Inf Retrieval J (2018) 21:24–55
https://doi.org/10.1007/s10791-017-9314-x

# Machine learning techniques for XML (co-)clustering by structure-constrained phrases

**Gianni Costa**[1] · **Riccardo Ortale**[1]

Received: 1 September 2016 / Accepted: 1 August 2017 / Published online: 4 August 2017


**Abstract** A new method is proposed for clustering XML documents by structure-constrained phrases. It is implemented by three machine-learning approaches previously unexplored in the XML domain, namely non-negative matrix (tri-)factorization, co-clustering and automatic transactional clustering. A novel class of XML features approximately captures structure-constrained phrases as n-grams contextualized by root-to-leaf paths. Experiments over real-world benchmark XML corpora show that the effectiveness of the three approaches improves with contextualized n-grams of suitable length. This confirms the validity of the devised method from multiple clustering perspectives. Two approaches overcome in effectiveness several state-of-the-art competitors. The scalability of the three approaches is investigated, too.

**Keywords** XML · Semi-structured data analysis · XML (co-)clustering by structure and nested text · Structure-constrained phrases · Contextualized n-grams

## 1 Introduction

The eXtensible Markup Language (XML) (W3C 2008) is a simple and flexible model for representing, storing and exchanging data under the form of XML documents (Abiteboul 1997; Abiteboul et al. 2000; Wilde and Glushko 2008). These are machine-readable and human-intelligible text files, wherein markup is used to annotate textual data. This is accomplished by embedding the latter in the context of elements, i.e., pairs of matching tags, that can also include further hierarchically nested elements. The flexibility with which

✉ Riccardo Ortale
ortale@icar.cnr.it

Gianni Costa
costa@icar.cnr.it

[1] ICAR-CNR, Via P. Bucci 41c, Rende, CS, Italy

tags can be chosen and nested makes the resulting logical structure of XML documents self-descriptive and explicative of the textual data, which enables better information retrieval (Aggarwal et al. 2007), filtering and, more generally, management.

These appealing features are at the basis of the widespread adoption of XML in many heterogeneous domains including (but not limited to) online documentations, electronic commerce, medicine (Piernik et al. 2015), health-care systems, legacy as well as scientific data repositories, financial services, chemistry (Piernik et al. 2015), mathematics (Piernik et al. 2015), bio-informatics, digital libraries, defense, aviation, data exchange and information systems on the Web. Besides, XML also allows for the addition of logical structure to the existing collections of unstructured data (Bratko and Filipic 2006), thus enabling focused access to the required information. Therein, it is worth noticing that many of the electronic files produced by modern text processors are essentially XML files.

Interestingly, although the possibility that XML documents conform to some fixed schema is admitted, a very large fraction of XML data exhibits the inherent properties of semistructured data, i.e., potentially irregular or incomplete data whose structure may change rapidly or unpredictably (Connolly and Begg 2002).

The analysis of XML data is the process of pattern discovery in (generally very large) volumes of XML documents. In particular, clustering is a fundamental task for the unsupervised analysis of XML data aimed, in its most general form, to divide a collection of XML documents into cohesive clusters, i.e., disjoint subsets of XML documents with a relevant extent of content and structural homogeneity. This is useful in several applicative settings including query processing, data integration and indexing, information retrieval and extraction, document filtering, browsing and summarization (Algergawy et al. 2011), in addition to Web mining, bioinformatics and spatial data management (Piernik et al. 2015). Moreover, XML security has recently emerged as a primary threat to modern information systems. In such a domain, XML clustering may be a helpful tool for the discovery and understanding of the vulnerabilities of information systems to patterns of anomalous XML encoding, which inform the separation of their input XML documents (or messages) into various types of harmless interactions as well as harmful cyber-attacks (Menahem et al. 2016).

XML clustering poses several challenges. Foremost, the explicit manipulation of XML documents to catch content and structural resemblance embraces several research issues, namely the alignment of their (sub)structures, the identification of similarities between such (sub)structures and between the textual data nested therein, along with the discovery of possible mutual semantic relationships among textual data and (sub)structure labels. Moreover, resemblance between the structures and textual contents of XML documents should be caught at a semantic (i.e., topical) level. Additionally, the discovery of clusters of XML documents should take advantage of the simultaneous discovery of clusters of related features, so that the XML documents would be grouped based on feature commonality, while features would be grouped based on the XML documents which they occur in. The interplay between both types of clusters is in principle beneficial, since specific groups of XML documents tend to be co-related only under certain subsets of features and would otherwise be hardly recognized (especially in a high-dimensional setting) as actually similar because of (possibly several) differences in the other features. Yet, devising representative features, that are informative of both the content and structure of the XML documents for effective, efficient and scalable clustering requires non trivial efforts, that often imply focusing on suitable fragments of the XML documents at hand. Therein, research efforts have hitherto focused on adopting a tree-like model of the logical structure of XML documents along with a *bag-of-words* representation for the contextualized textual

data and, accordingly, a great variety of clustering features have been proposed having tags, twigs, paths or subtrees as logical structures and word character n-grams or whole words as content. Although representative of both the structural and content aspects of XML documents, such features do not allow the clustering process to reliably unveil topical relatedness among XML documents, since the *bag-of-words* representation does not catch the actual meaning of the textual data. It is thus possible that the occurrence of a number of same words in the context of similar substructures from multiple XML documents can be erroneously considered as an evidence of structural and topical relatedness, even though the meaning of the particular combination (i.e., order of occurrence) of those words in the context of the individual substructures differs.

In this manuscript, we explore several new research directions to deal with the foresaid challenging issues. In particular, we propose a novel method, i.e., clustering XML documents by structure-constrained (approximated) phrases (or, equivalently, n-grams), that is conceived to preserve (as much as possible) the meaning of the structure-constrained text data for improved partitioning effectiveness. We draw inspiration from text clustering by phrases, which has been traditionally studied in various fields such as text mining, information retrieval and natural language processing as an effective approach to capture the meaning of word sequences in purely textual documents. Motivated by these findings, the intuition behind the devised XML clustering method is to exploit a *bag-of-phrases* representation for the text items of the XML documents together with word n-grams. These are used to deal with the uncertainty in the meaning of the individual text items, by taking advantage of the inherent disambiguation provided by contextualized phrases (i.e., sequences of adjacent text items), wherein the grouping of multiple text items along with their explicit ordering better capture and retain the meaning of nested text. In turn, this is also beneficial to deal with possible ambiguities arising from the contextualizing logical structures, whose meaning is thus better caught on a semantic level. In order to gain a deeper understanding of the devised method and comparatively investigate its effectiveness, we present and investigate three different instantiations of clustering XML documents by structure-constrained n-grams into as many approaches. XC-NMF (*XML Clustering based on Non-negative Matrix Factorization*) (Costa and Ortale 2013b) partitions a corpus of XML documents by structure-constrained n-grams into topically homogeneous groups through non-negative matrix factorization. The latter is performed through an alternating least squares method, which incorporates expedients to attenuate the burden of large-scale factorizations. This is especially relevant when massive text-centric XML corpora are processed. XCo-Clust (*XML Co-Clustering*) (Costa and Ortale 2014) allows for simultaneously clustering a collection of XML documents and their respective XML features by taking advantage their mutual interactions. Co-clustering is operated via a non-negative matrix tri-factorization technique, that efficiently processes large-scale input data, which is useful with large corpora of text-centric XML documents. XPart (*XML Partitioning*) (Costa and Ortale 2015a, 2017) is a fully-automatic technique to partition a body of XML documents via a transactional clustering scheme, where no user intervention is required to specify the number of clusters to find.

XC-NMF, XCo-Clust and XPart adopt flattened representations of the XML documents over a set of discriminative XML features, i.e., n-grams in the context of root-to-leaf paths, which are truly representative of content nesting into structure. The structural and content homogeneity of the resulting clusters of XML documents is obtained from these XML features, rather than by explicitly computing the pair-wise similarity of their tree-like representations.

We emphasize that XC-NMF, XCo-Clust and XPart involve state-of-the-art machine-learning techniques, that in this manuscript are exploited in a novel manner, i.e., to operate on the foresaid XML features. Although the design of new techniques is certainly interesting and worthy of further efforts, we believe that most-effective conventional machine-learning techniques are tools of choice for scientific investigation in cutting edge research for three major reasons. Firstly, various machine learning techniques can be employed to work with suitably flattened representations of the XML documents. This enables the investigation of the validity of focusing on n-grams within root-to-leaf paths as XML clustering features from multiple perspectives. Secondly, it is possible to systematically study and assess the potential of traditional machine-learning techniques in the context of the newly-identified research directions, according to spirit of the XML Document Track held at the INEX competition (Denoyer and Gallinari 2007, 2008). Thirdly, the identification of the most effective machine-learning techniques for the task of XML clustering by structure-constrained phrases informs subsequent research on advanced ad-hoc processing approaches.

To the best of our knowledge, this is the first research effort aimed to investigate the benefits of addressing structure-constrained n-grams in the XML clustering process from different perspectives, i.e., non-negative matrix (tri-)factorization, co-clustering and automatic transactional clustering.

The devised approaches are bundled with a suitable technique for XML feature selection, with which to face the *curse-of-dimensionality* phenomenon arising from the possible combinations in the above flattened representations of the logical substructures of the XML documents with their word n-grams, that becomes particularly problematic in very large corpora of text-centric XML documents (i.e., XML documents with large amounts of textual data). In particular, XCo-Clust and XPart operate a preliminary selection of the XML features by their relevance to the XML clustering process. Two new schemes are presented for quantifying feature selection.

An intensive experimentation of XC-NMF, XCo-Clust and XPart over real-world benchmark XML corpora is conducted to evaluate the validity of XML clustering by structure-constrained phrases and compare its effectiveness against the performance of several state-of-the-art competitors for XML clustering. The relative scalability of XC-NMF, XCo-Clust and XPart is investigated as well.

In this manuscript, the previous approaches in Costa and Ortale (2013b, 2014, 2015) are drawn together into a single cohesive study, which reviews, extends and advances a whole line of research in the XML domain by several original contributions. These substantially innovate the two earlier approaches in Costa and Ortale (2013b, 2014) in the spirit of the preliminary research efforts later presented in Costa and Ortale (2015a, 2017). Overall, the ideas in Costa and Ortale (2015a) are consolidated, deepened, experimentally compared and expanded into an original research vein, through an innovative and supplementary understanding of Costa and Ortale (2013b, 2014) from a significantly new perspective. A detailed preview of the novel contributions of this manuscript is reported below.

- Several unexplored research directions for XML clustering are proposed at two distinct hierarchical levels. At the upper level, a new general method is introduced for XML clustering by structure-constrained phrases. At the lower level, three different approaches (i.e., XC-NMF, XCo-Clust and XPart) implement the devised method in terms of as many conventional machine-learning techniques (non-negative matrix (tri-)factorization, co-clustering and automatic transactional clustering).

