Info Vis

Article

Information Visualization
2026, Vol. 25(2) 156–167

DOI: 10.1177/14738716251381651
journals.sagepub.com/home/ivi
Sage

Interactively visualizing pattern-based summaries of fuzzy tensors

Victor Henrique Silva Ribeiro[1] and Loïc Cerf[1]

Abstract

Many datasets are n-way fuzzy tensors, that is, n-dimensional tables where every cell grades in $[0, 1]$ the truth of a statement instantiated by the n elements indexing the cell. In the special case where $n = 2$, fuzzy matrices encode to what extent objects (the rows) have attributes (the columns). Boolean tensors are another special case, in which the statement is always either false (0) or true (1). Theoretically simple pattern-based models exist for n-way Boolean or fuzzy tensors, and so do algorithms to discover sets of patterns well summarizing the tensors, according to these models. However, intuitively understanding such summaries has remained difficult. Tools specifically assisting data scientists in that endeavor are lacking. This article addresses that issue. It proposes the first interactive visualization for pattern-based summaries of fuzzy tensors. Pieces of information that are relevant to the interpretation of a summary are extracted from the study of the disjunctive box cluster model. They are turned into visual objects, attributes or interaction techniques. Special attention is given to allowing the exploration of summaries with many large patterns, as shown with the analysis of 55 patterns involving thousands of elements and summarizing a $12 \times 170\,670 \times 29$ real-world tensor.

Keywords

interactive visualization, multidimensional, pattern, fuzzy tensor, disjunctive box cluster model

Introduction

Many datasets of interest relate two or more *entity types*. For example, a data scientist may study the influence of Twitter users (first entity type) writing about Brazilian soccer teams (second entity type) week after week (third entity type). Given a Twitter user u, a soccer team t and a week w, the affirmation "u was influential when she wrote about t during w" may be graded on a Likert scale, for example, with either "Strongly disagree" or "Disagree" or "Neither agree nor disagree" or "Agree" or "Strongly agree," which can be respectively mapped to 0, 0.25, 0.5, 0.75, and 1. Alternatively, the total number of retweets u received whenever she mentioned t during w may be turned into a number in $[0, 1]$, for example, through the process described in caption of Figure 1.

In both cases, the resulting number in $[0, 1]$ is a *membership degree* in the framework of the fuzzy set theory. It grades to what extent the 3-tuple (u, t, w) belongs to the ternary relation indicating whether "u was influential when she wrote about t during w". Given a set of users, a set of teams and a set of weeks, a 3-way fuzzy tensor associates every possible combination of a user, a team and a week with a membership degree. More generally, given $n \in \mathbb{N}$ finite sets called *dimensions*, an *n-way fuzzy tensor* maps every n-tuple of their Cartesian product to a value in $[0, 1]$. Boolean tensors, whose values are in $\{0, 1\}$, are a special case. Although a dimension is temporal in the example, this

[1]Computer Science, Universidade Federal de Minas Gerais, Belo Horizonte, Brazil

Corresponding author:
Loïc Cerf, Computer Science, Universidade Federal de Minas Gerais, Predio do ICEx - Sala 4310, Av. António Carlos, 6627 - Pampulha, Belo Horizonte, Minas Gerais 31270-901, Brazil.
Email: lcerf@dcc.ufmg.br

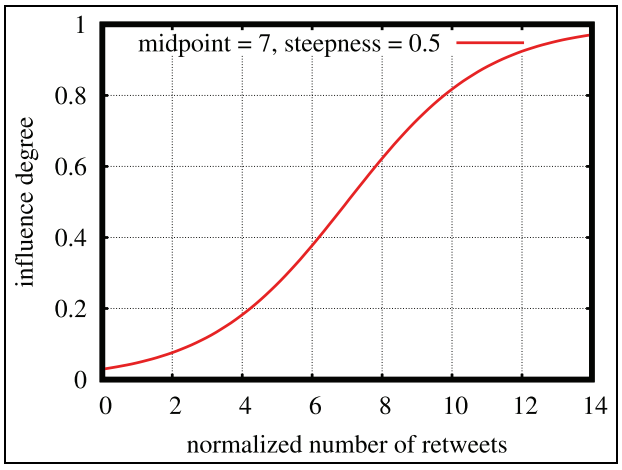

Figure 1. Numbers of retweets are normalized so that summing those relating to any team gives the average number of retweets per team. The depicted logistic function turns them into membership degrees.

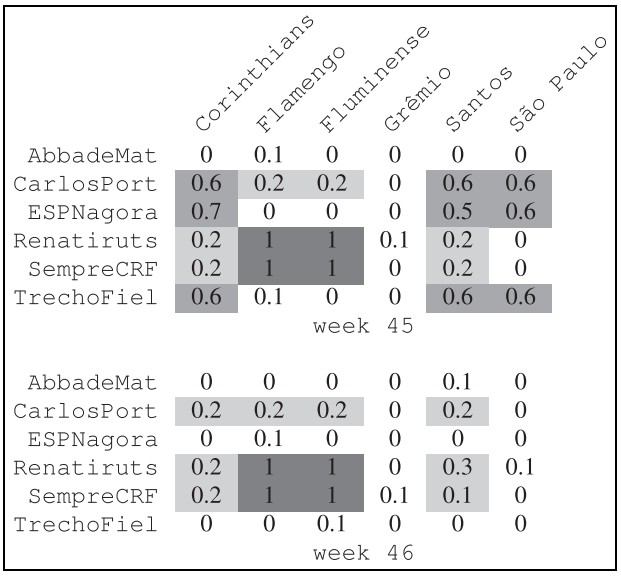

Figure 2. A fuzzy (sub-)tensor whose dimensions contain 2 weeks (the tables), 6 Twitter users (the rows), and six soccer teams (the columns). Three patterns in it are emphasized: {week 45, week 46}×{Renatiruts, SempreCRF}×{Flamengo, Fluminense}, {week 45, week 46}×{CarlosPort, Renatiruts, SempreCRF}×{Corinthians, Flamengo, Fluminense, Santos} and {week 45}×{CarlosPort, ESPNagora, TrechoFiel}×{Corinthians, Santos, São Paulo}. Their respective densities are 1, 0.5, and 0.6. The second pattern includes the first pattern and partially overlaps with the third pattern.

work treats it as a set (of weeks) with no meaningful order.

Unless a fuzzy tensor is tiny, as in Figure 2, an analyst cannot read all its membership degrees. She needs a summary. Pattern-based summaries are usually easy to interpret. In the context of this article, a *pattern* is the Cartesian product of subsets of each of the n dimensions. In the example and in Figure 2, a subset of weeks, a subset of Twitter users and a subset of soccer teams together define a pattern: it is the set of 3-tuples in their Cartesian product.

Showing to an analyst a single pattern can tell her at once that every n-tuple in it is associated with the value 1. Showing several patterns may support a good reconstruction of a whole Boolean tensor: any n-tuple is mapped to 1 if it is in at least one pattern, 0 otherwise. The popular Boolean CANDECOMP/PARAFAC (CP) decomposition[1–8] uses that simple model. It summarizes a Boolean tensor with a few patterns heuristically minimizing the number of incorrectly reconstructed values. That is equivalent to minimizing the Frobenius distance between the input tensor and the reconstructed one. Indeed, with values only in $\{0, 1\}$, any absolute difference is either $0 = 0^2$, for a correct reconstruction, or $1 = 1^2$, if incorrect. To be worthwhile in a summary, a pattern must correspond, in the summarized tensor, to a sub-tensor where most (but not necessarily all) values are 1. Informing the actual proportion together with the pattern is useful: mapping its n-tuples to that proportion (rather than 1) further minimizes the Frobenius distance, that is, better models the related sub-tensor.

