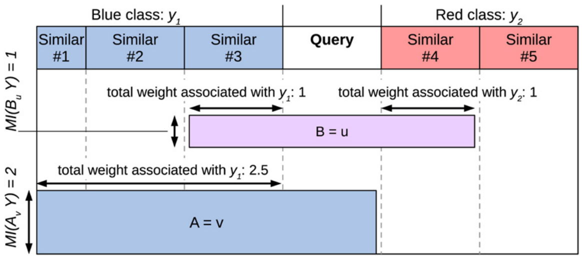


Fig. 5. Example of the computation of scores S_y.

account the box MI (*i.e.* box height), the distribution of the cases in the box between the two classes (*i.e.* box color), and the weight of similar cases (*i.e.* column width). This leads to the formula:

$$S_y = \sum_{Z_v} (\mathrm{MI}(Z_v Y) \times \sum \{w_{2 \le i \le n} | x_i \in Z_v \cap y\})$$

with $y \in \{y_1, y_2\}$. e-Component #1 gives the full RBIA algorithm.

Fig. 5 shows an example of the computation of S_y with 2 boxes. y_1 is the blue class and y_2 the red one. The lower box A_v has a total case weight belonging to class y_1 equal to $\sum \{w_{2 \le i \le n} | x_i \in A_v \cap y_1\} = 2.5$ and 0 for class y_2. The upper box B_u has a total case weight belonging to class y_1 of 1 and 1 for class y_2. The resulting scores are:

$$S_{y_1} = \mathrm{MI}(A_v Y) \times 2.5 + \mathrm{MI}(B_u Y) \times 1 = 6$$

$$S_{y_2} = \mathrm{MI}(A_v Y) \times 0 + \mathrm{MI}(B_u Y) \times 1 = 1$$

Since $S_{y_1} > S_{y_2}$, q is classified in y_1 (blue class).

5. Application to breast cancer machine-learning datasets

Breast cancer is one of the most common type of cancer affecting women in Europe, and it is associated with a high survival rate at 10 years. However, with the development of new therapies and clinical practice guidelines, the management of the disease is becoming more and more complex. In particular, clinicians in the multidisciplinary Breast Units (BUs) have to make several important decisions, including making a diagnostic and prescribing a treatment.

In this section, we apply the proposed approach to machine-learning and simulated datasets, using a very simple CBR setting and Euclidean distance.

5.1. Datasets

We used three public datasets related to breast cancer. The Breast Cancer Wisconsin (BCW) dataset[2] includes 683 cases (after removing 16 cases with missing values), 9 dimensions with integer values ranging from 0 to 10 (computed from a digitized image of fine needle aspirate of breast mass) and 2 classes (whether the diagnostic is benign or malignant). The Mammographic Mass (MM) dataset[3] includes 830 cases, 2 numeric dimensions (age and Breast Imaging Reporting And Data System value, BI-RADS), 3 categorical dimensions (shape, margin and density of the mass) and 2 classes (whether the diagnostic is benign or malignant). The Breast Cancer (BC) dataset[4] includes 286 cases, 4 numeric dimensions (age, tumor size, *etc.*), 4 categorical dimensions (breast quadrant, *etc.*) and 2 classes (whether the cancer is recurrent or not). Second, to demonstrate our visual approach on therapeutic decision with more than two classes, we simulated a dataset (SD) with 4050 cases and 75 dimensions (22 boolean, 14 integer and 39 nominal) and 4 classes, corresponding to the four main categories of treatment for breast cancer: surgery, chemotherapy, radiotherapy and endocrine therapy.

5.2. Examples

Fig. 6 shows the application of our visual interface to a case of the BCW dataset (the query case was extracted from the dataset, but its class was ignored). The "benign" class was associated with yellow, and the "malignant" one with red. We extracted 7 similar cases ($n = 8$, counting q) and we selected at most $m = 11$ boxes (however, fewer boxes are displayed due to the low number of attributes). In Fig. 6, in both scatter plot and rainbow boxes, we can see that 4 similar cases were benign, while 3 were malignant. The scatter plot indicates that the most similar case was malignant (red square is closest to the center of the target). Rainbow boxes headers can be used to visually sum the weights of the similar cases, and conclude that the weight of the malignant cases is higher (this is a visual WkNN). Rainbow boxes show the common characteristics between q and the similar cases. The most informative (bigger) box is "Bare nuclei ≥ 3″, and it includes all, and only, malignant cases. Thus the box is red. Other boxes are smaller, and orange/reddish. Some labels are not visible, but can be obtained through mouse-overing. Here, the visual interface argues strongly in favor of a malignant lesion. Moreover, the physician can mitigate his conclusion depending *e.g.* on the importance he gives to bare nuclei.