- The general method, i.e., XML clustering by structure-constrained phrase, and one machine-learning instantiation, i.e., XPart, were introduced in Costa and Ortale (2015a). In this manuscript, XC-NMF, XCo-Clust are generalized, reviewed and tested as further instantiations of XML clustering by structure-constrained n-grams, in order to gain a deeper comprehension of the potential of the devised method. Remarkably, in their original references (Costa and Ortale 2013b, 2014), XC-NMF, XCo-Clust were originally intended as simpler and unrelated approaches, focused on simpler XML features (i.e., individual words in the context of root-to-leaf paths), that do not properly catch phrase meaning.
- The new class of XML features originally designed for XPart is also used in XC-NMF and XCo-Clust, for the purpose of absorbing non-negative matrix (tri-)factorization under XML clustering by structure-constrained phrase. This grounds XC-NMF, XCo-Clust and XPart in the exploitation of a common representation for XML documents, that is truly representative of text nesting into logical structure and, additionally, allow for a controlled extent of phrase approximation.
- A unified notation is adopted to cover XC-NMF, XCo-Clust and XPart.
- Feature selection without any tunable threshold is designed to choose a subset of the XML features on the basis of their clustering relevance. The latter is quantified through two new definitions of XML feature relevance.
- An intensive empirical evaluation is conducted to compare XC-NMF, XCo-Clust and XPart against several state-of-the art competitors. Experiments new to the XML domain are carried out. All tests go far beyond the ones in the original references (Costa and Ortale 2013b, 2014, 2015a). More precisely, in Costa and Ortale (2013b, 2014), XC-NMF and XCo-Clust are proposed and evaluated as techniques for clustering XML documents only by contextualized unigrams (i.e., n-grams of length 1 in the context of root-to-leaf paths). Instead, in this manuscript, XC-NMF and XCo-Clust are generalized to operate with contextualized n-grams. Accordingly, a new and broader empirical assessment is designed and carried out, in order to investigate the previously unexplored effectiveness of XC-NMF and XCo-Clust against XPart as well as several other state-of-the-art competitors, when contextualized n-grams of varying length are targeted.
- The validity of XML clustering by structure-constrained phrases on the chosen XML corpora is confirmed by the experimental results.
- The exploitation of contextualized n-grams into XCo-Clust and XPart is shown to result into a superior effectiveness with respect to several state-of-the-art competitors for XML clustering. Interestingly, this also reflects the suitability of certain conventional machine-learning techniques to the structured XML domain, provided that flatten representations of the XML documents over suitable XML features are identified.
- The relative scalability of XC-NMF, XCo-Clust and XPart is studied.

The rest of this manuscript proceeds as follows. Section 2 introduces notation and preliminaries. Section 3 covers the XC-NMF approach. Section 4 discusses the XCo-Clust approach. Section 5 treats the XPart approach. Section 6 presents a comparative experimentation of XC-NMF, XCo-Clust, XPart on real-world benchmark XML corpora. Section 7 reviews a selection of related works. Lastly, Sect. 8 concludes and highlights future research.

## 2 Preliminaries

In this section we introduce the notation adopted throughout the manuscript along with some basic concepts.

### 2.1 Tree-based XML document representation

A suitable XML tree representation is adopted to model XML documents with no references (Abiteboul et al. 2000). Essentially, such a representation refines the conventional *rooted labeled tree* to account for the contextualization of content into structure.

*An* **XML tree** *is a rooted, labeled, tree* $\mathbf{t} = (\mathbf{V_t}, r_\mathbf{t}, \mathbf{E_t}, \lambda_\mathbf{t})$, *where*:

- $\mathbf{V_t} \subseteq \mathbb{N}$ is a set of nodes, with $r_\mathbf{t} \in \mathbf{V_t}$ being the root of $\mathbf{t}$;
- $\mathbf{E_t} \subseteq \mathbf{V_t} \times \mathbf{V_t}$ is a set of edges, catching the parent–child relationships between nodes of $\mathbf{t}$;
- $\lambda_t : \mathbf{V_t} \mapsto \Sigma$ is a labeling function, with $\Sigma$ being the domain of node tags (or, equivalently, labels).

Let $\mathbf{t}$ be a generic XML tree. $\mathbf{V_t}$ can be partitioned into the set $\mathbf{L_t}$ of *leaves* and the set $\mathbf{V_t} - \mathbf{L_t}$ of *inner nodes*. Leaves are nodes with no children, which can only contextualize textual data. Instead, inner nodes have at least one child.

A root-to-leaf path $p_l^{r_\mathbf{t}}$ in $\mathbf{t}$ is a sequence of nodes encountered along the path from the root $r_\mathbf{t}$ to a leaf node $l$ in $\mathbf{L_t}$, i.e., $p_l^{r_\mathbf{t}} = <r_\mathbf{t}, \ldots, l>$. Notation $\lambda_\mathbf{t}(p_l^{r_\mathbf{t}})$ represents the sequence of labels, which are associated in the XML tree $\mathbf{t}$ with the nodes of path $p_l^{r_\mathbf{t}}$, i.e., $\lambda_\mathbf{t}(p_l^{r_\mathbf{t}}) = <\lambda_\mathbf{t}(r_\mathbf{t}), \ldots, \lambda_\mathbf{t}(l)>$. The set of all root-to-leaf paths in $\mathbf{t}$ is denoted as $paths(\mathbf{t}) = \{p_l^{r_\mathbf{t}} | l \in \mathbf{L_t}\}$.

Let $l$ be a leaf in $\mathbf{L_t}$ and $terms(l) = \{w_1, \ldots, w_h\}$ the sequence of occurrences of textual items in $l$. Elements $w_i$ (with $i = 1..h$) are actually term stems obtained as described in Sect. 6.3. A generic n-gram $w^{(n)} = \{s_1, \ldots, s_n\}$ of length $n < h$ is any subsequence of $terms(l)$, denoted $w^{(n)} \subset terms(l)$, such that there exists an integer offset $0 \leq o \leq h - n$ and $s_i = w_{o+i}$ for each $i$ between 1 and $n$. Notation $\lambda_\mathbf{t}(p_l^{r_\mathbf{t}}).w^{(n)}$ indicates a contextualized n-gram, i.e., a sequence of $n$ adjacent term stems $w^{(n)}$ in the context of the root-to-leaf path $p_l^{r_\mathbf{t}}$. The set of all contextualized n-grams of length $n$ in $\mathbf{t}$ is indicated as $\mathcal{G}^{(n)}(\mathbf{t}) = \cup_{l \in \mathbf{L_t}, w^{(n)} \in terms(l)} \{\lambda_\mathbf{t}(p_l^{r_\mathbf{t}}).w^{(n)}\}$. In the rest of the manuscript, XML document and XML tree are used as synonyms. Also, the generic contextualized n-gram of length $n$ is indicated as $p.w^{(n)}$ to avoid cluttering notation.

### 2.2 XML features and corpus summarization

Clustering XML trees involves facing all the difficulties discussed in Sect. 1. Therefore, all approaches presented in this manuscript rely on a common intuition of projecting the original XML corpus $\mathcal{D}$ into some convenient space $\mathcal{F}$ of XML features, wherein the structural and content homogeneity of XML documents is obtained by looking at such targeted XML features, rather than by explicitly computing the pair-wise similarity of their tree-like representations.

The tree-like view of XML documents enables the identification of the XML features. In principle, such XML features can be informative of either the structure, content or both aspects of the XML trees.

Focusing only on the content of the XML trees can be accomplished by setting $\mathcal{F} = \cup_{\mathbf{t}\in\mathcal{D}, l\in\mathbf{L_t}} text(l)$, where *text*(l) is the collection of all textual items within leaf $l$. This amounts to treating $\mathcal{D}$ as a corpus of unstructured text documents, whose contents are represented through the *bag-of-words* model. However, such a choice raises the challenging issues covered in Sect. 1. The *bag-of-phrases* model allows for avoiding these issues by considering the word n-grams of a certain length $n$, which amounts to setting $\mathcal{F} = \cup_{\mathbf{t}\in\mathcal{D}, l\in\mathbf{L_t}} \{w^{(n)} | w^{(n)} \subseteq terms(l)\}$.

Instead, accounting for the structure alone of the XML trees in $\mathcal{D}$ involves choosing among various types of substructures such as, e.g., nodes, edges, paths (Costa et al. 2011, 2013; Costa and Ortale 2012b, 2013a; Joshi et al. 2003), (sub)trees (Dalamagas et al. 2006; Francesca et al. 2003; Zaki and Aggarwal 2003) and so forth. Such a choice should be the result of some reasonable compromise involving their number, discriminatory power and structural complexity. Root-to-leaf paths are a convenient choice of structural XML features, that was proven to enable highly effective, efficient and scalable (un)supervised XML classification (Costa and Ortale 2012a, 2013a; Costa et al. 2011, 2013; Joshi et al. 2003).

However, considering only the structure or the content of the XML documents generally penalizes the effectiveness of XML partitioning. Indeed, clustering by structure cannot effectively divide the input corpus $\mathcal{D}$, whenever the XML documents exhibit strongly matching or undifferentiated structures, despite meaningful differences in their textual contents. Dually, clustering by content cannot effectively divide the input corpus $\mathcal{D}$, whenever the XML documents exhibit overlapping content, despite meaningful differences in their logical structures. This suggests to focus on suitable XML features, that are truly informative of the nesting of textual content into structure across the individual XML trees. In this manuscript, we choose $\mathcal{F}$ to be the set of XML features corresponding to all of the contextualized n-grams in $\mathcal{D}$. These are especially interesting XML features, that borrow the high discriminatory power of root-to-leaf paths, allow for a controlled extent of phrase approximation and, additionally, enable a mutual refinement between the semantics of the prefixing path elements and the n-grams in the context of the root-to-leaf paths.

The selection of the set $\mathcal{F}$ of XML features enables the summarization of $\mathcal{D}$ into a $|\mathcal{F}| \times |\mathcal{D}|$ matrix $\mathbf{D}$. The latter can be real-valued or binary valued, depending on weather or not the clustering process takes advantage of the information concerning the relevance of the XML features to the individual XML documents. More precisely, XC-NMF and XCo-Clust explicitly account for feature relevance. Thus, in such approaches, the generic XML tree $\mathbf{t}$ is projected into a vector space, wherein it is represented as a vector $\overrightarrow{v}^{(\mathbf{t})}$, such that $\overrightarrow{v}^{(\mathbf{t})}$ has as many entries as the cardinality of $\mathcal{F}$ and the generic $i$th entry $\overrightarrow{v}_i^{(\mathbf{t})}$ indicates the relevance of the $i$th XML feature to the XML tree $\mathbf{t}$. The matrix $\mathbf{M} \in \mathbb{R}^{|\mathcal{F}|\times|\mathcal{D}|}$ is constructed essentially by arranging row vectors $\overrightarrow{v}^{(\mathbf{t})}$ for each $\mathbf{t}$ in $\mathcal{D}$ as columns of $\mathbf{M}$ itself. Instead, the XPart approach addresses the XML features of $\mathcal{F}$ without accounting for their relevance. Thus, the generic XML tree $\mathbf{t}$ is represented as a transaction $\mathbf{x}^{(\mathbf{t})} \subseteq \mathcal{F}$ of relevant features: the features explicitly present in $\mathbf{x}^{(\mathbf{t})}$ take value true, whereas all others assume value false. The corpus summarization resulting from the arrangement of transactions $\mathbf{x}^{(\mathbf{t})}$ (for each $\mathbf{t}$ in $\mathcal{D}$) as columns of $\mathbf{M}$ is a binary matrix $\{0,1\}^{|\mathcal{F}|\times|\mathcal{D}|}$.

### 2.3 XML feature relevance

We shall use different notions of XML feature relevance. The top-level distinction is based on whether the latter is intended to the individual XML trees or to the whole XML corpus. Let $p.w^{(n)}$ be a contextualized n-gram, $\mathcal{D}$ a corpus of XML trees and $\mathbf{t}$ a specific XML tree from $\mathcal{D}$.