The disjunctive box cluster model considers that more accurate reconstruction of the Boolean tensor. At the intersection of several patterns of the summary, the estimated membership degree is the highest proportion of 1 among those of the overlapping patterns. Fuzzy logic justifies the aggregation with max: it is the Zadeh operator corresponding to the logical disjunction (OR) in the Boolean logic that the Boolean CP decomposition uses. In fact, the disjunctive box cluster model applies not only to Boolean tensors but more generally to fuzzy tensors,[9,10,11] including fuzzy matrices.[12–14] The proportions of 1 then become *densities*, that is, arithmetic means of the membership degrees in the sub-tensors that correspond to the patterns.

For instance, the density of the second pattern in caption of Figure 2 is $\frac{12}{24} = 0.5$. That pattern and its density are interpreted in this way: the Twitter users CarlosPort, Renatiruts and SempreCRF were moderately influential (a 0.5 influence degree on average) when they wrote about the Brazilian soccer teams Corinthians, Flamengo, Fluminense, and Santos during week 45 and week 46. Figure 3 shows the tensor reconstructed from three patterns in Figure 2. In the pattern of density 0.5, every 3-tuple is associated with 0.5, unless the 3-tuple belongs as well to a denser

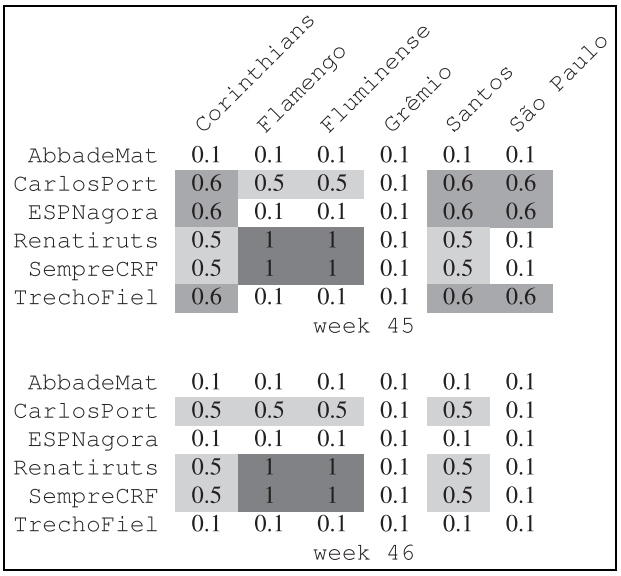

Figure 3. Approximate reconstruction of the (sub-)tensor in Figure 2 from the summary consisting of the three patterns it emphasizes. In any cell, the value is the density of the densest pattern covering the cell or $\hat{\lambda}_0 = 0.1$ if no pattern covers it.

pattern. Whoever interprets a pattern and its density should therefore be aware of its intersections with other patterns. This article identifies such pieces of information that are worth bringing to the attention of the data scientist who summarized a fuzzy tensor using the disjunctive box cluster model.

Given such a tensor and patterns in it, here are tasks that are useful to their analysis:

- T1 Identify the patterns that contribute the most to summarizing the tensor;
- T2 Determine the involvement of specific elements in a (potentially large) pattern, to interpret it;
- T3 Navigate to related patterns, including by similarity;
- T4 Filter the patterns to get a coarser summary that is faster to interpret.

The tasks are generically phrased so that they apply whatever the model. The next section discusses related work with them. Then, the disjunctive box cluster model is formally presented and the concepts above (the *contribution* to the summary, the *similarity*, etc.) are precisely defined in light of that model. The section after turns them into visual objects, attributes, and interaction techniques that ease the interpretation in a visualization tool. To the best of our knowledge, that tool, under a free software license, is the first ever for visualizing pattern-based summaries of fuzzy (or even Boolean) tensors. Its usage is exemplified on a 55-pattern summary of a $12 \times 170\,670 \times 29$ real-world tensor. A conclusion ends the article.

Related work

The following sections propose an interactive visualization that relies on the disjunctive box cluster model,[15] where patterns summarize fuzzy tensors. The 15 references so far present methods to compute such summaries. None of them are detailed here, because the present work starts where their executions end: with a set of patterns to interpret. Although the reconstruction of a tensor from a Boolean CP decomposition ignores the densities of the patterns, the proposed visualization applies. Indeed, as explained in the introduction, the disjunctive box cluster model, using the same patterns, more accurately reconstructs the tensor according to the Frobenius distance. To the best of our knowledge, there exists no previous research on the visualization of patterns in n-way fuzzy tensors. The closest works tackle two special cases: visualizing biclusters, when $n = 2$, or particular patterns in 3-way Boolean tensors.

Visualizing biclusters

Given patterns in a matrix, heuristics[16–20] have been proposed to reorder its rows and columns. If possible, the rows/columns involved in a pattern are put one after another. Nevertheless, that may be impossible. For example, three overlapping patterns involving the sets of rows $\{a, b\}$, $\{a, c\}$, and $\{b, c\}$ impose three adjacency constraints (a next to b, a next to c, and b next to c) that no permutation of $\{a, b, c\}$ satisfies. In such a situation, either some rows/columns are duplicated[16] or some patterns are discontiguously represented.[17–21]

The reordered matrix is directly visualized as a grid heat map[16–19] or the sorted rows and columns are seen as two sets of vertices and edge bundles in the visualization of the bipartite graph stand for the patterns.[20,21] Either way, the visualization does not scale to matrices with thousands of rows or columns (T2) and there is no obvious generalization to n-way tensors with $n \geqslant 3$. Moreover, the patterns contributing the most to summarizing the matrix are not easily identified (T1) and there is no way to get a coarser summary that would be faster to interpret (T4).

Concatenating reordered matrices (e.g., Wu et al.[22]) or bundling edges of a multipartite graph (e.g., Zhao et al.[23] and Sun et al.[24]) supports the visualization of *bicluster chains*, that is, patterns in several matrices sharing dimensions. If every edge bundle is a rectangle whose base represents the $[0, 1]$ interval, the membership degrees in the fuzzy sub-matrix defined by the pattern can be marked.[23] Multidimensional

scaling can position the edge bundles on the screen too,[24] to support a similarity-based navigation between patterns (T3). This article reuses the latter idea, but with a finer and better-founded definition of the (dis)similarity. Other existing tools, Furby[25] for example, have inspired the drawing of segments between patterns that partially overlap, another type of relation (T3).

Visualizing triconcepts

Triconcepts in 3-way Boolean tensors generalize formal concepts in Boolean matrices, that is, a *triconcept* is a pattern of density 1 that no other pattern of density 1 strictly includes. There can exist up to $2^{\min\{m+n, m+o, n+o\}}$ triconcepts in an $m \times n \times o$ tensor. Unless the tensor is tiny, it is impossible in practice to visualize them all at once in a diagram[26] generalizing the formal concept lattice. However, given a triconcept $X_1 \times X_2 \times X_3$, the entrywise conjunction of the matrices that are the "slices" of the Boolean tensor indexed by the elements of either X_1 or X_2 or X_3 may contain a reasonable number of formal concepts. Visualized in their lattice, each of them corresponds to one triconcept. It involves the elements of the formal concept plus a superset of the elements defining the conjunction. An analyst can go from $X_1 \times X_2 \times X_3$ to one of those triconcepts (T3) using the FCA Tools Bundle,[27] where FCA stands for Formal Concept Analysis. That repeated navigation only reaches a subset of the triconcepts.[28] Also, it does not apply to patterns generalizing the formal concepts to n-way *fuzzy* tensors,[29,30] even if $n = 3$.