Occasionally, the quantitative and qualitative approaches may disagree. In these cases, rainbow boxes provide explanations for the RBIA algorithm and the scatter plot for the WkNN algorithm. Since our system is primarily a visual CBR system, we preferred to let the user have the last word, rather than arbitrarily computing the mean of the results of the two algorithms, or using a voting system.

The SD dataset is more complex because it has 4 classes. We associated a color to each classes of treatment: red for surgery, blue for radiotherapy, green for chemotherapy and yellow for endocrine therapy. Fig. 7 shows an example, with $n = 13$ and $m = 11$. The scatter plot shows that there are 5 similar cases treated by surgery, 4 by radiotherapy, 2 by endocrine therapy and 1 by chemotherapy. The closest case was treated by endocrine therapy. In rainbow boxes, only the two main classes are retained: surgery and endocrine therapy. The red color is dominant, hence, the visual interface advocates for prescribing surgery. However, the "Nuclear grade = Grade1″ criteria may be considered by clinicians, orienting toward endocrine therapy.

5.3. Experiments

We measured the accuracy of the automatic algorithms on the three public datasets, by testing all cases (*i.e. leave-one-out* validation). Fig. 8 shows the results. We tested three algorithms: kNN, WkNN and RBIA (detailed in Section 4). For each algorithm, we selected the value of n (for rainbow boxes) or k (for kNN and WkNN) that yields the best results. RBIA has a better accuracy on two datasets: for BCW, the accuracy is 97.8% (for $n = 8$) *vs* 97.5% (WkNN with $k = 9$, corresponding to $n = 10$). For BC, the accuracy is 77.3% ($n = 18$) *vs* 76.6% (kNN, $k = 5$). On MM, RBIA's accuracy is 80.8% ($n = 21$), *vs* 82.0% (kNN, $k = 7$). Consequently, this shows that the information displayed in rainbow boxes is relevant and allows a good classification, although it is a very small subset of the information available in case database X or even in similar cases X'.

For n above 25, the results evolve as follows. For BCW, the accuracy of the 3 algorithms remains very similar, but the overall results decrease slightly with n for all algorithms. For MM, the algorithms converge to the same accuracy of about 80%. For BC, WkNN accuracy remains high but kNN accuracy decreases, and RBIA accuracy is in-between.

[2] https://archive.ics.uci.edu/ml/datasets/breast+cancer+wisconsin+(original).

[3] https://www.kaggle.com/overratedgman/mammographic-mass-data-set/data.

[4] https://archive.ics.uci.edu/ml/datasets/breast+cancer.

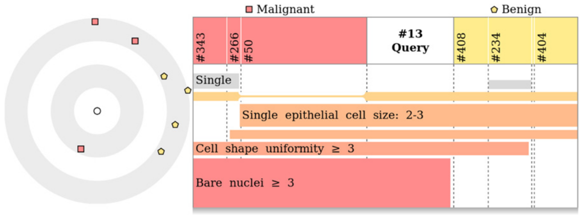


Fig. 6. Visualization of a case in the BCW dataset, with $n = 8$ cases and 7 boxes. Numbers (*e.g.* "#13") are case identifiers.

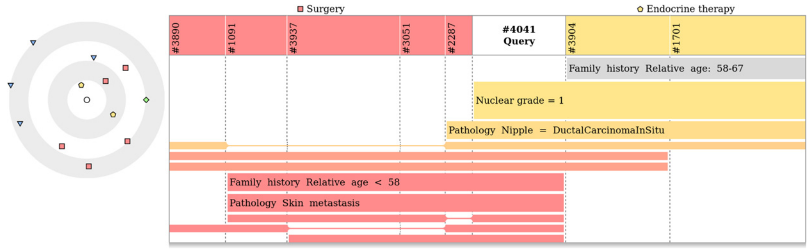

Surgery
Endocrine therapy
#3890
#1091
#3937
#3051
#2287
#4041
Query
#3904
#1701
Family history Relative age: 58-67
Nuclear grade = 1
Pathology Nipple = DuctalCarcinomaInSitu
Family history Relative age < 58
Pathology Skin metastasis

Fig. 7. Screenshot of the visual interface for CBR on the simulated dataset for the therapeutic decision.

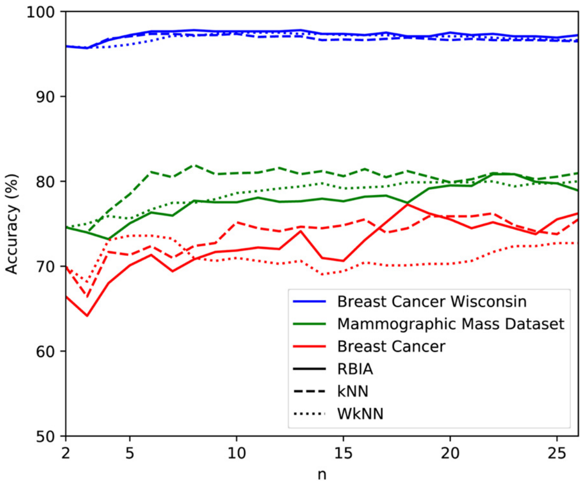


Fig. 8. Accuracy obtained for various values of n on three datasets (identified by colors) and algorithms (identified by line styles). n is the number of cases considered; thus, for kNN and WkNN, $k = n - 1$. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

5.4. User study

The visual interface was presented to medical experts. In the initial interface we presented, all similar cases were shown in rainbow boxes and they were not grouped by class. Experts found this initial interface too complex. Thus we simplified it by limiting rainbow boxes to two classes, as described here (see Section 3.4), leading to a second version of the interface.