*Relevance of an XML feature to an XML tree* The most basic way to quantify the relevance $s_{\mathbf{t}}(p.w^{(n)})$ of the XML feature $p.w^{(n)}$ to the XML tree $\mathbf{t}$ is by using the traditional *TFIDF* weighting scheme (Salton 1991). Accordingly,

$$s_{\mathbf{t}}(p.w^{(n)}) = TF^{(\mathbf{t})}(p.w^{(n)}) \cdot IDF^{(\mathcal{D})}(p.w^{(n)}) \tag{1}$$

where $TF^{(\mathbf{t})}(p.w^{(n)})$ is the occurrence frequency of $p.w^{(n)}$ in $\mathbf{t}$ and $IDF^{(\mathcal{D})}(p.w^{(n)})$ is the below inverse document frequency

$$IDF^{(\mathcal{D})}(p.w^{(n)}) = log \frac{|\mathcal{D}|}{|\mathcal{D}_{p.w^{(n)}}|}$$

with $\mathcal{D}_{p.w^{(n)}}$ being the subset of XML trees from $\mathcal{D}$ in which $p.w^{(n)}$ occurs.

An adaptation of the *TFIDF* weighting scheme (Salton 1991) to the peculiarities of the XML domain can be obtained by ensuring that the contextualized n-gram $p.w^{(n)}$ is actually representative of the XML tree $\mathbf{t}$, when $w^{(n)}$ occurs predominantly within the context of the root-to-leaf path $p$ of $\mathbf{t}$ (Costa and Ortale 2014). Outer occurrences in the context of either different root-to-leaf paths of $\mathbf{t}$ or XML trees other than $\mathbf{t}$ should imply a decreased discriminatory power of $p.w$ with respect to $\mathbf{t}$. Formally, let $\mathcal{G}^{(n)}_{w^{(n)}}(\mathbf{t})$ be the subset of contextualized n-grams from $\mathbf{t}$, whose sequence of textual items is $w^{(n)}$ and $\mathcal{D}_{w^{(n)}}$ the subset of XML trees from $\mathcal{D}$ in which $w^{(n)}$ occurs. The relevance $s_{\mathbf{t}}(p.w^{(n)})$ of $p.w^{(n)}$ to $\mathbf{t}$ can be defined as

$$s'_{\mathbf{t}}(p.w^{(n)}) = TF^{(\mathbf{t})}(p.w^{(n)}) \cdot C(p.w^{(n)}, \mathbf{t}) \cdot IDF^{(\mathcal{D})}(p.w^{(n)}) \cdot log \frac{|\mathcal{D}|}{|\mathcal{D}_{w^{(n)}}|} \tag{2}$$

where $log \frac{|\mathcal{D}|}{|\mathcal{D}_{w^{(n)}}|}$ penalizes the relevance of $p.w^{(n)}$ to $\mathbf{t}$, when $w^{(n)}$ occurs in XML trees other than $\mathbf{t}$. Besides, $C(p.w^{(n)}, \mathbf{t})$ weighs the relevance of $p.w^{(n)}$ to $\mathbf{t}$ based on the occurrences of $w^{(n)}$ across the distinct root-to-leaf paths of $\mathbf{t}$. In particular, $C(p.w^{(n)}, \mathbf{t})$ rewards $p.w^{(n)}$, if $w^{(n)}$ occurs in the context of few distinct root-to-leaf paths of $\mathbf{t}$. Instead, $C(p.w^{(n)}, \mathbf{t})$ penalizes $p.w^{(n)}$, if $w^{(n)}$ occurs in the context of many distinct root-to-leaf paths of $\mathbf{t}$. More precisely, $C(p.w^{(n)}, \mathbf{t})$ is directly proportional to the number of distinct root-to-leaf paths in which $w^{(n)}$ does not occurs and inversely proportional to the number of distinct root-to-leaf paths in which $w^{(n)}$ does occur, according to the following definition

$$C(p.w^{(n)}, \mathbf{t}) = \begin{cases} 1 & \text{if } |\mathcal{G}^{(n)}(\mathbf{t})| = |\mathcal{G}^{(n)}_{w^{(n)}}(\mathbf{t})| = 1 \\ \dfrac{|\mathcal{G}^{(n)}(\mathbf{t}) - \mathcal{G}^{(n)}_{w^{(n)}}(\mathbf{t})|}{|\mathcal{G}^{(n)}_{w^{(n)}}(\mathbf{t})|} & \text{otherwise} \end{cases}$$

Another definition of relevance of the XML feature $p.w^{(n)}$ to the XML tree $\mathbf{t}$ follows by refining the one of Eq. 2 to also account for the occurrence frequency of n-grams locally to

the context of the root-to-leaf paths of $\mathbf{t}$ (Costa and Ortale 2015a). Assume that $TF^{(\mathbf{t})}(p, w^{(n)})$ is the occurrence frequency of the n-gram $w^{(n)}$ within the context of the root-to-leaf path $p$ in $\mathbf{t}$. The relevance $p.w^{(n)}$ to $\mathbf{t}$ becomes

$$s''_{\mathbf{t}}(p.w^{(n)}) = TF^{(\mathbf{t})}(p.w^{(n)}) \cdot \overline{C}(p.w^{(n)}, \mathbf{t}) \cdot IDF^{(\mathcal{D})}(p.w^{(n)}) \cdot log\frac{|\mathcal{D}|}{|\mathcal{D}_{w^{(n)}}|} \tag{3}$$

where $\overline{C}(p.w^{(n)}, \mathbf{t})$ is a real-valued coefficient, that weighs the relevance of $p.w^{(n)}$ to $\mathbf{t}$ by looking at the frequency of occurrences of $w^{(n)}$ locally to the distinct root-to-leaf paths of $\mathbf{t}$. More precisely,

$$\overline{C}(p.w^{(n)}, \mathbf{t}) = \frac{TF^{(\mathbf{t})}(p, w^{(n)})}{\sum_{p' \in paths(\mathbf{t})} TF^{(\mathbf{t})}(p', w^{(n)})}$$

Notably, $\overline{C}(p.w^{(n)}, \mathbf{t})$ achieves its maximum value 1 if $w^{(n)}$ only occurs in the context of $p$. Occurrences of $w^{(n)}$ within the context of root-to-leave paths other than $p$ penalize the relevance of $p.w^{(n)}$ by lowering the value of $\overline{C}(p.w^{(n)}, \mathbf{t})$. Penalization is negligible if $w^{(n)}$ occurs predominantly within the context of $p$.

*Relevance of an XML feature to the XML corpus* Given any of the above definitions of relevance $s_{\mathbf{t}}(p.w^{(n)})$ of an XML feature $p.w^{(n)}$ to the generic XML tree $\mathbf{t}$, the relevance $s_{\mathcal{D}}(p.w^{(n)})$ of $p.w^{(n)}$ to the whole XML corpus $\mathcal{D}$ can be easily quantified (in accordance with the average *TFIDF* value introduced in Tang et al. (2005)) as

$$s_{\mathcal{D}}(p.w^{(n)}) = \frac{1}{|\mathcal{D}|} \sum_{\mathbf{t} \in \mathcal{D}} s_{\mathbf{t}}(p.w^{(n)}). \tag{4}$$

### 2.4 Problem statement

Clustering a forest $\mathcal{D} = \{\mathbf{t}_1, \ldots, \mathbf{t}_N\}$ of $N$ XML trees by content and structure is the task of forming a partition $\mathcal{P} = \{\mathcal{C}_1, \ldots, \mathcal{C}_K\}$ of nonempty clusters, such that $\mathcal{C}_i \subset \mathcal{D}$, $\mathcal{C}_i \cap \mathcal{C}_j = \emptyset$ (for all $i, j = 1, \ldots, K$ with $i \neq j$) and $\cup_i \mathcal{C}_i = \mathcal{D}$. Additionally, the degree of structural and content homogeneity exhibited by the XML trees in the same cluster is high, whereas the extent of structural and content homogeneity between XML trees within distinct clusters is low. This is accomplished differently by the XC-NMF, XCo-Clust and XPart approaches. Let $\mathcal{F}$ be a set of XML features and $\mathbf{M}$ a $|\mathcal{F}| \times |\mathcal{D}|$ matrix summarization of $\mathcal{D}$. The task fulfilled by the foresaid approach consists in partitioning $\mathcal{D}$ as follows.

- Given a number $K$ of latent topics, XC-NMF allows for learning a mapping $C_{\mathcal{D}} : \mathcal{D} \mapsto \{\mathcal{C}_1, \ldots, \mathcal{C}_K\}$ such that $C_{\mathcal{D}}(\mathbf{t}_i) = argmax_j \mathbf{V}_{ij}$, with $\mathbf{V}$ being a factor matrix (encoding the soft membership of the XML trees to the latent topics), that results from the non-negative factorization of $\mathbf{M}$ into two lower-rank matrices $\mathbf{U}$ and $\mathbf{V}^T$, such that $\mathbf{M} \approx \mathbf{U}\mathbf{V}^T$, $\mathbf{U} \in \mathbb{R}^{|\mathcal{F}| \times K}$ and $\mathbf{V}^T \in \mathbb{R}^{K \times |\mathcal{D}|}$.
- XCo-Clust simultaneously clusters $\mathcal{D}$ and $\mathcal{F}$, respectively, into $K$ and $H$ disjoint clusters through the non-negative tri-factorization of $\mathbf{M}$. This corresponds to learning two mappings $C_{\mathcal{D}}$ and $C_{\mathcal{F}}$ for the columns and rows of $\mathbf{M}$, respectively, such that $C_{\mathcal{D}} : \mathcal{D} \mapsto \{\mathcal{C}_1, \ldots, \mathcal{C}_K\}$ and $C_{\mathcal{F}} : \mathcal{F} \mapsto \{\mathcal{F}_1, \ldots, \mathcal{F}_H\}$. Here, $C_{\mathcal{D}}(\mathbf{t}_i) = argmax_j \boldsymbol{D}_{ij}$ and $C_{\mathcal{F}}(p.w) = argmax_j \boldsymbol{F}_{p.w^{(n)},j}$, with $\boldsymbol{F} \in \mathbb{R}^{|\mathcal{F}| \times H}$, $\boldsymbol{S} \in \mathbb{R}^{H \times K}$ and $\boldsymbol{D} \in \mathbb{R}^{|\mathcal{D}| \times K}$ deriving from the non-negative tri-factorization of $\boldsymbol{M}$ into $\boldsymbol{FSD}^T$, such that $\boldsymbol{M} \approx \boldsymbol{FSD}^T$.

- XPart partitions the columns of $\mathbf{M}$ by XML feature overlap, which corresponds to learning a mapping $C_{\mathcal{D}} : \mathcal{D} \mapsto \{C_1, \ldots, C_K\}$.

### 2.5 Common three-step structure of the proposed approaches

XC-NMF, XCo-Clust, and XPart partition the input XML corpus $\mathcal{D}$ by performing the three steps reported below:

- selection of a suitable set $\mathcal{F}$ of XML features;
- summarization of the original XML corpus $\mathcal{D}$ into a matrix $\mathbf{M}$;
- application of a specific processing approach to $\mathbf{M}$.

One extra post-processing step is required only in XC-NMF to derive the hard clustering of $\mathcal{D}$ into the topic clusters.

Sections 3, 4 and 5 provide a detailed coverage of the above steps, respectively, in the context of the XC-NMF, XCo-Clust, and XPart approaches.

## 3 The XC-NMF approach

This approach assumes that the XML trees of an XML corpus $\mathcal{D}$ deal with $K$ topics. In particular, each XML tree is either entirely devoted to one particular topic, or differently related to various topics. Clustering $\mathcal{D}$ thus simply involves assigning each XML tree $\mathbf{t}$ in $\mathcal{D}$ to the predominant topic in $\mathbf{t}$. The strength of association between the XML trees and the assumed topics is quantified through matrix factorization. Non-negative matrix factorization is widely used in text clustering. The adoption of such a technique for XML clustering calls for expedients with which to lower the burden of large-scale factorizations, that are generally involved in the partitioning of massive collections of text-centric XML documents.

### 3.1 XML features

The set of XML features addressed by XC-NMF is $\mathcal{F} \triangleq \cup_{\mathbf{t} \in \mathcal{D}} \mathcal{G}^{(n)}(\mathbf{t})$.