The more 3-tuples in a triconcept, the greater its contribution to summarizing a Boolean 3-way tensor. The FCA Tools Bundle neither visualizes that piece of information (T1) nor allows to get a few patterns whose union includes many 3-tuples (T4). However, it proposes an interactive search of patterns in n-way tensors, whatever n.[31] With it, a constraint on the patterns is gradually turned stronger by adding, one by one, elements the patterns must involve and other elements they must not involve.[32] This article proposes a similar (but different and arguably more convenient) type of constraint to filter the patterns.

Relevant information derived from the disjunctive box cluster model

This section first presents the disjunctive box cluster model. It then emphasizes the information that, in this framework, is essential to properly interpret any pattern-based summary of a fuzzy tensor: the density of the tensor, those of the patterns, their areas, and their intersections. The section also proposes the computation of coarser summaries (by forward selection), the addition of user-defined constraints (that filter the patterns) and the gradual exploration of the summary by hierarchy and by similarity between patterns.

Disjunctive box cluster model

Given n *dimensions* (i.e., n finite sets) D_1, …, D_n, a *fuzzy tensor* $\mathbf{T}$ maps any n-tuple $t \in \prod_{i=1}^{n} D_i$ (where $\prod$ denotes the Cartesian product) to a value $\mathbf{T}_t \in [0, 1]$, called *membership degree* of t. Figure 2 shows (part of) a (larger) fuzzy tensor. Its $n = 3$ dimensions are $D_1 =$ {week 45, week 46}, $D_2 =$ {AbbadeMat, CarlosPort, ESPNagora Renatiruts, SempreCRF, TrechoFiel}, and $D_3 =$ {Corinthians, Flamengo, Fluminense, Grêmio, Santos, São Paulo}. In that tensor $\mathbf{T}$, $\mathbf{T}_{(\text{week 46, Renatiruts, Santos})} = 0.3$ is a membership degree.

A pattern in $\mathbf{T}$ is a set of n-tuples covered by a subtensor. More precisely, $X \subseteq \prod_{i=1}^{n} D_i$ is a *pattern* if and only if $\forall i \in \{1, \ldots, n\}$, $\exists X_i \subseteq D_i$ such that $X = \prod_{i=1}^{n} X_i$. The *area* of a pattern X is the number $|X|$ of n-tuples it includes. The arithmetic mean $\frac{\sum_{t \in X} \mathbf{T}_t}{|X|}$ of the membership degrees $\mathbf{T}$ associates with those n-tuples is the *density* of X. The areas of the three patterns listed in caption of Figure 2 are $2 \times 2 \times 2 = 8$, $2 \times 3 \times 4 = 24$, and $1 \times 3 \times 3 = 9$. Their respective densities are $\frac{8}{8} = 1$, $\frac{12}{24} = 0.5$, and $\frac{5.4}{9} = 0.6$.

The *disjunctive box cluster model*, as first defined by Mirkin and Kramarenko,[15] is a regression model. Given a set of patterns $\mathcal{X}$ (the explanatory variables) in a tensor $\mathbf{T}$, it models the membership degree of any n-tuple $t \in \prod_{i=1}^{n} D_i$:

$$\mathbf{T}_t = \lambda_0 + \max_{X \in \mathcal{X} \text{ s.t. } t \in X} \lambda_X + \epsilon \tag{1}$$

In that model, $\mathcal{X}$ is called the (pattern-based) *summary* of $\mathbf{T}$, λ_0 is an intercept, a parameter λ_X is associated with any pattern $X \in \mathcal{X}$, and ϵ is the error term. The usual convention $\max_{X \in \emptyset} \lambda_X = 0$ applies. As a consequence, $\lambda_0 + \epsilon$ models the membership degree associated with any n-tuple t out of $\cup_{X \in \mathcal{X}} X$, that is, in any cell no pattern of $\mathcal{X}$ covers. On the contrary, if the n-tuple t belongs to at least one pattern of $\mathcal{X}$, $\mathbf{T}_t = \lambda_0 + \lambda_X + \epsilon$, where λ_X is the greatest parameter among those associated with the pattern(s) of $\mathcal{X}$ that include(s) t.

Modeling $\mathbf{T}$ using no pattern ($\mathcal{X} = \emptyset$ in equation (1)), hence as a constant tensor $\hat{\mathbf{T}}$, the least-square estimator $\hat{\lambda}_0$ of λ_0 is $\frac{\sum_{t \in \prod_{i=1}^{n} D_i} \mathbf{T}_t}{\left|\prod_{i=1}^{n} D_i\right|}$. Then, minimizing the residual sum of squares $RSS_{\mathbf{T}}(\mathcal{X}) = \sum_{t \in \prod_{i=1}^{n} D_i} \left(\mathbf{T}_t - \hat{\mathbf{T}}_t\right)^2$ of the summary $\mathcal{X}$ with one single

arbitrary pattern X ($\mathcal{X} = \{X\}$ in equation (1)),[15] shows that $\hat{\lambda}_X = \frac{\sum_{t \in X}(\mathbf{T}_t - \hat{\lambda}_0)}{|X|}$ is optimal. Those estimators are easy to interpret: $\hat{\lambda}_0$ is the density of (the pattern $\prod_{i=1}^{n} D_i$ covering) the whole n-way fuzzy tensor and $\hat{\lambda}_0 + \hat{\lambda}_X = \frac{\sum_{t \in X} \mathbf{T}_t}{|X|}$ is the density of the pattern X. Figure 3 gives the tensor $\hat{\mathbf{T}}$ as estimated by the summary composed of the three patterns in Figure 2 and assuming the density of $\mathbf{T}$ is $\hat{\lambda}_0 = 0.1$ (because only part of it is shown). $\hat{\mathbf{T}}_t$ is $\hat{\lambda}_0$ if $t \notin \cup_{X \in \mathcal{X}} X$. Otherwise, it is $\hat{\lambda}_0 + \hat{\lambda}_X$, where X is the densest pattern such that $t \in X$.

Contribution of a single pattern to the summary

As the article introducing the disjunctive box cluster model[14] already notes, $RSS_{\mathbf{T}}(\emptyset) - RSS_{\mathbf{T}}(\{X\}) = |X|\hat{\lambda}_X^2$ measures the individual contribution of a pattern X to explaining the fuzzy tensor $\mathbf{T}$. Indeed, it is the difference between the residual sums of squares of the summary with no pattern (i.e., $\mathcal{X} = \emptyset$) and of the summary that only uses X as an explanatory variable (i.e., $\mathcal{X} = \{X\}$). In this way, the individual contribution of X to the summary grows with $|X|$, its area, and with the contrast between its density, $\hat{\lambda}_0 + \hat{\lambda}_X$, and that of the whole tensor, $\hat{\lambda}_0$.

Coarser summaries

To accurately summarize the fuzzy tensor $\mathbf{T}$, the set $\mathcal{X}$ in equation (1) may contain hundreds or more patterns. The analyst may not have enough time to interpret them all. In that situation, it is valuable to have the patterns ranked in such a way that only interpreting the top-k patterns provides a coarser summary of $\mathbf{T}$. If the analyst chooses $k = 1$, she asks for the *single* best pattern $X \in \mathcal{X}$ to summarize $\mathbf{T}$. As the previous subsection justified, X must maximize $|X|\hat{\lambda}_X^2$. In a ranking, that pattern X should be first.