This second version was tested with the simulated dataset during a small online user study. 5 cases were presented, each of them corresponding to a query case and 12 similar patients. The cases were of increasing difficulty. The first four were associated with a correct answer, while, in the fifth case, scatter plot and rainbow boxes lead to contradictory conclusions and no correct answer was expected. This case was added to see how users would react in such a situation.

The study was performed using a dynamic website that recorded responses and response times, and users were allowed to send personal comments. The top of the screen displayed the visual interface. The bottom of the screen allowed entering the answer. For each case, the user had to indicate the treatment type he would choose on the basis of the presented data (4 possible values), and his level of confidence in his decision (5-value Likert scale: Not confident, Rather not confident, Mildly confident, Rather confident, Confident, coded as a 1–5 value with 5 being Confident). Statistical analysis was performed using R software version 3.5.1 [40].

We recruited 11 evaluators: 6 physicians working in breast cancer units and 5 physicians working in medical informatics. 3 were female. Mean age was 50.0 years. The mean accuracy of the response was 81.8% (72.7% for case #1, 90.9% for #2, 72.7% for #3 and 90.9% for #4; accuracy was not measured for case #5 since no right answer was expected). Notice that "inaccurate" responses may not be regarded as wrong: as said above, the physician might have mitigated his response with his medical knowledge, *e.g.* giving more importance to a box involving a dimension that he considers as very important. The mean level of confidence was 3.75, with values decreasing with case difficulty (4.1, 4.0, 3.5, 3.4 and 3.4, respectively). Statistical analysis showed that question number (associating with case difficulty) had a significant impact on confidence ($p = 0.0301$, ANalysis Of VAriance). Thus confidence is related to the case difficulty: the visual interface is able to convey the difficulty of the case well. This is important, because CBR may not be well-suited to all cases, and the user must be able to determine whether CBR suits well to a given case, or not.

The analysis of comments showed that experts were enthusiastic with regard to the visual CBR approach. One of them found that it was "like a game". Experts found the qualitative approach interesting as it links system recommendations with patient characteristics in a way that speaks to them. On the other hand, they have sometimes been puzzled, particularly when the scatter plot and rainbow boxes lead to different conclusions (for example if the most similar cases on the scatter plot does not correspond to the treatment toward which rainbow boxes orientate). Analysis of the results of case #3 shows that physicians might give too much importance to the case closest to the query (the closest case was mentioned during the presentation of the visual interface and possibly we insisted too much on it). Certainly, the visual interface requires a short training before being used optimally.

In conclusion to the user study, we have shown that, using the proposed interface, a user was able to perform CBR visually. Moreover, the interface was able to provide the user a good indication of the confidence level of his/her choice. However, these preliminary results must be confirmed in a larger study.

6. Application to real data in breast cancer

The DESIREE project includes several decision-support modules for breast cancer therapy, based on different strategies: implementation of clinical practice guidelines [41], machine-learning, and, of course, CBR [42]. With regard to CBR, breast cancer is known to be particularly difficult because of the attribute type heterogeneity and the difficulty in eliciting attribute weights from experts [43]. In this section, we apply the proposed approach to a real dataset on breast cancer, using a more sophisticated and knowledge-intensive CBR setting than in previous section.

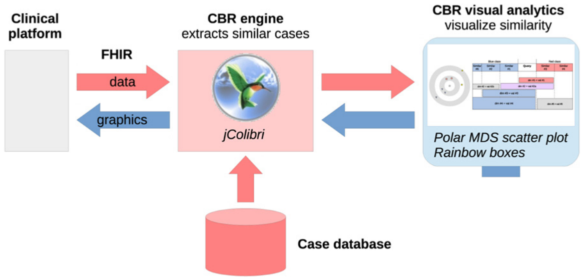


Fig. 9. General architecture of the DESIREE CBR system.