### 3.2 XML corpus summarization

Each XML tree $\mathbf{t}$ in $\mathcal{D}$ is represented as a $|\mathcal{F}|$-dimensional vector $\overrightarrow{v}^{(\mathbf{t})}$, such that generic $i$th entry $\overrightarrow{v}_i^{(\mathbf{t})}$ indicates the relevance of the $i$th element of $\mathcal{F}$ to $\mathbf{t}$. More precisely, if $f_i = p.w^{(n)}$ is the generic $i$th element (or, equivalently, XML feature) of $\mathcal{F}$ (with $i = 1, \ldots, |\mathcal{F}|$), then $\overrightarrow{v}_i^{(\mathbf{t})} = TFIDF(f_i, \mathbf{t})$ (see Eq. 1 at Sect. 2.3 for details concerning the definition of $TFIDF$).

Thus, a matrix $\mathbf{M} \in \mathbb{R}^{|\mathcal{F}| \times |\mathcal{D}|}$ can be constructed to summarize $\mathcal{D}$, by arranging row vectors $\overrightarrow{v}^{(\mathbf{t})}$ for each $\mathbf{t}$ in $\mathcal{D}$ as columns of $\mathbf{M}$ and normalizing them to unit length. The factorization of $\mathbf{M}$ covered in Sect. 3.3 is suitably exploited in Sect. 3.4 for clustering purposes.

### 3.3 NMF in the XML domain

NMF is a technique meant to factorize a non-negative matrix into the product of two non-negative factor matrices (Lee and Seung 2001; Pauca et al. 2004; Xu et al. 2003). In the context of text mining, NMF factorizes the feature-document matrix of a document corpus into the feature-factor matrix and the document-factor matrix. While the generic entry of the feature-document matrix indicates the relevance of a certain feature in the context of a particular document, the interpretation of the feature-factor and document-factor matrices involves the latent semantics of the document corpus. Precisely, the generic entry of the feature-factor matrix is the degree to which a specific feature is associated with a given topic. Analogously, the generic entry of the document-factor matrix is the degree with which an individual document belongs to a certain topic. Essentially, the application of NMF to the feature-document matrix projects the original document corpus into a $K$-dimensional semantic space, where $K$ is the number of latent topics in the corpus and each axis corresponds to a specific topic. Within such a semantic space, the original documents are represented as a linear (additive) combination of the $K$ topics and clustering can be readily achieved in a simple and intuitive manner (Xu et al. 2003). Indeed, by viewing each topic as a coherent group of semantically related documents, the original documents can be separated by assigning them to the axis (i.e., topic) with largest projection value.

The exploitation of NMF for XML clustering requires dealing with a challenging issue, that is not encountered in the traditional domain of unstructured textual documents, i.e., deriving a semantic space of latent topics from both the textual content and logical structure of XML documents. From this perspective, the matrix $\mathbf{M}$ appears to be a reasonable target for the application of NMF.

Let $K$ be the number of latent semantic topics in the input XML corpus $\mathcal{D}$, such that $K << min(|\mathcal{F}|, |\mathcal{D}|)$. NMF factorizes the corpus summarization $\mathbf{M}$ into two non-negative and lower-rank matrices $\mathbf{U}$ and $\mathbf{V}^T$, such that $\mathbf{M} \approx \mathbf{U}\mathbf{V}^T$, $\mathbf{U} \in \mathbb{R}^{|\mathcal{F}| \times K}$ and $\mathbf{V}^T \in \mathbb{R}^{K \times |\mathcal{D}|}$. The generic entry $\mathbf{U}_{ij}$ of matrix $\mathbf{U}$ is the degree to which the $i$th contextualized n-gram of $\mathcal{F}$ is associated with topic (i.e., cluster) $j$. Analogously, the generic entry $\mathbf{V}_{ij}$ of matrix $\mathbf{V}$ is the degree to which the $i$th XML tree $\mathbf{t}_i$ of $\mathcal{D}$ belongs to topic (i.e., cluster) $j$. Since $K << min(|\mathcal{F}|, |\mathcal{D}|)$, NMF can be viewed as producing a compressed approximation $\mathbf{U}\mathbf{V}^T$ of $\mathbf{M}$.

Matrices $\mathbf{U}$ and $\mathbf{V}$ are found by requiring that $\mathbf{U}\mathbf{V}^T$ should approximately factorize $\mathbf{M}$ as well as possible. Therein, one possibility consists in the minimization of the error $E(\mathbf{M}, \mathbf{U}\mathbf{V}^T) = \frac{1}{2}||\mathbf{M} - \mathbf{U}\mathbf{V}^T||_F^2$ subject to non-negativity constraints $\mathbf{U}, \mathbf{V} \geq 0$, where $||\cdot||_F^2$ is the Frobenius norm.

A prototypical multiplicative update algorithm for solving the aforesaid constrained optimization problem was proposed in Lee and Seung (2001). However, it requires processing the whole matrix $\mathbf{M}$, which is impractical with text-centric XML corpora, that generally exhibit very large values of $|\mathcal{F}|$ and $|\mathcal{D}|$.

A block-wise NMF approach for large-scale factorization is presented in Cichocki et al. (2009). Its application to $\mathbf{M}$ consists in exploiting only some random portion of $\mathbf{M}$ itself, while ignoring the rest. Assume that a selection criterion enables the identification of $R$ relevant rows and $C$ relevant columns from $\mathbf{M}$. Let $\mathbf{M}_r \in \mathbb{R}^{R \times |\mathcal{D}|}$ and $\mathbf{M}_c \in \mathbb{R}^{|\mathcal{F}| \times C}$ be two matrices formed from the selected rows and columns of $\mathbf{M}$. Besides, assume that $\mathbf{U}_r \in \mathbb{R}^{R \times K}$ and $\mathbf{V}_c^T \in \mathbb{R}^{K \times C}$ are two reduced matrices, constructed by using the same indices for the rows and columns as the ones chosen for the formation of the corresponding sub-matrices $\mathbf{M}_r$ and $\mathbf{M}_c$. The large-scale factorization implied by the minimization of the error

$\frac{1}{2}||\mathbf{M}-\mathbf{U}\mathbf{V}^T||_F^2$ subject to non-negativity constraints $\mathbf{U},\mathbf{V}\geq 0$ can thus be replaced by the sequential minimization of two linked errors, subject to suitable non-negativity constraints, in which much smaller matrices are used. In particular, the two errors are $E^{(r)}(\mathbf{M}_r,\mathbf{U}_r\mathbf{V}^T)=\frac{1}{2}||\mathbf{M}_r-\mathbf{U}_r\mathbf{V}^T||_F^2$ with fixed $\mathbf{U}_r$ and $E^{(c)}(\mathbf{M}_c,\mathbf{U}\mathbf{V}_c^T)=\frac{1}{2}||\mathbf{M}_c-\mathbf{U}\mathbf{V}_c^T||_F^2$ with fixed $\mathbf{V}_c^T$. Clearly, the non-negativity constraints are $\mathbf{U},\mathbf{V}^T\geq 0$. The sequential minimization of the foresaid errors subject to non-negativity constraints can be performed through the below (alternating least squares) updates (Cichocki et al. 2009)

$$\mathbf{V}^T \leftarrow max\left\{\epsilon,\left[\left(\mathbf{U}_r^T\cdot\mathbf{U}_r\right)^{-1}\cdot\mathbf{U}_r^T\cdot\mathbf{M}_r\right]\right\} \tag{5}$$

$$\mathbf{U} \leftarrow max\left\{\epsilon,\left[\mathbf{M}_c\cdot\mathbf{V}_c\cdot\left(\mathbf{V}_c^T\cdot\mathbf{V}_c\right)^{-1}\right]\right\} \tag{6}$$

where $\epsilon$ is a small parameter (typically, $10^{-16}$) and the *max* operator is applied element-wise (i.e., the value of every matrix entry is compared with $\epsilon$).

Algorithm 1 sketches an ANLS (*Alternating Non-negative Least-Square*) procedure (Cichocki et al. 2009) implementing the above reviewed technique for large-scale NMF. Matrices $\mathbf{M}_r$ and $\mathbf{M}_c$ are initially formed (at lines 1 and 2, respectively) through some suitable strategy for selecting rows and columns from $\mathbf{M}$. We randomly choose $R$ rows and $C$ columns from $\mathbf{M}$ with probability proportional to their squared Euclidean norm, with $R$ and $C$ such that $K<R\leq 4K$ and $K<C\leq 4K$ (Cichocki et al. 2009). $\mathbf{U}_r$ is initialized (at line 3) as a random dense non-negative matrix (other initializations can be found in Albright et al. (2006)). The algorithm then enters a loop (lines 4–10), which is reiterated until convergence or a preestablished number $I$ of iterations. The generic iteration of such a loop involves normalizing the columns of $\mathbf{U}_r$ to unit length (at line 5), computing matrix $\mathbf{V}$ through update (5) (at line 6), extracting $\mathbf{V}_c$ from $\mathbf{V}$ (at line 7), computing matrix $\mathbf{U}$ through update (6) (at line 8) as well as extracting the current $\mathbf{U}_r$ from the previously computed $\mathbf{U}$ (at line 9). Convergence is met when the distance $||\mathbf{M}-\mathbf{U}\mathbf{V}^T||_F^2$ falls below a certain error (alternative stopping criteria can be found in Cichocki et al. (2009)).

**Algorithm 1** ANLS-based non-negative factorization for large-scale matrices

LS-ANLS($\mathbf{M}, K$)
**Input:** a matrix $\mathbf{M}\in\mathbb{R}^{|\mathcal{F}|\times|\mathcal{D}|}$ and a number $K$ of latent topics;
the number $K$ of latent topics;
**Output:** $\mathbf{U}\in\mathbb{R}^{|\mathcal{F}|\times K}$ and $\mathbf{V}\in\mathbb{R}^{|\mathcal{D}|\times K}$.
1: choose $R$ rows from $\mathbf{M}$ to form matrix $\mathbf{M}_r$;
2: choose $C$ columns from $\mathbf{M}$ to form matrix $\mathbf{M}_c$;
3: randomly initialize matrix $\mathbf{U}_r$ with non-negative values;
4: **repeat**
5: normalize the columns of $\mathbf{U}_r$ to unit length;
6: compute $\mathbf{V}$ according to update (5);
7: extract $\mathbf{V}_c$ from $\mathbf{V}$;
8: compute $\mathbf{U}$ according to update (6);
9: extract $\mathbf{U}_r$ from $\mathbf{U}$;
10: **until** convergence or some pre-established number $I$ of iterations is performed

### 3.4 XML clustering

Algorithm 2 sketches the actual XC-NMF (*XML Clustering based on Non-negative Matrix Factorization*) clustering algorithm, that operates in the latent semantic space derived through the application of NMF to the matrix $\mathbf{M}$. XC-NMF receives an XML corpus $\mathcal{D}$ and

a number $K$ of latent topics as input and outputs a partition $\mathcal{P}$ of $\mathcal{D}$ consisting of $K$ clusters. Essentially, the LS-ANLS algorithm of Algorithm 1 is applied (at line 3) to the $\mathbf{M}$ (formed at line 2). Then each XML tree in $\mathcal{D}$ is assigned (at line 7) to the cluster representing the topic which that particular tree is associated to with maximum degree.