If $k = 2$, a second pattern Y can complement the partial explanation of the fuzzy tensor that X gives. The pattern with the second highest *individual* contribution to the summary is not necessarily the best choice for Y. Indeed, if X and Y overlap, they carry redundant information: from the use of X as the sole explanatory variable to that of both X and Y, the accuracy of the estimated membership degrees at the intersection $X \cap Y$ cannot improve as much as they would from the empty summary (i.e., $\mathcal{X} = \emptyset$) to the one using only Y (i.e., $\mathcal{X} = \{Y\}$), that is, not as much as the difference $RSS_{\mathbf{T}}(\emptyset) - RSS_{\mathbf{T}}(\{Y\}) = |Y|\hat{\lambda}_Y^2$ suggests.

Following that rationale, the best pattern after X is the pattern Y that maximizes $RSS_{\mathbf{T}}(\{X\}) - RSS_{\mathbf{T}}(\{X, Y\})$, that is, that minimizes $RSS_{\mathbf{T}}(\{X, Y\})$. More generally, given a subset of patterns $\mathcal{X}' \subset \mathcal{X}$, the pattern $Y \in \mathcal{X} \setminus \mathcal{X}'$ that minimizes $RSS_{\mathbf{T}}(\mathcal{X}' \cup \{Y\})$ best complements $\mathcal{X}'$. Known in the literature on regression under the name *forward selection*, that greedy selection repeated $|\mathcal{X}|$ times sorts $\mathcal{X}$.

Whatever the number $k \in \{1, \ldots, |\mathcal{X}|\}$ the analyst chooses, she is shown the first k patterns in the sorted set. A non-greedy selection with a higher computational cost could find k patterns of $\mathcal{X}$ that better fit the fuzzy tensor, that is, that provide a smaller evaluation of $RSS_{\mathbf{T}}$. Nevertheless, it would be confusing in the context of an interactive tool: increasing the number of visualized patterns would discard patterns that were previously shown.

For an accurate summary of the whole tensor $\mathbf{T}$, k should not be chosen too small. Visualizing the function mapping k to $RSS_{\mathbf{T}}(\text{top-}k \text{ patterns of } \mathcal{X})$ helps making an informed decision. Its curve may even reveal an elbow: every pattern ranked after it improves the fit much less than every pattern ranked before. In this case, it makes sense to stop interpreting the ordered summary at the elbow. That heuristic is known in the literature as the *elbow method*.

Inclusions between patterns

A summary, $\mathcal{X}$ in equation (1), may contain *nested* patterns, that is, two patterns X and Y such that $X \subset Y$. For instance, in Figure 2, the second pattern includes the first one, which refines the analysis: the second pattern is not homogeneous; it includes a denser pattern, the first pattern. Only displaying the patterns of $\mathcal{X}$ that are not included in any other pattern of $\mathcal{X}$ provides a high-level summary of the fuzzy tensor. Nevertheless, the existence of patterns of $\mathcal{X}$ that are included in a visualized pattern Y should be brought to the attention of the analyst. Indeed, as exemplified above, they support a finer analysis of the n-tuples in Y, whose membership degrees are not uniformly distributed. The subset of patterns $\{X \in \mathcal{X} \mid X \subset Y\}$ is fundamentally a summary of the sub-tensor that Y defines, hence a nested summary. It may itself contain nested patterns. The interactive visualization should therefore allow the navigation in a hierarchy of summaries.

Considering the patterns of $\mathcal{X}$ as vertices and, for all $(X, Y) \in \mathcal{X}^2$, every inclusion $X \subset Y$ as the edge $Y \to X$, a directed acyclic graph (DAG) is obtained. Its sources are the patterns in the highest-level summary of the fuzzy tensor. A transitive reduction removes every edge $Y \to X$ if there exists $Z \in \mathcal{X}$ such that $X \subset Z \subset Y$. After that reduction, the successors of any vertex Y are all the patterns to visualize to start the interpretation of

Algorithm 1: filter

Input: set of patterns $\mathcal{V}$, constraint c, function *out* returning the successors of the input vertex
$\mathcal{S} \leftarrow \emptyset$
for $v \in \mathcal{V}$ **do**
  if $c(v)$ **then**
    $\mathcal{S} \leftarrow \mathcal{S} \cup \{v\}$
  else
    $\mathcal{S} \leftarrow \mathcal{S} \cup$ filter$(out(v), c, out)$
  end
end
return $\mathcal{S}$

the nested summary of the sub-tensor that Y defines. More precisely, they are "all the patterns to visualize" but some may be discarded by a constraint the analyst chooses to enforce.

Constraints on the patterns

A *constraint* is a function that takes as input a pattern and returns either "true", which means "valid", or "false", which means "invalid". An example of a constraint has already been given: "being among the top-k patterns of the summary". It can be formalized as $c(X) \equiv \text{pos}(X) \leqslant k$, where $\text{pos}(X)$ is the position of X in the sorted summary. Another example is $c(X) \equiv \text{pos}(X) \leqslant k \wedge \text{Flamengo} \in X_3$. It complements the previous constraint by additionally forcing the pattern to involve Flamengo, an element of the third dimension of the fuzzy tensor that may particularly interest the analyst.

Some sources of the DAG (for the highest-level summary of the fuzzy tensor) or some successors of a vertex (for the related nested summary) may be invalid. If so, they must not be shown. They must be substituted by their successors, or, for those that are invalid as well, their successors, etc. Given the set of patterns $\mathcal{V}$ that would be visualized in absence of constraint (either the sources of the DAG or the successors of a vertex), Algorithm 1 returns the patterns to actually visualize if the constraint c is enforced.

Partial intersections between patterns

Patterns can *partially* overlap. In caption of Figure 2, the second and third patterns share two 3-tuples: $\{\text{week 45}\} \times \{\text{CarlosPort}\} \times \{\text{Corinthians,Santos}\}$. At the intersection, the densest pattern prevails, in equation (1). Interpreting a pattern X of a summary $\mathcal{X}$, the analyst should therefore be aware of every overlapping pattern of $\mathcal{X}$, especially if the intersection between the two patterns is large: patterns that significantly overlap with X significantly alter the understanding that would be drawn from X alone. Visualizing the sizes of the intersections and describing the actual intersections reduce the risk of misinterpretation.

Similarities between patterns

The size of the intersection between two patterns is a possible measure of their similarity. Nevertheless, a finer measure can take into account that the patterns are explanatory variables of the disjunctive box cluster model. In particular, the measure proposed below naturally comes up and can grade the similarity between non-overlapping patterns.

Any two patterns X and Y of a summary of the fuzzy tensor $\mathbf{T}$ only influence the way equation (1) models a sub-tensor $\mathbf{T}'$ of $\mathbf{T}$: the smallest sub-tensor where both patterns are defined. It therefore makes sense to have the similarity between X and Y only depend on $\mathbf{T}'$. Intuitively, if its membership degrees are almost uniformly distributed, X and Y are similar and summarizing $\mathbf{T}'$ with $\{X, Y\}$ through equation (1) is not much better than using the density of $\mathbf{T}'$ everywhere, that is, not much better than its empty summary, $\emptyset$. On the contrary, $\{X, Y\}$ much better summarizes $\mathbf{T}'$ than $\emptyset$ if X and Y are dissimilar, that is, if they do not intersect (much) and if, inside and outside the two patterns, the membership degrees of $\mathbf{T}'$ sharply contrast. In this way, the difference $RSS_{\mathbf{T}'}(\emptyset) - RSS_{\mathbf{T}'}(\{X, Y\})$ comes up as a natural measure of the dissimilarity between X and Y.