6.1. System architecture and implementation

The clinical platform contains several decision support modules, as well as other modules (*e.g.* for medical imaging). Fig. 9 shows the general CBR system architecture. The CBR module includes three components: (a) the case base, (b) the CBR engine, in charge of retrieving the cases that are the most similar to the query, and (c) the CBR visual analytics component, in charge of producing the visualization. The CBR engine was implemented in Java with JColibri [44]. The CBR visual analytics component was implemented in Python 3. Interoperability between the clinical platform and the CBR module is achieved using the FHIR (Fast Healthcare Interoperability Resources) standard [45] from HL7 (Health Level 7). This standard allows the exchange of health-related messages between medical services, including electronic patient records. In terms of performances, the generation of the visual interface takes less than a second.

6.2. Data extraction

Four hospitals are involved in DESIREE, from France and Spain. Through FHIR, we extracted from the DESIREE Decision Support and Information Management System (DESIMS) an anonymized dataset including 315 patients with breast cancer at the initial stage of the treatment. At this stage, patients are oriented in two “scenarios”: B (chemotherapy, or occasionally endocrine therapy) or D (surgery). In DESIMS, data are structured at 3 levels: patient, side (two sides per patient, *i.e.* left and right) and lesion (zero, one or several lesions per side), according to an OWL domain ontology (described in [46,47]). Data were extracted using the Owlready2 [48,49] ontology-oriented programming module for Python, and treated as follows. Per-side and per-lesion dimensions were aggregated by retained only the worst value available. For example, if a patient has several lesions, the “tumor size” dimension is actually the size of the largest lesion. Missing data were fixed by replacing them with the most frequent value (for categorical and boolean dimensions) or by the mean of the values observed (for integer and real dimensions). The “number of abortions” dimension was an exception: since clinicians filled this field only when non-null, missing values were replaced by 0. The resulting dataset includes 51 dimensions (22 Boolean, 15 integer, 1 real and 13 categorical dimensions) and two classes (118 patients in B/chemotherapy and 197 in D/surgery).

6.3. CBR setting

Dimensions were initially selected and weighted using the Weka BestFirst and Ranker algorithms. Then the result was discussed with an expert (BS) that worked on therapeutic guidelines for breast cancer. Following her recommendations, a new attribute was added to the selection (Histologic type) and the weights of three attributes were increased. Table 1 shows the 13 selected dimensions and their weights.

Table 1
The dimensions selected and their weights.

Dimension	Type	Weight
Stage	Categorical	0.255598
TNM cM	Categorical	0.041492
TNM cN	Categorical	0.126464
TNM cT	Categorical	0.274324
Ki67 result	Integer	0.08387
PR result	Integer	0.075758
Suspicion of invasion	Boolean	0.059697
Tumor size	Real	0.112173
Clinical lymph nodes	Boolean	0.079908
Molecular subtype	Categorical	0.064898
Histologic type[a]	Categorical	0.2[a]
ER status	Boolean	0.15[a]
SBR grade	Integer	0.15[a]

TNM is a classification of malignant tumors; T represents the size of the original tumor, N for nearby (regional) lymph nodes involved and M is for metastasis. Ki67 is a cellular proliferation marker. PR stands for progesterone receptors. ER stands for estrogen receptors. SBR is a categorization of the aggressiveness of a cancer.

[a] Elements introduced or modified following the expert recommendations.

We used as dissimilarity measure *s* the Euclidean distance, considering all retained dimensions, each being weighted by its weight w_j

$$s(a,\ b) = \sqrt{\sum_{j=0}^{p} w_j \times \mathrm{dist}(a_j,\ b_j)^2}$$

Here, *dist*() is a function that computes the distance between two values of a given dimension. Three *dist*() functions are used, depending of the type of the dimension: boolean, numeric (integer or real) or categorical. For boolean dimensions, it simply compare the two values. For numeric dimensions, it computes a classical Euclidean distance. For categorical dimensions, we proposed a semantic distance (described in [50,42]), based on the hierarchical placement of concepts in the domain ontology. The 3 *dist*() functions are as follows:

$$\mathrm{dist}_{\mathrm{boolean}}(t_1,\ t_2) = \begin{cases} 1 & \text{if } t_1 \neq t_2 \\ 0 & \text{if } t_1 = t_2 \end{cases}$$

$$\mathrm{dist}_{\mathrm{numeric}}(t_1,\ t_2) = \frac{|t_1 - t_2|}{\max(t) - \min(t)}$$

$$\mathrm{dist}_{\mathrm{sem}}(t_1,\ t_2) = 1 - \frac{|\mathrm{super}(t_1) \cap \mathrm{super}(t_2)|}{\sqrt{|\mathrm{super}(t_1)|} \times \sqrt{|\mathrm{super}(t_2)|}}$$

where *super*(*t*) is a function that returns, for a given individual *t*, the set of classes it belongs to in the ontology, including superclasses up to the root class for this dimension. For example, considering the following simple ontology:

$$\mathrm{NonInvCarc} \sqsubseteq \mathrm{Carc}$$
$$\mathrm{InvCarc} \sqsubseteq \mathrm{Carc}$$
$$\mathrm{InvDuctalCarc} \sqsubseteq \mathrm{InvCarc}$$