**Algorithm 2** The XC-NMF algorithm

XC-NMF($\mathbf{M}$, $K$, $n$)
**Input:** The corpus $\mathcal{D} = \{\mathbf{t}_1, \ldots, \mathbf{t}_N\}$ of XML trees;
The n-gram length $n$;
The number $K$ of latent topics;
**Output:** A partition $\mathcal{P} = \{C_1, \ldots, C_K\}$ of $\mathcal{D}$.
1: let $\mathcal{F} = \cup_{\mathbf{t} \in \mathcal{D}} \mathcal{G}^{(n)}(\mathbf{t})$;
2: let $\mathbf{M} \in \Re^{|\mathcal{F}| \times |\mathcal{D}|}$ be the matrix summarizing $\mathcal{D}$ constructed as discussed in Section 3.2;
3: let $\mathbf{U}$ and $\mathbf{V}^T$ be the factor matrices obtained through the application of the LS-ANLS procedure of Algorithm 1 to $\mathbf{M}$ with $K$ topics;
4: $C_h \leftarrow \emptyset$ for each $h = 1, \ldots, K$;
5: **for each** $i = 1, \ldots, |\mathcal{D}|$ **do**
6: let $l = argmax_j \mathbf{V}_{ij}$;
7: $C_l \leftarrow C_l \cup \mathbf{t}_i$;
8: **end for**
9: RETURN $\mathcal{P}$;

## 4 The XCo-Clust approach

XML co-clustering takes advantage of the interplay between XML documents and their respective XML features, while clustering both simultaneously. Expectedly, this is beneficial for improving the effectiveness of XML clustering.

### 4.1 XML features

The set of XML features targeted by XCo-Clust is $\mathcal{F} \triangleq \cup_{\mathbf{t} \in \mathcal{D}} \mathcal{G}^{(n)}(\mathbf{t})$. However, the input XML corpus $\mathcal{D}$ is usually a large collection of (text-centric) XML data. Thus, the cardinality of the set $\mathcal{F}$ of XML features may be very large in practical applications. This raises two issues. Firstly, the factorization of $\mathbf{M}$ may be computationally infeasible. Secondly, the inherently sparsity of $\mathbf{M}$ likely lowers the effectiveness with which the original XML corpus $\mathcal{D}$ is partitioned.

It is therefore desirable to reduce the cardinality of the set $\mathcal{F}$ through the removal of irrelevant and redundant XML features, as an attempt to reduce the curse of dimensionality and expedite co-clustering. We resort to feature selection in order to distill a subset $\mathcal{F}'$ of relevant XML features from $\mathcal{F}$, that retain their original semantic meaning (as opposed to feature extraction, that instead would identify artificial features with less clear meaning). More precisely, in order to automatically select a smaller set $\mathcal{F}'$ from $\mathcal{F}$ without introducing any tunable threshold, the average relevance $\overline{s}$ of all the XML features is defined as

$$\overline{s} = \frac{1}{|\mathcal{F}|} \sum_{p.w^{(n)} \in \mathcal{F}} s_{\mathcal{D}}(p.w^{(n)})$$

where $s_{\mathcal{D}}(p.w^{(n)})$ is the average relevance (specified by Eq. 4 at Sect. 2.3) of the contextualized n-gram $p.w^{(n)}$ to the whole XML corpus $\mathcal{D}$. Thus, the smaller set $\mathcal{F}'$ of XML features can be chosen from the original set $\mathcal{F}$ by retaining those features, whose average

relevance to the XML corpus $\mathcal{D}$ is greater than $\overline{s}$, i.e., $\mathcal{F}' = \{p.w^{(n)} \in \mathcal{F} | s_{\mathcal{D}}(p.w^{(n)}) > \overline{s}\}$. Hereafter, we will write $\mathcal{F}$ to mean $\mathcal{F}'$ (if not otherwise specified).

### 4.2 XML corpus summarization

The representation of the individual XML trees and the construction of the summarization matrix $\mathbf{M}$ are as described in Sect. 3.2 with the following two exceptions. Specifically, for each XML tree $\mathbf{t}$ from $\mathcal{D}$, the relevance to $\mathbf{t}$ of the generic $i$th XML feature $f_i$ is $\overrightarrow{v}_i^{(\mathbf{t})} = s'_{\mathbf{t}}(f_i)$ (according to the definition provided by Eq. 2 at Sect. 2.3). Moreover, $\mathbf{M}$ is not subjected to normalization.

Matrix $\mathbf{M}$ is the target for the application of non-negative matrix tri-factorization to co-cluster $\mathcal{D}$ and $\mathcal{F}$. Unlike the XC-NMF approach treated in Sect. 3, the partitioning of $\mathcal{D}$ directly follows from the non-negative tri-factorization of $\mathbf{M}$. No post-processing is required. Notation $\mathbf{M}_{r\cdot}$ and $\mathbf{M}_{\cdot c}$ will be used to denote, respectively, the $r$th row and $c$th column of $\mathbf{M}$.

### 4.3 The non-negative matrix tri-factorization procedure

The scheme of the XCo-Clust (*XML Co-Clustering*) approach in pseudo code is sketched in Algorithm 3. A set of relevant XML features $\mathcal{F}$ is used (at line 1) to summarize the input set $\mathcal{D}$ of XML trees.

The matrix $\mathbf{M}$ is then constructed to summarize $\mathcal{D}$ (at line 2). $\mathbf{M}$ is co-clustered by means of the NMTF procedure (at line 4) to simultaneously and separately partition $\mathcal{D}$ and $\mathcal{F}$ based on their interdependencies. The latter involves arbitrary initializations of the factor matrices $\mathbf{F}$ and $\mathbf{D}$ and, thus, its execution is reiterated multiple times (lines 3–6). Partitioning effectiveness is computed after each execution of the NMTF procedure (at line 5) according to Sect. 6.4 and the average partitioning effectiveness across the different executions of the NMTF co-clustering procedure is eventually reported (at line 9).

**Algorithm 3** The XCo-Clust approach

XCo-Clust($\mathcal{D}$,$K$,$H$,$R$,$n$)
**Input:** A forest $\mathcal{D} = \{\mathbf{t}_1, \ldots, \mathbf{t}_N\}$ of XML trees;
The number $K$ of clusters of XML documents to find in $\mathcal{D}$;
The number $H$ of clusters of XML features to find in $\mathcal{D}$;
The overall number $R$ of reiterations of the NMTF procedure;
The n-gram length $n$;
**Output:** The macro-averaged and micro-averaged effectiveness on $\mathcal{D}$
1: Let $\mathcal{F} = \{f_1, \ldots, f_M\}$ be a set of XML features chosen from $\cup_{\mathbf{t} \in \mathcal{D}} \mathcal{G}^{(n)}(\mathbf{t})$ as described in Section 4.1;
2: let $\mathbf{M} \in \mathbb{R}^{M \times N}$ be the matrix summarizing $\mathcal{D}$ constructed as discussed in Section 5.2;
3: **for** $r = 1, \ldots, R$ **do**
4: Let $\mathbf{FSD}^T$ be the factorization of $\mathbf{M}$ produced by NMTF($\mathbf{M}$,K,H);
5: Let $Macro\text{-}averaged\ purity^{(r)}$ and $Micro\text{-}averaged\ purity^{(r)}$ (as discussed in Sec. 6.4) be the effectiveness at iteration $r$;
6: **end for**
7: $Macro\text{-}averaged\ purity \leftarrow \frac{1}{R} \sum_{r=1}^{R} Macro\text{-}averaged\ purity^{(r)}$;
8: $Micro\text{-}averaged\ purity \leftarrow \frac{1}{R} \sum_{r=1}^{R} Micro\text{-}averaged\ purity^{(r)}$;
9: **return** $Macro\text{-}averaged\ purity$ and $Micro\text{-}averaged\ purity$;

The NMTF procedure (at line 4 of Algorithm 3) could in principle adopt one of a wide variety of methods to co-cluster $\mathcal{D}$ and $\mathcal{F}$ based on their mutual relationships summarized by $\mathbf{M}$, such as, e.g., Cho et al. (2004), Costa et al. (2008), Dhillon (2001), Dhillon et al.

(2003), Long et al. (2006), Shan and Banerjee (2008), Song et al. (2010) and Wang et al. (2009). The focus of this manuscript is on non-negative matrix tri-factorization (Ding et al. 2006; Li et al. 2010; Wang et al. 2011), which has gained increasing attention because of its mathematical elegance and effectiveness. The idea behind the non-negative tri-factorization of $\mathbf{M} \in \mathbb{R}^{|\mathcal{F}|\times|\mathcal{D}|}$ is to decompose the latter into the product $\mathbf{FSD}^T$, such that $\mathbf{M} \approx \mathbf{FSD}^T$, where $\mathbf{F} \in \mathbb{R}^{|\mathcal{F}|\times H}$ is indicative of the XML feature clustering, $\mathbf{D} \in \mathbb{R}^{|\mathcal{D}|\times K}$ is indicative of XML document clustering and $\mathbf{S} \in \mathbb{R}^{H\times K}$ increases the degrees of freedom to enable accurate lower dimensional approximations of $\mathbf{M}$ accounting for the different scales of $\mathbf{M}$, $\mathbf{F}$ and $\mathbf{D}$ (Ding et al. 2006; Wang et al. 2011). In general, various decompositions addressing as many aspects of co-clusterings can be obtained depending on the constraints placed on the matrices $\mathbf{M}$, $\mathbf{F}$, $\mathbf{S}$ and $\mathbf{D}$. However, the challenging issue in practical applications is the identification of constraints leading to optimization problems, that can be efficiently solved. A known limitation of conventional non-negative matrix tri-factorization is that such optimization problems are often solved through iterative algorithms, that perform a large amount of intensive matrix multiplications at each iteration step. This makes co-clustering computationally inefficient and unscalable to process large-scale data matrices, which are commonly encountered in the XML domain, especially with massive text-centric XML corpora, where the size of the input data matrix $\mathbf{M}$ is determined by the very large number of XML features (resulting from the combination of structure and content information) and XML documents.

An efficient technique for large-scale matrix tri-factorization is presented in Wang et al. (2011). It is aimed to compute a three-factor decomposition $\mathbf{FSD}^T$, in which $\mathbf{F}$ and $\mathbf{D}$ are cluster indicator matrices (i.e., binary matrices whose rows individually sum to 1). These are particular non-negative matrices, with which the targeted optimization problem can be divided into simpler subproblems requiring much less matrix multiplications. Such an appealing feature is clearly beneficial to make co-clustering scalable to efficiently process large-scale input data matrices and motivates the adoption of the technique in Wang et al. (2011) as the NMTF procedure of the XCo-Clust approach.

A brief review of the exploited NMTF is provided next for the sake of self-containment. The interested reader is referred to Wang et al. (2011) for further details.

Let $\mathbf{I}^{R\times C}$ be a generic $R \times C$ binary matrix. $\mathbf{I}$ is a cluster indicator matrix if its rows indicate hard cluster membership and, thus, individually exhibit one and only one entry equal to 1, i.e., $\sum_{j=1}^{C} \mathbf{I}_{rj} = 1$ for each $1 \le r \le R$. The set of all possible cluster indicator matrices is denoted as $\mathcal{I} = \{\mathbf{I}^{R\times C} | R, C > 0 \text{ and } \sum_{j=1}^{C} \mathbf{I}_{rj} = 1 \text{ for each } 1 \le r \le R\}$.

In the context of Algorithm 3, the NMTF procedure aims to minimize the below objective function

$$E = ||\mathbf{M} - \mathbf{FSD}^T||^2$$

under the constraints $\mathbf{F}^{|\mathcal{F}|\times H} \in \mathcal{I}$ and $\mathbf{D}^{|\mathcal{D}|\times K} \in \mathcal{I}$ (Wang et al. 2011).