If $X \sqcup Y = \prod_{i=1}^{n} X_i \cup Y_i$ denotes the smallest pattern including $X = \prod_{i=1}^{n} X_i$ and $Y = \prod_{i=1}^{n} Y_i$, the dissimilarity above is $RSS_{\mathbf{T}}(\{X \sqcup Y\}) - RSS_{\mathbf{T}}(\{X \sqcup Y, X, Y\})$ too. Indeed, the summaries of $\mathbf{T}$ and $\mathbf{T}'$ on a same side of the difference ($\{X \sqcup Y\}$ and $\emptyset$ on the left; $\{X \sqcup Y, X, Y\}$ and $\{X, Y\}$ on the right) identically model the membership degree of any n-tuple $t \in X \sqcup Y$ and, for $t \notin X \sqcup Y$, so do the summaries of $\mathbf{T}$ ($\{X \sqcup Y\}$ or $\{X \sqcup Y, X, Y\}$) on both sides of the difference, hence a cancelation.

So far, the dissimilarity quantifies in an *absolute* way how much the two patterns better summarize the sub-tensor than its empty summary. Since there are more membership degrees in larger sub-tensors, larger patterns tend to be more dissimilar. To correct that bias, the absolute dissimilarity between $X = \prod_{i=1}^{n} X_i$ and $Y = \prod_{i=1}^{n} Y_i$ is divided by a dissimilarity of reference that depends on $|X_1|$, …, $|X_n|$ and on $|Y_1|$, …, $|Y_n|$. A natural denominator is the dissimilarity that would be obtained if X and Y would still individually cover the same n-tuples with the same membership degrees, if

$X_i \cap Y_i = \emptyset$ for all $i \in \{1, \ldots, n\}$, and if $\mathbf{T}_t = \hat{\lambda}_0$ for all $t \notin X \cup Y$. In that scenario, the estimate for $\lambda_{X \sqcup Y}$ would be $\tilde{\lambda}_{X \sqcup Y} = \frac{|X|\hat{\lambda}_X + |Y|\hat{\lambda}_Y}{\prod_{i=1}^{n} |X_i| + |Y_i|}$ and the dissimilarity of reference is $|X|\left(\hat{\lambda}_X - \tilde{\lambda}_{X \sqcup Y}\right)^2 + |Y|\left(\hat{\lambda}_Y - \tilde{\lambda}_{X \sqcup Y}\right)^2$.

Visual codification and interaction

The previous section studied the fundamental pieces of information that the disjunctive box cluster model carries, as well as opportunities to progressively interpret, in that framework, pattern-based summaries of fuzzy tensors. Building upon that study, this section gives appropriate visual objects, attributes, and interaction techniques that ease the interpretation of such summaries. In particular, the tasks T1–T4 listed in the introduction are well supported. A tool written in Rust (for the backend), Angular (for the frontend) and D3.js (for the visualization) implements the proposed interactive visualization. It is published under the terms of the GNU GPLv3+ on https://homepages.dcc.ufmg.br/~lcerf/en/prototypes.html#boxcluster-vis.

Initialization and filters

The analysis starts with picking two files specifying the n-way fuzzy tensor $\mathbf{T}$ and a summary $\mathcal{X}$ of it. The visualization tool then computes (once and for all) the density $\hat{\lambda}_0$ of $\mathbf{T}$, the density $\hat{\lambda}_X$ of every pattern $X \in \mathcal{X}$, the areas of those patterns, their pairwise inclusions, the transitive reduction of those inclusions, and it sorts $\mathcal{X}$ by forward selection.

The highest-level summary of $\mathbf{T}$ is shown, that is, every pattern of $\mathcal{X}$ that relates to a source of the transitively-reduced DAG. For fewer patterns, buttons in a side panel allow to enforce constraints (T4). One such button shows a Cartesian coordinate system with the curve of the function mapping $k \in \{1, \ldots, |\mathcal{X}|\}$ to $RSS_{\mathbf{T}}(\text{top-}k \text{ patterns of } \mathcal{X})$. Its x-axis is a slider whose indicator moves together with a vertical line on top of the curve. It makes it easy to define a number of patterns corresponding to an elbow of the curve. Whatever that number k, Algorithm 1 enforces the constraint $c(X) \equiv \text{pos}(X) \leqslant k$ and returns the patterns to visualize.

However, an additional constraint can be applied through the other button. It leads to a tool where, for each dimension $D_i \in \{D_1, \ldots, D_n\}$, a drop-down menu with a textual search field allows to select a subset of interest $S_i \subseteq D_i$. The related constraint on any pattern $X = \prod_{i=1}^{n} X_i$ is $c(X) \equiv \prod_{i=1}^{n} X_i \cap S_i \neq \emptyset$, that is, the valid patterns are those involving at least one element of each of the n subsets. Whatever $i \in \{1, \ldots, n\}$, the default subset S_i is D_i and no constraint deals with the i th dimension.

Pattern representation

At the top of the visualization tool, a legend maps shades of red on a linear gradient to densities between 0 (white) and 1 (red). The background color of the main panel corresponds to the density of the visualized (sub-)tensor: $\hat{\lambda}_0$ for the highest-level summary; $\hat{\lambda}_0 + \hat{\lambda}_Y$ when visualizing the nested summary of the sub-tensor that the pattern $Y \in \mathcal{X}$ defines.

Disks are drawn on top of that uniformly-colored background. Each disk stands for a valid pattern $X \in \mathcal{X}$. The color of the disk, interpreted on the same scale as the one used for the background, corresponds to $\hat{\lambda}_0 + \hat{\lambda}_X$. As a consequence, the contrast between the background color and the foreground color relates to the difference of density that grows with the individual contribution of X to summarizing the (sub-)tensor. In the previous section, that contribution was also shown to linearly depend on $|X|$. That is why the area of the disk representing X is proportional to $|X|$. A legend maps the smallest and the largest disks to the numbers of n-tuples their areas represent. The redder and the larger a disk, the more attention it catches, as desired (T1).

If the mouse cursor is over the disk standing for the pattern $X = \prod_{i=1}^{n} X_i$, the side panel textually describes X: its area $|X|$, its density $\hat{\lambda}_0 + \hat{\lambda}_X$, how many elements of each dimension it involves, $|X_1|$, …, $|X_n|$, and what those elements are, X_1, …, X_n. In some dimension, too many elements may be involved to write them all in the available space. If so, a partial list is displayed. However, clicking on it gives the whole list in a modal window with a scroll bar and a search field to check whether specific elements appear (T2). For any $i \in \{1, \ldots, n\}$, the elements of X_i in the side panel or in the modal window are written in ascending order of density of the related *slices* of the tensor. Formally, $f \in X_i$ is after $e \in X_i$ if $\prod_{j=1}^{i-1} D_j \times \{f\} \times \prod_{j=i+1}^{n} D_j$ is denser than $\prod_{j=1}^{i-1} D_j \times \{e\} \times \prod_{j=i+1}^{n} D_j$. In this way, the elements whose presences are more surprising are listed first.

Navigation between patterns

A click on the disk representing a pattern pins its textual description and draws segments connecting that disk to every other disk that stands for an overlapping pattern. Every segment therefore represents an intersection. It can be clicked for a textual description, similar to that of a pattern (if X and Y are patterns, $X \cap Y$

is actually a pattern). The color of the segment is that of the darker of the two disks at its ends, in agreement with equation (1). Moreover, every connected disk becomes a pie chart: a blue slice covers a proportion of the disk that is the proportion taken by the related intersection or, for the clicked disk, by the union of all these intersections.