we have:

$$\mathrm{super}(t \in \mathrm{Carc}) = \{\mathrm{Carc}\}$$
$$\mathrm{super}(t \in \mathrm{NonInvCarc}) = \{\mathrm{NonInvCarc},\ \mathrm{Carc}\}$$
$$\mathrm{super}(t \in \mathrm{InvCarc}) = \{\mathrm{InvCarc},\ \mathrm{Carc}\}$$
$$\mathrm{super}(t \in \mathrm{InvDuctalCarc}) = \{\mathrm{InvDuctalCarc},\ \mathrm{InvCarc},\ \mathrm{Carc}\}$$

and we obtain:

$$\mathrm{dist}_{\mathrm{sem}}(t_1 \in \mathrm{InvCarc},\ t_2 \in \mathrm{InvCarc}) = 0$$
$$\mathrm{dist}_{\mathrm{sem}}(t_1 \in \mathrm{InvDuctalCarc},\ t_2 \in \mathrm{InvCarc}) = 0.18$$
$$\mathrm{dist}_{\mathrm{sem}}(t_1 \in \mathrm{InvDuctalCarc},\ t_2 \in \mathrm{Carc}) = 0.42$$
$$\mathrm{dist}_{\mathrm{sem}}(t_1 \in \mathrm{InvCarc},\ t_2 \in \mathrm{NonInvCarc}) = 0.5$$
$$\mathrm{dist}_{\mathrm{sem}}(t_1 \in \mathrm{InvDuctalCarc},\ t_2 \in \mathrm{NonInvCarc}) = 0.59$$

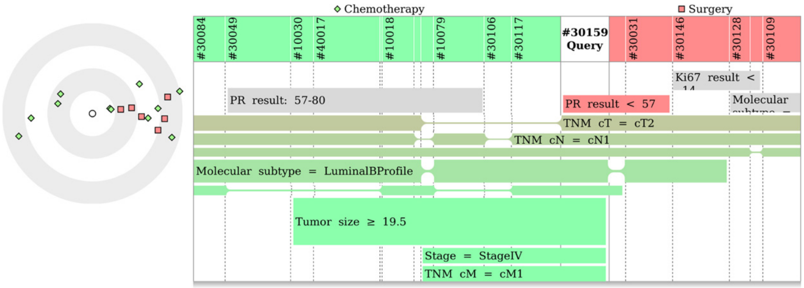


Fig. 10. Screenshot of the visual interface for CBR on the real dataset for initial therapeutic decision in breast cancer.

6.4. Results

Using WkNN, the best classification accuracy we obtained was 80.3%, for $k = 18$ (*i.e.* $n = 17$). Fig. 10 shows a screenshot of the proposed interface on the real breast cancer dataset. Here, the horizontal length covered by green column headers is longer than the one of red header, thus WkNN recommend the green treatment, *i.e.* chemotherapy/endocrine therapy (scenario B). Rainbow boxes provide arguments in favor of this treatment. The main argument is a high tumor size. On the contrary, the low “PR result” value is an argument that a clinician might consider for prescribing surgery.

7. Discussion

We proposed a visual Case-Based Reasoning method for classifying a new query case in a small number (2–10) of classes. It relies on a visual interface that translates the CBR problem into a question of “color dominance” and combines two complementary approaches: a polar-MDS scatter plot showing quantitative similarities and rainbow boxes showing qualitative similarities. We also formalized the expected visual reasoning using algorithms. In particular, rainbow boxes were able to explain visually *why* the similar cases are similar to the query case, and on which dimensions and values the similarity holds. Since boxes show dimension names and their associated values, they are easily understandable for a human user.

We applied the proposed method to breast cancer, and we tested it on three public datasets for diagnostic decision and one simulated and one real dataset for therapeutic decision. Domain experts were interested in this visual approach.

In the visual interface, we proposed an MDS method in polar coordinates that generates a scatter plot centered on the query. It preserves all distances involving the query, and restricts the information loss on the other distances. It is similar to previously mentioned Klawonn et al. [26] case-centered MDS, but in two dimensions instead of three, with a simpler algorithm. To our knowledge, this is the first CBR polar MDS in two dimensions: while polar MDS is well-suited to CBR, it has actually been rarely used in this context.

We also proposed a qualitative approach, using rainbow boxes. Although we used quantitative attribute distances to build rainbow boxes, they display textual box labels (*e.g.* “menopause = premenopausal”) in addition to MI. These labels are qualitative by nature. Consequently, from a visualization point of view, rainbow boxes can be considered as a qualitative approach, and this is why we described them as such. On the contrary, the scatter plot displays only quantitative distances. We also extended rainbow boxes with a new visual variable: column width.