The solution to the above optimization problem is computed through the iterative procedure sketched in Algorithm 4, that operates as follows. Initially, *NMTF* arbitrarily initializes the cluster indicator matrices $\mathbf{F}$ and $\mathbf{D}$ (at line 1). Then, it enters a loop (lines 2–6) meant to reiterate the updates of $\mathbf{S}$, $\mathbf{F}$ and $\mathbf{D}$ until convergence. In particular, $\mathbf{S}$ is updated (at line 1) through

$$\mathbf{S} = (\mathbf{F}^T\mathbf{F})^{-1}\mathbf{F}^T\mathbf{M}\mathbf{D}(\mathbf{D}^T\mathbf{D})^{-1} \tag{7}$$

Matrices $\mathbf{F}$ and $\mathbf{D}$ are instead easily updated by taking advantage of the cluster indicator constraints. These allow to decouple the original optimization problems behind the computation of $\mathbf{F}$ and $\mathbf{D}$ into simpler subproblems involving the enumeration of vector norms rather than matrix multiplications. More precisely, for each $1 \leq i \leq |\mathcal{D}|$ and $1 \leq j \leq K$, $\mathbf{D}_{ij}$ is updated as shown next

$$\mathbf{D}_{ij} = \begin{cases} 1 & \text{if } j = \arg\min_c ||\mathbf{M}_{\cdot i} - (\mathbf{F}\mathbf{S})_{\cdot c}||^2 \\ 0 & \text{otherwise} \end{cases} \tag{8}$$

Analogously, for each $1 \leq i \leq |\mathcal{F}|$ and $1 \leq j \leq H$, $\mathbf{F}_{ij}$ is updated by means of the formula below

$$\mathbf{F}_{ij} = \begin{cases} 1 & \text{if } i = \arg\min_r ||\mathbf{M}_{j\cdot} - (\mathbf{S}\mathbf{D}^T)_{r\cdot}||^2 \\ 0 & \text{otherwise} \end{cases} \tag{9}$$

**Algorithm 4** The NMTF procedure

NMTF($\mathbf{M}$,$K$,$H$)
**Input:** A matrix $\mathbf{M} \in \mathbb{R}^{|\mathcal{F}| \times |\mathcal{D}|}$;
**Output:** $\mathbf{F} \in \mathbb{R}^{|\mathcal{F}| \times H}$ and $\mathbf{D} \in \mathbb{R}^{|\mathcal{D}| \times K}$.
1: initialize $\mathbf{F}$ and $\mathbf{D}$ as arbitrary cluster indicator matrices;
2: **repeat**
3: update $\mathbf{S}$ by means of Eq. 7;
4: update $\mathbf{D}$ by means of Eq. 8;
5: update $\mathbf{F}$ by means of Eq. 9;
6: **until** convergence

## 5 The XPart approach

XPart (*XML Partitioning*) is a transactional approach to XML clustering. The idea behind XPart is to project the input corpus $\mathcal{D}$ into a space of boolean features, wherein the individual XML trees are separated by the homogeneity of their respective transactional representations.

### 5.1 XML features

Let $\mathcal{G}^{(n)} = \cup_{\mathbf{t} \in \mathcal{D}} \mathcal{G}^{(n)}(\mathbf{t})$ be the set of all contextualized n-grams of length $n$ in the XML corpus $\mathcal{D}$. XPart projects $\mathcal{D}$ into a space $\mathcal{F} \triangleq \{\mathcal{F}_{p.w^{(n)}} | p.w^{(n)} \in \mathcal{G}^{(n)}\}$, such that the generic feature $\mathcal{F}_{p.w^{(n)}}$ is a boolean attribute, whose value indicates the presence/absence of the contextualized n-gram $p.w^{(n)}$ of $\mathcal{G}^{(n)}$ within the individual XML trees.

Since $\mathcal{D}$ is generally a large corpus of (text-centric) XML documents, a smaller set of relevant features is preliminarily distilled from the ones corresponding to all elements of $\mathcal{G}^{(n)}$ through feature selection. The latter is performed as described in Sect. 4.1, apart from

the adopted definition of the XML feature relevance, that in XPart is instead quantified through Eq. 3 at Sect. 2.3.

### 5.2 XML corpus summarization

Each XML tree $\mathbf{t}$ from $\mathcal{D}$ is represented as a transaction $\mathbf{x}^{(\mathbf{t})} \subseteq \mathcal{F}$, wherein the value of each attribute $\mathcal{F}_{p.w^{(n)}}$ within $\mathbf{x}^{(\mathbf{t})}$ is 1 (or, equivalently, *true*) if $p.w^{(n)}$ is a contextualized n-gram of $\mathbf{t}$, and 0 (or, equivalently, *false*) otherwise. Hence, $\mathbf{x}^{(\mathbf{t})}$ can be modeled as a proper subset of $\mathcal{F}$, namely $\mathbf{x}^{(\mathbf{t})} \triangleq \{\mathcal{F}_{p.w^{(n)}} \in \mathcal{F} | p.w^{(n)} \in \mathcal{G}^{(n)}(\mathbf{t})\}$, with the meaning that the features explicitly present in $\mathbf{x}^{(\mathbf{t})}$ take value true, whereas the others assume value false.

The corpus $\mathcal{D}$ is summarized into the matrix $\mathbf{M}$ by arranging the transactions $\mathbf{x}^{(\mathbf{t})}$ for each XML tree from $\mathcal{D}$ as columns of $\mathbf{M}$. In the following, $\mathbf{D} = \mathbf{M}^T$ will be conveniently used to intend $\mathbf{M}^T$ as a transactional dataset, i.e., a collection of transactions (rows in $\mathbf{D}$) over the selected features (columns in $\mathbf{D}$), which is a more intuitive view of the summarized XML corpus in transactional clustering.

### 5.3 Transactional clustering

The scheme of the XPart approach is sketched in Algorithm 5. XPart initially projects (lines 2–4) the individual XML trees within the input forest $\mathcal{D}$ into a space of clustering features in $\mathcal{F}$, consisting of all distinct contextualized n-grams of the XML trees (line 1). Such a projection results in the transactional representation $\mathbf{D}$ of the original forest $\mathcal{D}$ (line 4).

**Algorithm 5** The XPart approach

XPart($\mathcal{D}$,$n$)
**Input:** a forest $\mathcal{D} = \{\mathbf{t}_1, \ldots, \mathbf{t}_N\}$ of XML trees;
**Input:** the length $n$ of word n-grams;
**Output:** a partition $\mathcal{P}$ of $\mathcal{D}$;
1: let $\mathcal{S} \leftarrow \cup_{t_i \in \mathcal{D}} \mathcal{G}^{(n)}(\mathbf{t}_i)$ be the set of all contextualized n-grams in $\mathcal{D}$;
2: let $\mathbf{x}^{(\mathbf{t}_i)} \leftarrow \{p.w^{(n)} \in \mathcal{S} | p.w^{(n)} \in \mathcal{G}^{(n)}(\mathbf{t}_i)\}$ for each $i = 1, \ldots, N$;
3: let $\mathbf{M}$ be the summarization of $\mathcal{D}$ constructed as discussed in Section 5.2;
4: $\mathbf{D} \leftarrow \mathbf{M}^T$;
5: $\mathcal{P} \leftarrow$ Generate-Clusters($\mathbf{D}$);
6: RETURN $\mathcal{P}$;

Looking for clusters in the transactional setting $\mathbf{D}$ involves dealing with three challenging issues. Firstly, transactions tend to form different clusters on distinct subsets of XML features, which penalizes the effectiveness of clustering and exacerbates its time requirements. Secondly, poor scalability with both the number and the dimensionality of transactions is usually a major limitation. Thirdly, an underestimation (resp. overestimation) of the number of groups to isolate in $\mathbf{D}$ misses (resp. uncovers) actual (resp. artificial) clusters in $\mathcal{D}$.

In order to deal with the above issues, $\mathbf{D}$ is partitioned (line 5) through Generate-Clusters (Cesario et al. 2007), i.e., an effective and parameter-free algorithm for transactional clustering, that automatically uncovers a natural number of clusters in $\mathbf{D}$.

Generate-Clusters is succinctly reviewed next to provide an understanding in the XML domain.

Algorithm 6 reports the pseudo code of the Generate-Clusters algorithm. The latter starts with a partition $\mathcal{P}$ consisting of one cluster, that includes the whole transactional

dataset **D** (line L1). At the heart of the algorithm is a loop (lines L2-L15), aimed to improve the currently discovered partition by cluster separation. More precisely, a cluster (chosen at line L4) is separated into two child clusters (line L5) and the quality of the newly resulting partition is evaluated (lines L6-L13). If the quality of the obtained partition is improved with respect to the quality of the partition including the unseparated cluster, the separation of whole clusters halts (line L10) and the partition is updated by replacing the original whole cluster with the child clusters (line L8). Otherwise, these are ignored and the next whole cluster is separated.

The PARTITION-CLUSTER procedure swaps (line L5) the membership of each transaction $\mathbf{x}^{(\mathbf{t})} \in \mathcal{C}_i \cup \mathcal{C}$, if this improves the degree of content and structural homogeneity of the two clusters.

*Quality*$(\mathcal{C})$ measures the degree of content and structural homogeneity within a cluster $\mathcal{C}$. Formally,

$$Quality(\mathcal{C}) = \Pr(\mathcal{C}) \sum_{p.w^{(n)} \in \mathcal{S}} \left[ \Pr(p.w^{(n)}|\mathcal{C})^2 - \Pr(p.w^{(n)}|\mathbf{D})^2 \right]$$

where summation is over the set $\mathcal{S}$ (defined at line 1 of Algorithm 5) of all distinct contextualized n-grams in the underlying XML collection. Notably, $\Pr(p.w^{(n)}|\mathcal{C})^2$ is the relative strength of the XML feature $p.w^{(n)}$ within $\mathcal{C}$ and $\Pr(\mathcal{C})$ is the relative strength of cluster $\mathcal{C}$. Accordingly, a cluster $\mathcal{C}$ exhibits a high quality if it includes a subset of relevant XML features whose occurrence frequency therein is significantly higher than in the whole dataset **D**.

The STABILIZE-CLUSTERS procedure aims to improve the overall partition quality *Quality*$(\mathcal{P})$ by placing each transaction in the most suitable cluster among the ones in the partition. *Quality*$(\mathcal{P})$ takes into account the homogeneity of clusters as well as their compactness according to the below definition

$$Quality(\mathcal{P}) = \sum_{\mathcal{C} \in \mathcal{P}} \Pr(\mathcal{C}) Quality(\mathcal{C})$$

**Algorithm 6** The GENERATE-CLUSTERS procedure

GENERATE-CLUSTERS($\mathbf{D}$)
**Input:** A set $\mathbf{D} = \{\mathbf{x}^{(\mathbf{t}_1)}, \ldots, \mathbf{x}^{(\mathbf{t}_N)}\}$ of transactions corresponding to XML trees;
**Output:** A partition $\mathcal{P} = \{\mathcal{C}_1, \ldots, \mathcal{C}_k\}$ of clusters of transactions corresponding to XML trees;
L1: let $\mathcal{P} \leftarrow \{\mathbf{D}\}$;
L2: **repeat**
L3: Generate a new cluster $\mathcal{C}$ of transactions, initially empty;
L4: **for each** cluster $\mathcal{C}_i \in \mathcal{P}$ **do**
L5: PARTITION-CLUSTER($\mathcal{C}_i, \mathcal{C}$);
L6: $\mathcal{P}' \leftarrow \mathcal{P} \cup \{\mathcal{C}\}$;
L7: **if** $Quality(\mathcal{P}) < Quality(\mathcal{P}')$ **then**
L8: $\mathcal{P} \leftarrow \mathcal{P}'$;
L9: STABILIZE-CLUSTERS($\mathcal{P}$);
L10: **break**
L11: **else**
L12: Restore all $\mathbf{x}^{(\mathbf{t}_j)} \in \mathcal{C}$ into $\mathcal{C}_i$;
L13: **end if**
L14: **end for**
L15: **until** no further cluster $\mathcal{C}$ can be generated
L16: RETURN $\mathcal{P}$;

## 6 Experimental evaluation

We here study and compare the performance of XC-NMF, XCo-Clust and XPart on real-world XML corpora in terms of both XML clustering effectiveness and scalability.