If a disk stands for a pattern Y that includes other patterns of the summary $\mathcal{X}$, the number $|\{X \in \mathcal{X} \mid X \subset Y\}|$ is written on it—indicating how many patterns of $\mathcal{X}$ are in the sub-tensor Y defines. After clicking on the disk for Y, a button leads to visualizing the related nested summary. Given $out(Y)$, the successors of Y in the transitively-reduced DAG, and the constraints (unchanged), Algorithm 1 returns the patterns to display and a button allows to go up the hierarchy, that is, back to the visualization with a disk for Y.

Finally, the disks on screen are positioned so that their distances are approximately proportional to the normalized dissimilarities between the represented patterns. Given any two distinct patterns X and Y of a set returned by Algorithm 1 to be visualized, $\frac{RSS_{\mathbf{T}}(\{X \sqcup Y\}) - RSS_{\mathbf{T}}(\{X \sqcup Y, X, Y\})}{|X|\left(\hat{\lambda}_X - \tilde{\lambda}_{X \sqcup Y}\right)^2 + |Y|\left(\hat{\lambda}_Y - \tilde{\lambda}_{X \sqcup Y}\right)^2}$ (see the end of the previous section) is computed in $O(|X \sqcup Y|)$ time, unless it has already been computed and cached. Given all the pairwise normalized dissimilarities, multidimensional scaling with the SMACOF algorithm[32] provides 2D coordinates for the centers of the disks. They are scaled so that the disks are scattered as much a possible while staying on screen. Besides the navigation to overlapping or even nested patterns, the navigation to similar patterns (T3) becomes available, moving the mouse cursor to nearby disks.

Case study

From 13 January 2014 to 6 April 2014, Twitter (now called X) messages related to the Brazilian soccer were collected and tagged with the mentioned teams. During those 12 weeks, 170,670 users were retweeted (i.e., reblogged) at least once when they wrote about some of the 29 considered teams. The process in caption of Figure 1 turns the total number of retweets received over any week by any user mentioning any team into a membership degree. The resulting $12 \times 170\,670 \times 29$ fuzzy tensor with 452,409 nonzero membership degrees is summarized using a state-of-the-art algorithm.[11] It gives 55 patterns.

Figure 4 shows the screen that is displayed right after they are loaded together with the fuzzy tensor in the proposed visualization tool. The background of the main area, on the right side, is white. The legend in the top right-hand corner converts that color to 0.

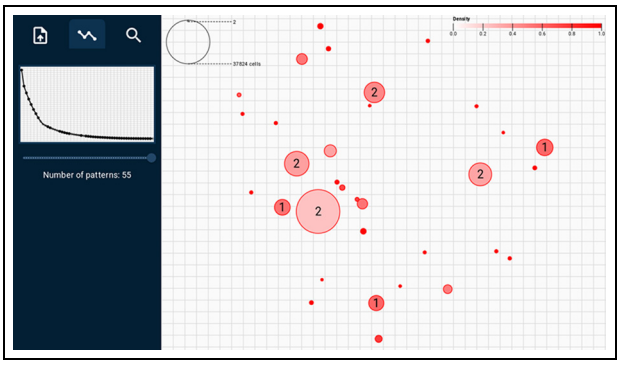

Figure 4. Highest-level summary of the 55 patterns.

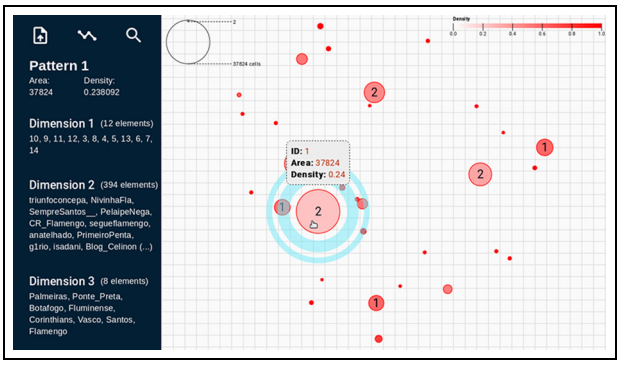

Figure 5. Pointing the disk that catches the most attention.

The actual density of the fuzzy tensor is $\hat{\lambda}_0 = 0.001$. Unsurprisingly, most of the 170,670 users have almost no influence. In fact, 82,573 of them received one single retweet over all 12 weeks.

The disks on the white background are filled with colors that contrast with it: they stand for sub-tensors that are significantly denser. The borders of the disks are pure red, corresponding to 1. Thus, the difference between the border color and the fill color represents how far the density of the related sub-tensor is from 1. The disks have varying areas. A legend proportionally maps them to pattern areas. Smaller disks are difficult to point without zooming, which is implemented and controlled by the mouse wheel. The color contrasts and the areas of the disks bring to the attention the patterns that individually contribute more to the summary.

In this case study, the largest pattern provides the greatest contribution and the corresponding disk in Figure 4 indeed catches the most attention. Hovering the mouse pointer over that disk gives the screen in Figure 5. The left panel textually describes the related pattern. Light blue circles around the pointed disk serve as a visual aid to link the current description to

Figure 6. Searching users whose names include "Palm".

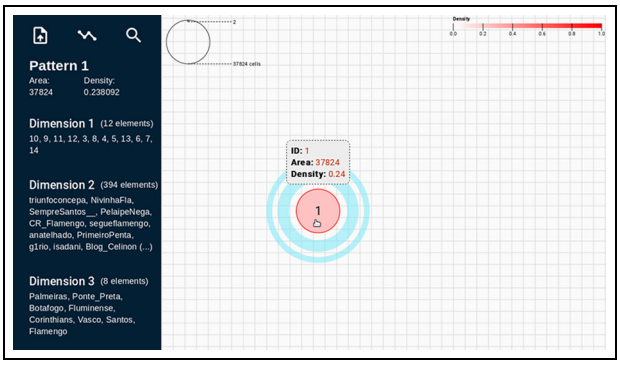

Figure 8. Visualizing the patterns involving Palmeiras.

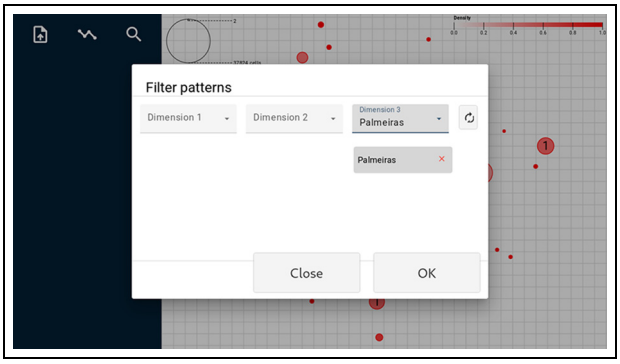

Figure 7. Forcing the involvement of the Palmeiras team.

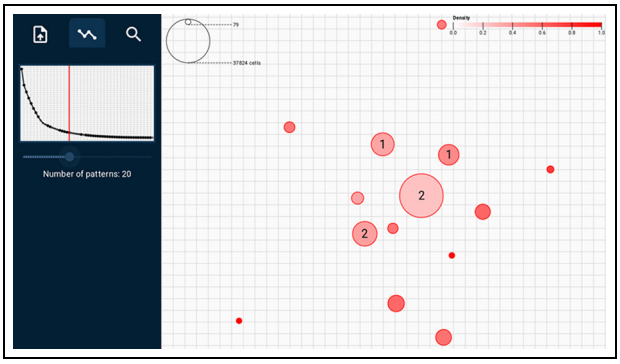

Figure 9. The top-20 patterns, a coarser summary.

the graphical representation of the corresponding pattern. It involves all 12 weeks (elements of the first dimension), 394 users (of the second dimension), and eight teams (of the third dimension): the four teams based in Rio de Janeiro and four major teams in São Paulo. Commentators of the Brazilian soccer in general tend to focus on those two states, with much rivalry between excellent teams.