We used Mutual Information (MI) for selecting and weighting boxes. MI is commonly used for feature selection [51] and for CBR [52]. However, we used MI at the local neighborhood of the query case, and not globally. In the literature, local similarity refers initially to “local in feature-space”, *i.e.* a similarity restricted to a single attribute [53]. However, recent works focused on the definition of “local similarity in case-space”, *i.e.* a similarity restricted to a subset of the case base. Liu et al. [54] proposed a local use of MI for feature selection. Badra [55] used co-variations to “express a local coïncidence of values of two properties” and then use these co-variations as a dissimilarity measure to perform similarity-based reasoning. Zabkar et al. [56] proposed a local similarity-based approach for searching for qualitative relations in categorical domains. Zabkar et al. consider all cases at the beginning and subdivide them in subsets recursively, building a tree. Thus, excepted at the root of the tree, the qualitative relations found are local to a given subset of the cases. Similarly, in this study, we worked at the local neighborhood of the query case, determined using a global similarity measure, and then we identified local similarities using MI; these similarities are valid in the neighborhood of the query but might not be true at the global level.

Our interface relies heavily on color for identifying the various classes (*e.g.* treatment types). However, a color-blind-friendly version of the interface could easily be set up. The scatter plot uses various shapes in addition to colors. In rainbow boxes, only two classes are compared, so colors are not required and we could use light and dark gray for distinguishing the left and right classes.

We proposed a formalization of the two parts of the visual interface as automatic algorithms. We showed that the quantitative part was similar to a WkNN algorithm. We proposed an algorithm for the qualitative part (RBIA) and we showed that the information displayed on the qualitative part, although very partial since limited to a few (dimension, value) pairs, allowed a good classification accuracy, comparable to kNN and WkNN, while being visually explainable. However, RBIA seems highly dependent on the value of the parameter *n*, especially for the BC dataset: while this algorithm produced the highest accuracy, its accuracy is actually above those of kNN and WkNN for only two values of *n*. Consequently, *n* must be chosen carefully.

In our opinion, the association of an automatic algorithm with a visual interface is a “win-win” methodology: the visual interface allows explaining the reasoning process to the user, who can enrich it by considering his personal knowledge, and the automatic algorithm allows a better formalization of the visual reasoning process. To our knowledge, this is an original approach.

We described two optimization problems required for generating the visual interface: optimizing angles in polar MDS and optimizing column order in rainbow boxes, under constraint. These two problems are very different: unconstrained global non-linear optimization *vs* constrained combinatorial optimization. However, we were able to solve both problems using the same metaheuristic, AFB. This highlights the adaptability of AFB.

We achieved a classification accuracy of 80.3% on the real dataset; this result is encouraging but still has some margin or improvement. In particular, the case base was limited to 315 patients. We are still collecting real cases, and a larger case base might improve the results. In particular, we were unable to distinguish the two treatments in scenario B (chemotherapy and endocrine therapy) because of the very small number (6) of endocrine therapy in the dataset.

8. Conclusion

We proposed a visual and explainable CBR system combining quantitative and qualitative approaches. This method was tested on three public datasets, a simulated and a real datasets related to breast cancer. The first perspective of this work is the clinical validation of the visual CBR approach, including a more comprehensive user study in controlled conditions. A second perspective is to improve the CBR system, for instance by considering covariations [57,55] (*i.e.* correlated variations of two variables, such as "recurrence of cancer increases when the size of the tumor increases") for identifying qualitative similarities. A third perspective is to develop a similar interface for analyzing medical datasets, but aimed at knowledge discovery rather than CBR.

Conflict of interest

None declared.

Acknowledgements

This work was funded by the European Union's Horizon 2020 research and innovation program, through the DESIREE project, H2020 PHC-30-2015, grant agreement no. 690238. The funding source had no involvement in the study design, the collection, analysis and interpretation of data, the writing of the manuscript and in the decision to submit the article for publication.

Appendix A. Supplementary data

Supplementary data associated with this article can be found, in the online version, at https://doi.org/10.1016/j.artmed.2019.01.001.

References

[1] de Mantaras RL. Case-based reasoning. Machine learning and its applications. Springer Berlin Heidelberg; 2001. p. 127–45.

[2] Aamodt A, Plaza E. Case-based reasoning: foundational issues, methodological variations, and system approaches. AI Commun 1994;7(1):39–59.

[3] Choudhury N, Begum SA. A survey on case-based reasoning in medicine. Int J Adv Comput Sci Appl 2016;7(8):136–44.

[4] Damiani G, Pinnarelli L, Colosimo SC, Almiento R, Sicuro L, Galasso R, et al. The effectiveness of computerized clinical guidelines in the process of care: a systematic review. BMC Health Serv Res 2010;10:2. https://doi.org/10.1186/1472-6963-10-2.