### 6.1 XML corpora

Two real-world XML corpora, namely *Wikipedia* and *Sigmod*, were adopted as standard benchmark datasets. Both have been largely used in the literature for assessing the approaches to XML classification and clustering. XC-NMF, XCo-Clust and XPart are compared with state-of-the-art competitors, whose effectiveness on the chosen corpora is known from previous studies.

*Wikipedia* was the reference collection of XML documents proposed for the task of XML clustering by both content and structure in the context of the XML Mining Track at INEX 2007 (Denoyer and Gallinari 2008). The XML documents of the *Wikipedia* corpus represent 47, 397 very long articles from the homonymous digital encyclopedia and are organized into 21 classes (or thematic categories), each corresponding to a distinct Wikipedia Portal.

The *Sigmod* corpus consists of 140 XML documents complying to two different structural class DTDs, namely `IndexTermsPage` and `OrdinaryIssuePage`. Originally, this corpus was used to measure the effectiveness of the approaches to XML clustering at grouping XML documents by structure alone (e.g., in) (Aggarwal et al. 2007; Costa et al. 2013). However, given the reduced number of structural classes, this was not deemed to be a challenging task. Therefore, we consider a classification of the *Sigmod* corpus into 5 classes (Kutty et al. 2009b). These were obtained to add complexity to the experimental evaluation, by using expert knowledge to group the underlying XML documents on the basis of their structure and content-based similarity.

As it is highlighted in Kutty et al. (2009b), *Wikipedia* and *Sigmod* represent an especially interesting selection of XML corpora, that allows for studying the effectiveness of the approaches to XML clustering by both content and structure on data sets with opposite characteristics. Indeed, *Wikipedia* is a very-large collection of schema-less text-centric XML documents, whose trees exhibit deeper structure and high out degree, whereas *Sigmod* consists of a much smaller number of XML documents conforming to two distinct schema definitions. Table 1 summarizes some major statistics of the *Wikipedia* and *Sigmod* corpora. Figure 1 shows the percentages of XML documents in the chosen XML corpora with different sizes (measured in bytes). Additionally, the plots in Figs. 2, 3, 4 and 5 illustrate the distributions and the cumulative distributions of the contextualized n-grams within *Wikipedia* and *Sigmod*.

Notably, synthetic XML corpora were not used for experimental purposes, since these do not generally provide coherent textual data in natural language.

**Table 1** Characteristics of the chosen XML corpora (abbreviation *cntx* means contextualized)

| XML corpus | Size | Classes | Max. out degree | Max. tree depth | Distinct root-to-leaf paths |
|---|---|---|---|---|---|
| *Wikipedia* | 47,397 | 21 | 1776 | 48 | 18,839 |
| *Sigmod* | 140 | 5 | 29 | 8 | 33 |

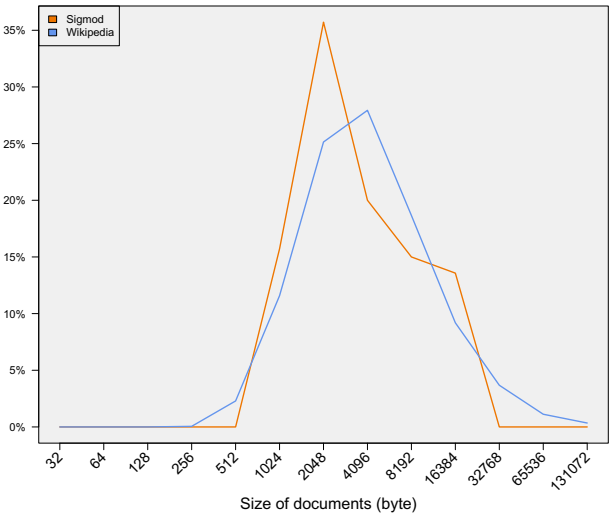


**Fig. 1** Fraction of XML documents in the chosen XML corpora across size

## 6.2 Competitors

XC-NMF, XCo-Clust and XPart are compared against state-of-the-art competitors, whose performances on *Wikipedia* and *Sigmod* are known from the literature. More precisely, the chosen competitors are XCFS (Kutty et al. 2009b), HCX (Kutty et al. 2009a), CRP as well as 4RP (Yao and Zerida 2007), SOM (Hagenbuchner et al. 2008) and LSK (Tran et al. 2008). The performances of XCFS, CRP, 4RP, SOM and LSK are reported from Kutty et al. (2009b). The results achieved by HCX are reported from Kutty et al. (2009a).

## 6.3 Preprocessing

The very large cardinality of the set of content items $\cup_{\mathbf{t}\in\mathcal{D}} terms(\mathbf{t})$ is likely a concern as far as the memory footprint, effectiveness, time efficiency and scalability of the clustering process are concerned. Therefore, the overall number of distinct items in the leaves of the available XML trees is preliminarily reduced through token extraction (numbers and words with less than 3 characters are discarded), stop-word removal (Fox 1992) and stemming (Porter 1980). The notion of content item is henceforth used to mean a word stem.

## 6.4 Evaluation measures

The effectiveness with which the chosen XML corpora are partitioned is measured in terms of *purity*. The latter measure was used in the context of the Mining Track at INEX 2007 (Denoyer and Gallinari 2008) and is widely exploited in the literature.

Purity is an external quality measure, that assumes knowledge of a predefined classification of the XML documents into a certain number $k$ of natural classes. Therefore, we study the partitioning effectiveness of the different competitors over collections of XML documents with known class labels and analyze the correspondence between the discovered and natural classifications. The available class labels are hidden to the instantiations of the XC-NMF, XCo-Clust and XPart approaches. A partition $\mathcal{P} = \{C_1, \ldots, C_l\}$ of an input

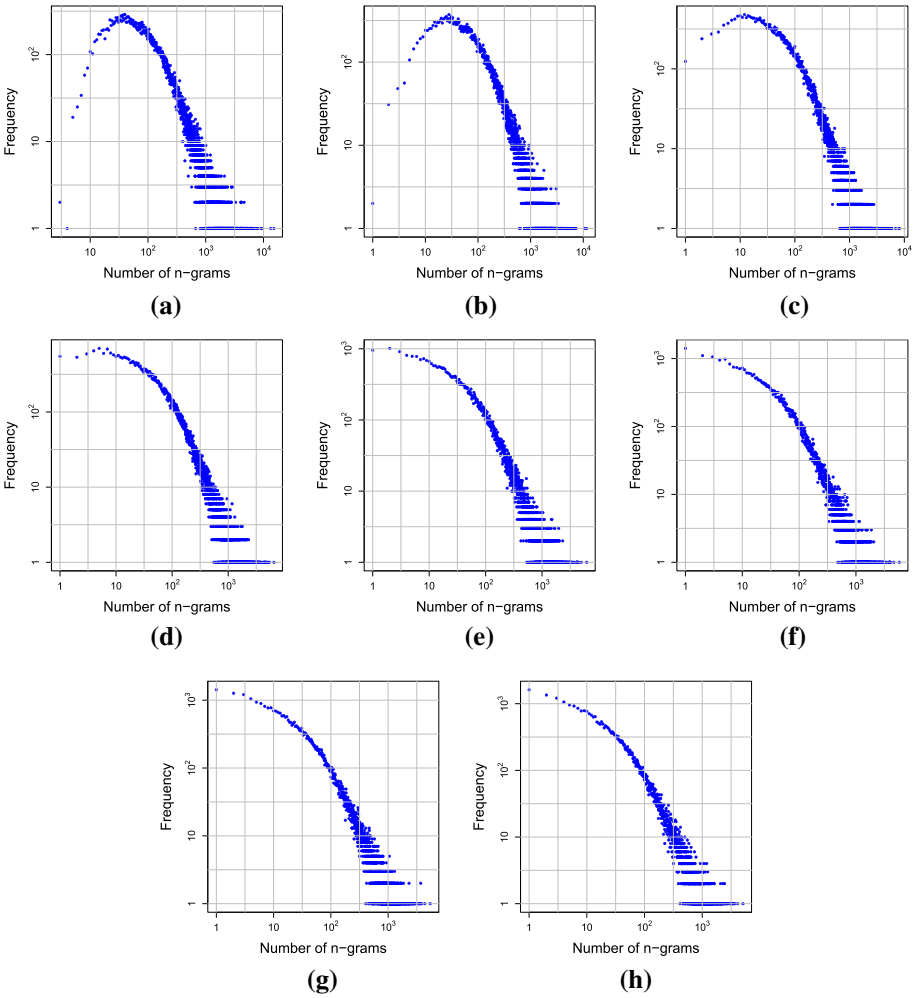


**Fig. 2** Distributions of contextualized n-grams of length from 1 to 8 within *Wikipedia*. **a** Length 1, **b** length 2, **c** length 3, **d** length 4, **e** length 5, **f** length 6, **g** length 7, **h** length 8

XML corpus $\mathcal{D}$ can be summarized into a contingency table $\mathbf{T}$, where columns represent discovered clusters and rows represent natural classes. Each entry $\mathbf{T}_{ij}$ indicates the number of XML documents in $\mathcal{D}$, that are assigned to cluster $C_j$ and actually belong to the natural class $\mathbf{C}_i$, with $1 \leq i \leq k$. Intuitively, each cluster $C_j$ corresponds to the class $\mathbf{C}_i$ that is best represented in $C_j$, i.e., such that $\mathbf{T}_{ij}$ is maximal. For any cluster $C_j$, the index $h(j)$ of the class with maximal $\mathbf{T}_{ij}$ is defined as $h(j) = argmax_i \mathbf{T}_{ij}$. Purity for a cluster $C_j$ is a measure of the degree to which $C_j$ contains XML documents primarily from $\mathbf{C}_{h(j)}$ (Algergawy et al. 2011). Formally, $Purity(C_j) = \frac{|\mathbf{C}_{h(j)}|}{|C_j|}$. Macro-averaged purity and micro-averaged purity extend the notion of purity for a single cluster to a whole partition $\mathcal{P}$. Precisely, macro-averaged purity for a partition $\mathcal{P}$ is defined as