There is not enough space in the panel to list all the 394 users who were rather influential (a 0.238 degree on average, the density) when they wrote about the eight teams over the entire data collection period. Only the users with the smallest overall numbers of normalized retweets appear. However, after clicking on the disk (to pin the description) and then on the partial list, all the users involved in the pattern can be browsed and searched. In Figure 6, while "Palm" is typed, the list of 394 users is reduced to the 15 users whose names include that prefix of "Palmeiras" (one of the teams).

If a particular interest in Palmeiras exists (for some reason), the pattern(s) involving that team can be filtered by clicking on the magnifying glass in the left panel, then on Palmeiras in the searchable drop-down menu relating to the third dimension and finally on the OK button. Figure 7 shows the window defining the generic constraint presented in the previous section. Here, the resulting visualization, in Figure 8, consists of a single disk. It stands for the already-discussed largest pattern of the summary. However, the number at the center of the disk has decreased from 2 to 1: two patterns of the summary are nested in its largest pattern, but only one of them involves Palmeiras.

The analyst, with maybe enough time to interpret 20 patterns, may decide to less drastically reduce the summary. The button after the drop-down menus in Figure 7 removes the whole constraint. Elements can also be deselected one by one. Back to the screen on Figure 4, a click on the second button in the left panel displays in function of k the curve of the residual sum of squares of the top-k patterns. In Figure 9, the slider on the x-axis sets k to 20. Thanks to the vertical line on top of the cursor, k could alternatively be easily defined as the abscissa where the curves bends, $k = 9$.

Focusing on the top-20 patterns, the visualization in Figure 9 is of course less cluttered than in Figure 4. The presence of nested patterns helps too: only 14 of the 20 patterns are in the highest-level summary. By clicking on a disk with a number at its center, a button appears on the right. It leads to the visualization of the

Figure 10. Patterns nested in the pattern pointed in Figure 5.

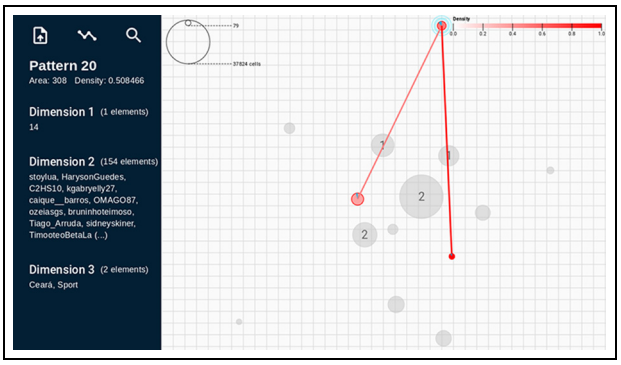

Figure 11. Visualizing overlapping patterns, among the top-20.

summary of the related sub-tensor, with as many patterns as the number informs.

Figure 10 shows the summary of the sub-tensor that the largest pattern defines. There are still two patterns in it, because both are among the top-20. The disks that represent them are graphically similar: each pattern covers around 3000 3-tuples and has an approximate density of 0.6. That density contrasts with that of the sub-tensor, 0.238, and, consequently, so does the color of a disk with the background. As noted earlier, Palmeiras appears in only one of the two patterns, alongside two other teams that are also based in the São Paulo state. The other pattern, described in Figure 10, involves all four teams from Rio de Janeiro. All 12 weeks appear in both cases.

It turns out that, setting aside the first pattern, each of the top-9 patterns, which matter the most for the accuracy of the summary, exclusively involves teams in a single different state and at least 11 of the 12 weeks. The upper button at the right of Figure 10 brings back the visualization in Figure 9. By moving the cursor over its largest nine disks, the analyst can sequentially read in the left panel the descriptions of the top-9 patterns. That movement naturally tends to go to nearby disks, representing similar patterns. One pattern involves the two major teams from the Minas Gerais state, those of Rio Grande do Sul appear in another pattern, etc. That makes sense: users who consistently post influential messages about their favorite team are often retweeted as well when they post about its main rival(s), in the same state. Every pattern involves sport journalists and supporters of the teams appearing in the pattern.

The residual sum of squares of the top-k patterns decreases much less after $k = 9$. It does decrease though: the patterns beyond the top-9 refine the summary. Some of them share 3-tuples. Clicking on a disk not only pins the description of the related pattern, but also connects the disk to those representing overlapping patterns and shows the proportion of the area every involved pattern shares.

In Figure 11, two patterns partially overlap with the pattern that the topmost disk represents. According to its description in the left panel, that pattern stands for 154 users who were influential when they mentioned two Northeastern region teams, Ceára and Sport, during the last week of data collection. It includes Wednesday 2 April 2014. That day, Ceára and Sport disputed the finals of the *Copa do Nordeste*, hence the rather high influence of many users writing about these teams, during that week: a 0.508 degree in average, the density of the pattern.

Approximately 13% of its area, 308, is shared with at least one of the two overlapping patterns, as the pie chart replacing the disk shows, in Figure 11. At the intersection that the redder segment represents, the corresponding density, close to 1, prevails according to equation (1). A click on the segment describes the intersection; a click on the disk on the other end, the overlapping pattern. It stands for seven users who were very influential when they mentioned the Ceára team not only during the last week of the data collection but over it all. Analogously, the other overlapping pattern, which is larger but less dense, deals with 67 users writing about the other team, Sport, over all 12 weeks.

Conclusion

This article has proposed the first tool to interactively visualize pattern-based summaries of fuzzy tensors. The disjunctive box cluster model applies in this generic context, which includes (but is not restricted to) Boolean tensors and fuzzy matrices. An in-depth study of that model has led to the identification of relevant information whose presentation and exploitation ease the interpretation of summaries.

For example, the article has proposed the difference between the residual sums of squares of two

summaries as a measure of dissimilarity between patterns, its normalization, and multidimensional scaling to obtain positions on the screen. This has enabled the similarity-based exploration of the summary. Nevertheless, empowered by other features of the tool, the analyst may instead decide to visualize the nested patterns (if any) of the currently-interpreted pattern, or an overlapping pattern with much intersection (visualized as well), or an unrelated large and dense pattern rightfully catching the attention for contributing a lot to the summary.

Special care has been taken to allow the interpretation of summaries with many patterns involving many elements. For instance, typing in search fields tests the presence of elements and filters can focus the visualization on the top-k patterns (where k can be defined by the elbow method, with the assistance of the tool) involving elements of interest. A case study using a real-world tensor has demonstrated those capacities. The tool has indeed supported an effective analysis of 55 patterns involving thousands of elements. Overall, this work has provided a valuable resource for practitioners and has contributed to bridging the gap between complex data models and human interpretability.

The considered case study has dealt with social media analysis. However, a variety of other domains can benefit from the developed visualization tool. For example, some transcriptomics technologies measure over time the expression of genes in different tissues. Membership degrees in a 3-way fuzzy tensor can quantify, at every time snapshot, to what extent the genes are under-expressed or over-expressed in the tissues. Visualizing patterns in that tensor may reveal dynamic and tissue-dependent co-expressions, maybe suggesting unknown co-regulations. For market basket analysis, a matrix classically indicates whether customers bought items. In the more general context of this work, the customers can be grouped by demographic segment, their numbers of purchases turned into interest degrees and several such matrices (maybe one per shop, in the case of a chain) analyzed altogether, as a 3-way fuzzy tensor. Visualizing a pattern-based summary of it may then support a division of the items into natural groups.