[5] Bouaud J, Séroussi B, Falcoff H, Julien J, Simon C, Denké DL. Consequences of the verification of completeness in clinical practice guideline modeling: a theoretical and empirical study with hypertension. AMIA symposium 2009 2009:60–4.

[6] Bichindaritz I, Kansu E, Sullivan KM. Case-based reasoning in CARE-PARTNER: gathering evidence for evidence-based medical practice. European workshop on advances in case-based reasoning 1998:334–45.

[7] Somashekhar SP, Sepúlveda MJ, Puglielli S, Norden AD, Shortliffe EH, Rohit Kumar C, et al. Watson for oncology and breast cancer treatment recommendations: agreement with an expert multidisciplinary tumor board. Ann Oncol 2018;29(2):418–23.

[8] Moxey A, Robertson J, Newby D, Hains I, Williamson M, Pearson SA. Computerized clinical decision support for prescribing: provision does not guarantee uptake. J Am Med Inform Assoc 2010;17(1):25–33. https://doi.org/10.1197/jamia.M3170.

[9] Liberati EG, Ruggiero F, Galuppo L, Gorli M, González-Lorenzo M, Maraldi M, et al. What hinders the uptake of computerized decision support systems in hospitals? A qualitative study and framework for implementation. Implement Sci 2017;12:113.

[10] Teach RL, Shortliffe EH. An analysis of physician attitudes regarding computer-based clinical consultation systems. Comput Biomed Res 1981;14(6):542–58.

[11] Villani C, Schoenauer M, Bonnet Y, Berthet C, Cornut AC, Levin F, et al. Donner un sens á l'intelligence artificielle: Pour une stratégie nationale et européenne. 2018.

[12] Biran O, Cotton C. Explanation and justification in machine learning: a survey. Workshop on Explainable AI (XAI) 2017:8–13.

[13] Lane HC, Core MG, Van Lent M, Solomon S, Gomboc D. Explainable artificial intelligence for training and tutoring. 2005.

[14] Van Lent M, Fisher W, Mancuso M. An explainable artificial intelligence system for small-unit tactical behavior. Proceedings of the national conference on artificial intelligence 2004:900–7.

[15] Lamy JB, Berthelot H, Capron C, Favre M. Rainbow boxes: a new technique for overlapping set visualization and two applications in the biomedical domain. J Vis Lang Comput 2017;43:71–82.

[16] Lamy JB, Berthelot H, Favre M, Ugon A, Duclos C, Venot A. Using visual analytics for presenting comparative information on new drugs. J Biomed Inform 2017;71:58–69.

[17] Lamy JB, Tsopra R. Translating visually the reasoning of a perceptron: the weighted rainbow boxes technique and an application in antibiotherapy. International conference Information Visualisation (iV). 2017. p. 256–61.

[18] Lamy JB, Sekar B, Guezennec G, Bouaud J, Séroussi B. Raisonnement á partir de cas visuel: méthodes et application au traitement du cancer du sein. Atelier Visualisation d'informations, Interaction, et Fouille de données (VIF) 2018.

[19] Rani P, Vashishtha J. An appraise of KNN to the perfection. Int J Comput Appl 2017;170(2):13–7.

[20] Dudani SA. The distance-weighted *k*-nearest-neighbor rule. IEEE Trans Syst Man Cybern 1976(4):325–7.

[21] Gou J, Du L, Zhang Y, Xiong T. A new distance-weighted *k*-nearest neighbor classifier. J Inf Comput Sci 2012;9(6):1429–36.

[22] Cunningham P, Doyle D, Loughrey J. An evaluation of the usefulness of case-based explanation. 5th international conference on case-based reasoning 2003:122–30.

[23] Mac Namee B, Delany SJ. CBTV: visualising case bases for similarity measure design and selection. International conference on case-based reasoning 2010:213–27.

[24] Zhu GN, Hu J, Qi J, Ma J, Peng YH. An integrated feature selection and cluster analysis techniques for case-based reasoning. Eng Appl Artif Intell 2015;39:14–22.

[25] Massie S, Craw S, Wiratunga N. Visualisation of case-base reasoning for explanation. Proceedings of the ECCBR 2004 workshops. 2004. p. 135–44.

[26] Klawonn F, Lechner W, Grigull L. Case-centred multidimensional scaling for classification visualisation in medical diagnosis. International conference on health information science 2013:137–48.

[27] Medjahed SA, Saadi TA, Benyettou A. Breast cancer diagnosis by using *k*-Nearest Neighbor with different distances and classification rules. Int J Comput Appl 2013;62(1).

[28] Kohonen T. Self-organizing maps. Berlin: Springer-Verlag; 1995.

[29] Jeong DH, Ziemkiewicz C, Fisher B, Ribarsky W, Chang R. iPCA: an interactive system for PCA-based visual analytics. Comput Graph Forum 2009;28(3):767–74.

[30] Borg I, Groenen P. Modern multidimensional scaling: theory and applications. Springer Science & Business Media; 2013.

[31] Bertini E, Tatu A, Keim D. Quality metrics in high-dimensional data visualization: an overview and systematization. IEEE Trans Vis Comput Graph 2011;17(12):2203–12.

[32] Li JX. Visualization of high-dimensional data with relational perspective map. Inf Vis 2004;3:49–59.

[33] Rehm F, Klawonn F, Kruse R. POLARMAP – efficient visualisation of high dimensional data. International conference on information visualization 2006.

[34] Alsallakh B, Micallef L, Aigner W, Hauser H, Miksch S, Rodgers P. The state-of-the-art of set visualization. Comput Graph Forum 2016;35(1):234–60.

[35] Yang XS. Nature-inspired metaheuristic algorithms. 2nd ed. Luniver Press; 2010.

[36] Lamy JB. Artificial Feeding Birds (AFB): a new metaheuristic inspired by the behavior of pigeons. Advances in nature-inspired computing and applications. Springer; 2019. p. 43–60.

[37] Tsopra R, Lamy JB, Sedki K. Using preference learning for detecting inconsistencies in clinical practice guidelines: methods and application to antibiotherapy. Artif Intell Med 2018;89:24–33.

[38] Fayyad UM, Irani KB. Multi-interval discretization of continuous-valued attributes for classification learning. Proceedings of the international joint conference on uncertainty in AI 1993:1022–7.

[39] Marinakis Y. Heuristic and metaheuristic algorithms for the traveling salesman problem. Encyclopedia of optimization. Springer-Verlag; 2009. p. 1498–506.

[40] R team development core. R: a language and environment for statistical computing. 2008. Vienna, Austria.

[41] Séroussi B, Guézennec G, Lamy JB, Muro N, Larburu N, Sekar BD, et al. Reconciliation of multiple guidelines for decision support: a case study on the multidisciplinary management of breast cancer within the DESIREE project. Proc AMIA annual symposium 2017.

[42] Sekar B, Lamy JB, Larburu N, Seroussi B, Guézennec G, Bouaud J, et al. Case-based decision support system for breast cancer management. Int J Comput Intell Syst 2018;12(1):28–38.

[43] Gu D, Liang C, Zhao H. A case-based reasoning system based on weighted heterogeneous value distance metric for breast cancer diagnosis. Artif Intell Med 2017;77:31–47.

[44] Recio-García JA, González-Calero PA, Díaz-Agudo B. jcolibri2: a framework for building Case-based reasoning systems. Sci Comput Program 2014;79:126–45.

[45] Bender D, Sartipi K. HL7 FHIR: an agile and RESTful approach to healthcare information exchange. IEEE 26th international symposium on Computer-Based Medical Systems (CBMS) 2013:326–31.

[46] Bouaud J, Guézennec G, Séroussi B. Combining the generic entity-attribute-value model and terminological models into a common ontology to enable data integration and decision support. Stud Health Technol Inform 2018;247:541–5.

[47] Sadki F, Bouaud J, Guézennec G, Séroussi B. Semantically structured web form and data storage: a generic ontology-driven approach applied to breast cancer. Stud Health Technol Inform 2018;255:205–9.

[48] Lamy JB. Ontology-oriented programming for biomedical informatics. Stud Health Technol Inform 2016;221:64–8.

[49] Lamy JB. Owlready: ontology-oriented programming in Python with automatic classification and high level constructs for biomedical ontologies. Artif Intell Med 2017;80:11–28.

[50] Sekar B, Lamy JB, Muro N, Pinedo AU, Seroussi B, Larburu N, et al. Intelligent clinical decision support systems for patient-centered healthcare in breast cancer oncology. Workshop on decision support systems for oncology at IEEE Healthcom 2018.

[51] Wei M, Chow TW, Chan RH. Heterogeneous feature subset selection using mutual information-based feature transformation. Neurocomputing 2015;168:706–18.
[52] Han M, Cao Z, Li Y. An improved case-based reasoning method based on fuzzy clustering and mutual information. Intelligent Control and Information Processing (ICICIP) 2014:293–300.
[53] Burkhard HD. Case completion and similarity in case-based reasoning. Int J Comput Sci Inf Syst 2004;1(2).
[54] Liu Q, Xiao J, Zhu H. Feature selection for software effort estimation with localized neighborhood mutual information. Clust Comput 2018:1–9. (in press).
[55] Badra F. Reasoning with co-variations. 17th international conference on Artificial Intelligence: Methodology, Systems, Applications (AIMSA 2016). 2016.
[56] Zabkar J, Bratko I, Demsar J. Extracting qualitative relations from categorical data. Artif Intell 2016;239:54–69.
[57] Badra F. A language of case differences. Computational analogy workshop, international conference on case-based reasoning. 2017.