ORCID iD

Loïc Cerf https://orcid.org/0000-0002-3597-0264

Ethical considerations

Not applicable.

Consent to participate

Not applicable.

Author contributions

The authors together designed the interactive visualization. Victor Henrique Silva Ribeiro implemented it. He also wrote most of the Related Work section and of the case study. Loïc Cerf supervised Victor and wrote the rest of the manuscript. Both authors reviewed it all.

Funding

The authors disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: Partial financial support was received from the Brazilian National Council for Scientific and Technological Development (CNPq), the Coordination of Superior Level Staff Improvement (CAPES) the Minas Gerais State Research Support Foundation (FAPEMIG), and by the projects INC TCyber, CIIA-Saude, and PCA (MPMG). Victor's scientific initiation is funded by the FAPEMIG/PRPq/UFMG.

Declaration of conflicting interests

The authors declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Data availability statement

The interactive visualization tool, the data and the patterns used for the case study are published under the terms of the GNU GPLv3+ on https://homepages.dcc.ufmg.br/~lcerf/en/prototypes.html.

References

1. Leenen I, Van Mechelen I, De Boeck P, et al. INDCLAS: a three-way hierarchical classes model. *Psychometrika* 1999; 64(1): 9–24.
2. Miettinen P. Boolean tensor factorizations. In: *IEEE international conference on data mining*, Vancouver, BC, Canada, 11–14 December 2011, pp.447–456.
3. Belohlavek R, Glodeanu C and Vychodil V. Optimal factorization of three-way binary data using triadic concepts. *Order* 2013; 30(2): 437–454.
4. Erdős D and Miettinen P. Walk 'n' Merge: A scalable algorithm for Boolean tensor factorization. In: *IEEE international conference on data mining*, Dallas, TX, USA, 7–10 December 2013, pp.1037–1042.
5. Metzler S and Miettinen P. Clustering Boolean tensors. *Data Min Knowl Discov* 2015; 29(5): 1343–1373.

6. Rukat T, Holmes CC and Yau C. Probabilistic Boolean tensor decomposition. In: *international conference on machine learning*, Stockholm, Sweden, 10–15 July 2018, pp.4410–4419.
7. Park N, Oh S and Kang U. Fast and scalable method for distributed Boolean tensor factorization. *VLDB J* 2019; 28(4): 549–574.
8. Wan C, Chang W, Zhao T, et al. Geometric all-way Boolean tensor decomposition. In: *Annual conference on neural information processing systems, virtual*, 6–12 December 2020, pp.2848–2857.
9. Maciel L, Alves J, dos Santos VF, et al. Climbing the hill with ILP to grow patterns in fuzzy tensors. *Int J Comput Intell Syst* 2020; 13(1): 1036–1047.
10. Ribeiro VHS and Cerf L. Summarizing fuzzy tensors with sub-tensors. In: *ACM/SIGAPP symposium on applied computing*, Tallinn, Estonia, 27–31 March 2023, pp.362–364.
11. Ribeiro VHS and Cerf L. Summarizing Boolean and fuzzy tensors with sub-tensors. *Information Sciences* 2025; 719: 122489.
12. Belohlavek R and Vychodil V. Factor analysis of incidence data via novel decomposition of matrices. In: *International conference on formal concept analysis*, Darmstadt, Germany, 21–24 May 2009, pp. 83–97.
13. Karaev S and Miettinen P. Algorithms for approximate subtropical matrix factorization. *Data Min Knowl Discov* 2019; 33(2): 526–576.
14. Belohlavek R and Trneckova M. Factorization of matrices with grades with overcovering. *IEEE Trans Fuzzy Syst* 2024; 32(4): 1641–1652.
15. Mirkin BG and Kramarenko AV. Approximate bicluster and tricluster boxes in the analysis of binary data. In: *International conference on rough sets, fuzzy sets, data mining and granular computing*, Moscow, Russia, 25–27 June 2011, pp.248–256.
16. Grothaus GA, Mufti A and Murali TM. Automatic layout and visualization of biclusters. *Algorithms Mol Biol* 2006; 1: 15.
17. Jin R, Xiang Y, Fuhry D, et al. Overlapping matrix pattern visualization: a hypergraph approach. In: *IEEE international conference on data mining*, Pisa, Italy, 15–19 December 2008, pp.313–322.
18. Colantonio A, Di Pietro R, Ocello A, et al. Visual role mining: a picture is worth a thousand roles. *IEEE Trans Knowl Data Eng* 2012; 24(6): 1120–1133.
19. Marette T, Miettinen P and Neumann S. Visualizing overlapping biclusterings and Boolean matrix factorizations. In: *European conference on machine learning and knowledge discovery in databases*, Turin, Italy, 18–22 September 2023, pp.743–758.
20. Tatti N and Miettinen P. Boolean matrix factorization meets consecutive ones property. In: *SIAM international conference on data mining*, Calgary, AB, Canada, 2–4 May 2019, pp.729–737.
21. Sun M, Zhao J, Wu H, et al. The effect of edge bundling and seriation on sensemaking of biclusters in bipartite graphs. *IEEE Trans Vis Comput Graph* 2019; 25(10): 2983–2998.
22. Wu H, Vreeken J, Tatti N, et al. Uncovering the plot: detecting surprising coalitions of entities in multi-relational schemas. *Data Min Knowl Discov* 2014; 28(5–6): 1398–1428.
23. Zhao J, Sun M, Chen F, et al. BiDots: visual exploration of weighted biclusters. *IEEE Trans Vis Comput Graph* 2018; 24(1): 195–204.
24. Sun M, Shaikh AR, Alhoori H, et al. SightBi: exploring cross-view data relationships with biclusters. *IEEE Trans Vis Comput Graph* 2022; 28(1): 54–64.
25. Streit M, Gratzl S, Gillhofer M, et al. Furby: fuzzy force-directed bicluster visualization. *BMC Bioinformatics* 2014; 15(S6): S4.
26. Lehmann F and Wille R. A triadic approach to formal concept analysis. In: *International conference on conceptual structures*, Santa Cruz, CA, USA, 14–18 August 1995, pp.32–43.
27. Kis LL, Săcărea C and Troancă D. FCA tools bundle - a tool that enables dyadic and triadic conceptual navigation. In: *International workshop "what can FCA do for artificial intelligence"?*, The Hague, The Netherlands, 30 August 2016, pp.43–50.
28. Rudolph S, Săcărea C and Troancă D. Towards a navigation paradigm for triadic concepts. In: *International conference on formal concept analysis*, Nerja, Spain, 23–26 June 2015, pp.252–267.
29. Georgii E, Tsuda K and Schölkopf B. Multi-way set enumeration in weight tensors. *Mach Learn* 2011; 82(2): 123–155.
30. Cerf L and Meira W Jr. Complete discovery of high-quality patterns in large numerical tensors. In: *IEEE international conference on data engineering*, Chicago, IL, USA, 31 March–4 April, 2014, pp.448–459.
31. Kis LL, Săcărea C and Şotropa DF. Visualizing conceptual structures using FCA tools bundle. In: *International conference on conceptual structures*, Edinburgh, UK, 20–22 June 2018, pp.193–196.
32. Rudolph S, Săcărea C and Troancă D. Membership constraints in formal concept analysis. In: *International joint conference on artificial intelligence*, Buenos Aires, Argentina, 25–31 July 2015, pp.3186–3192.
33. Kruskal JB. Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis. *Psychometrika* 1964; 29(1): 1–27.