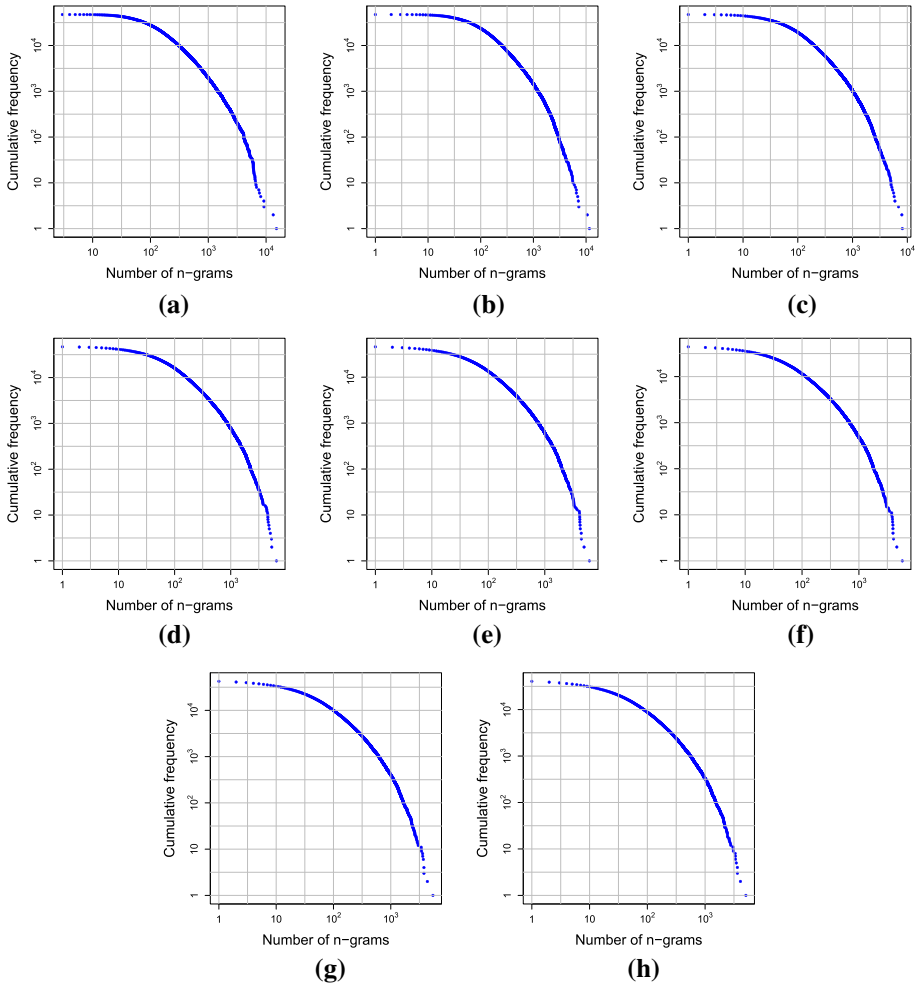


**Fig. 3** Cumulative distributions of contextualized n-grams of length from 1 to 8 within *Wikipedia*. **a** Length 1, **b** length 2, **c** length 3, **d** length 4, **e** length 5, **f** length 6, **g** length 7, **h** length 8

$$\textit{Macro-averaged purity}\,(\mathcal{P}) = \frac{1}{h}\sum_{C\in\mathcal{P}} \textit{Purity}(C)$$

Macro-averaged purity is an arithmetic mean, that assigns a same weight to each cluster. Instead, micro-averaged purity weighs each cluster by a weight proportional to the size of the cluster itself, i.e.,r

$$\textit{Micro-averaged purity}\,(\mathcal{P}) = \frac{\sum_{C\in\mathcal{P}} |C| \cdot \textit{Purity}(C)}{N}$$

Obviously, micro-averaged purity is more strongly influenced by larger clusters.

Both micro-averaged purity and macro-averaged purity are measured in our experiments. Larger values of such measures are indicative of higher partitioning effectiveness.

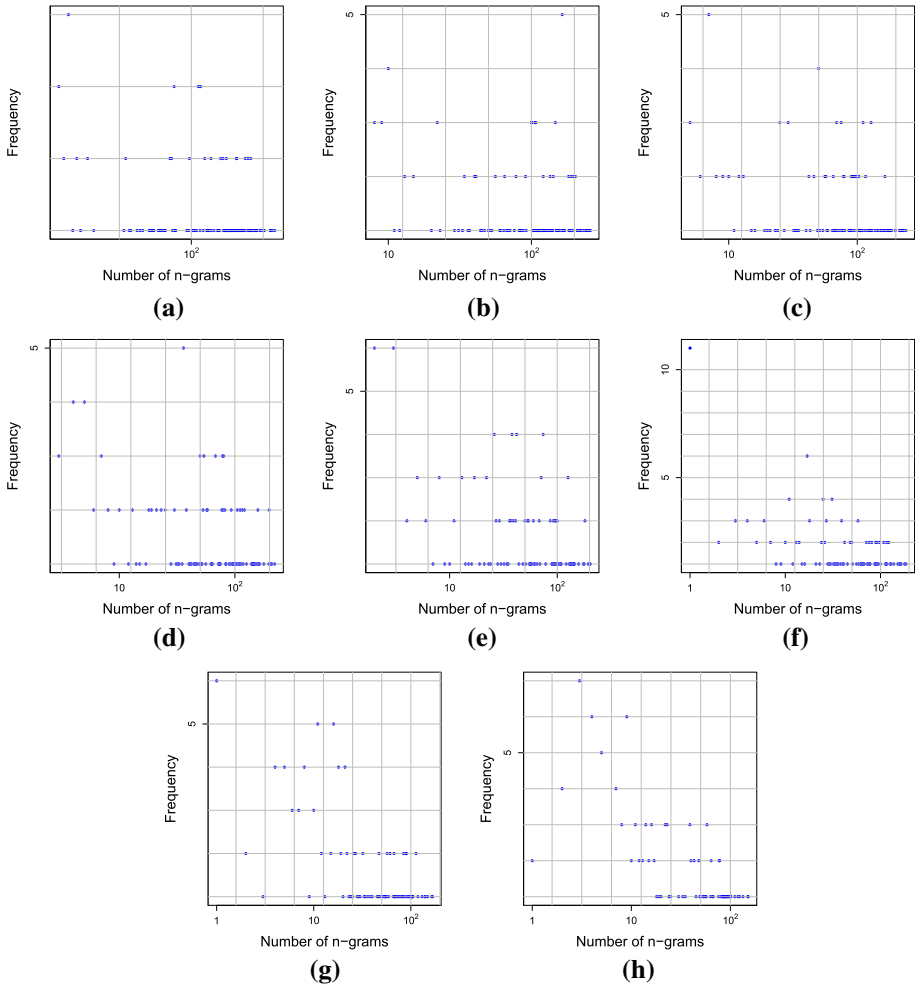


**Fig. 4** Distributions of contextualized n-grams of length from 1 to 8 within *Sigmod*. **a** Length 1, **b** length 2, **c** length 3, **d** length 4, **e** length 5, **f** length 6, **g** length 7, **h** length 8

### 6.5 Partitioning effectiveness

To start with, we investigate the effectiveness of XC-NMF, XCo-Clust and XPart when the length of contextualized n-grams is set to 1. This essentially amounts to studying the performance of the most basic instances of the foresaid approaches, i.e., those focused on the manipulation of unigrams in the context of root-to-leaf paths. The input parameters of XC-NMF and XCo-Clust are empirically set as detailed in Table 2, whereas no further parameter tuning is required by XPart. Table 3 summarizes the effectiveness of all competitors. XCo-Clust exhibits an overcoming macro-averaged purity, which is due to the exploitation of the interplay between XML documents and their respective XML features, while clustering both simultaneously.

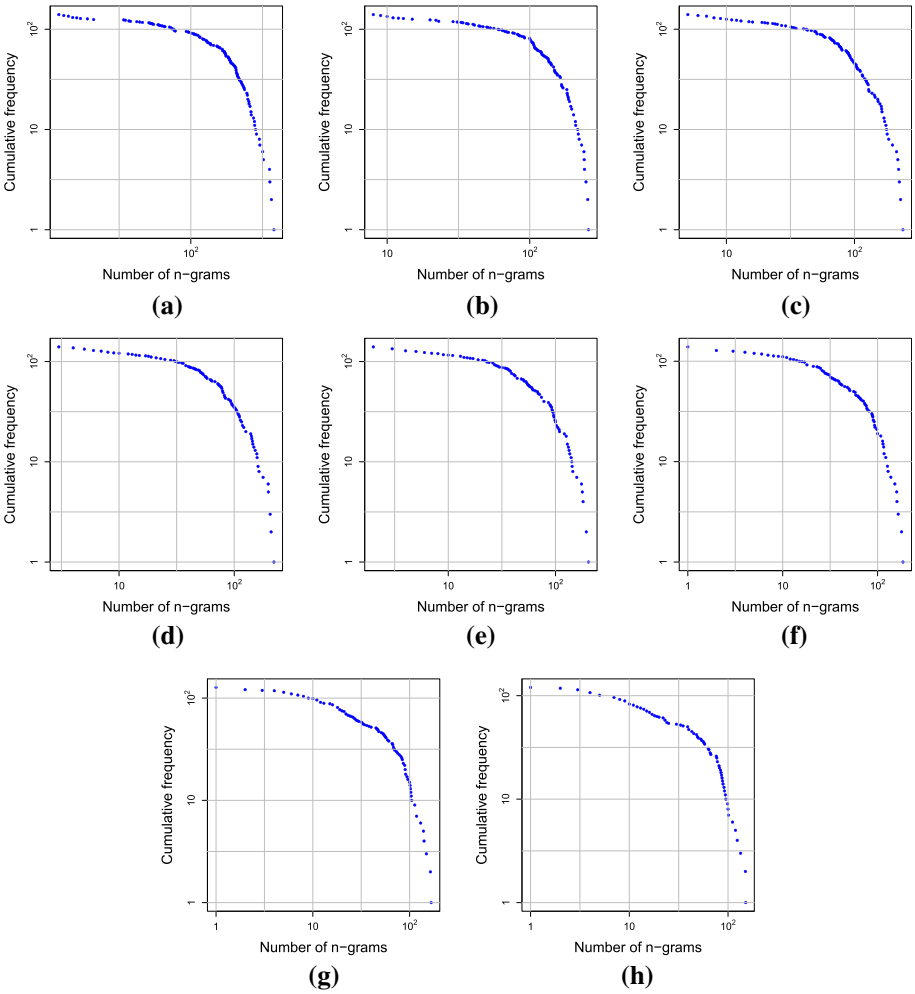


**Fig. 5** Cumulative distributions of contextualized n-grams of length from 1 to 8 within *Sigmod*. **a** Length 1, **b** length 2, **c** length 3, **d** length 4, **e** length 5, **f** length 6, **g** length 7, **h** length 8

We expect to observe an improvement in the partitioning effectiveness of XC-NMF, XCo-Clust and XPart, when approximate phrases are taken into account within root-to-leaf paths rather than simpler unigrams. In other words, when the length $n$ of contextualized n-grams is set to some reasonable value larger than 1, the clustering approaches should exhibit a higher effectiveness in partitioning the chosen XML corpora, being able to better capture the meaning of contextualized phrases. To shed light on this aspect, we study how the effectiveness of XC-NMF, XCo-Clust and XPart varies, when $n$ ranges in the empirically-determined interval [2, 8]. Again, the input parameters of XC-NMF and XCo-Clust are set to the values reported in Table 2. To the best of our knowledge, this test is new to the XML domain.

Figure 6a–d illustrates the relative performance of the individual approaches on the chosen XML corpora. Notice that the performance corresponding to $n = 1$ is reported from

**Table 2** Empirical settings for the input-parameters of the XC-NMF and XCo-Clust approaches

| Approach | Parameter | Description | Value | |
|---|---|---|---|---|
| XC-NMF | $K$ | Number of topic clusters (or, equivalently, XML document clusters) | *Wikipedia* | 21 |
| | | | *Sigmod* | 5 |
| | $I$ | Pre-established number of iterations | | 1000 |
| XCo-Clust | $K$ | Number of clusters of XML documents | *Wikipedia* | 21 |
| | | | *Sigmod* | 5 |
| | $H$ | Number of clusters of XML features | *Wikipedia* | 500 |
| | | | *Sigmod* | 50 |
| | $R$ | Number of reiterations of the co-clustering procedure | | 10 |

**Table 3** XML clustering effectiveness observed when accounting for contextualized unigrams

| Corpus | Competitor | Output clusters | Micro-averaged purity | Macro-averaged purity |
|---|---|---|---|---|
| *Wikipedia* | XCo-Clust | 21 | 0.58 | 0.75 |
| | XC-NMF | 21 | 0.45 | 0.60 |
| | XPart | 21 | 0.48 | 0.60 |
| | XCFS (Kutty et al. 2009b) | 21 | 0.58 | 0.64 |
| | HCX (Kutty et al. 2009a) | 21 | 0.59 | 0.66 |
| | CRP (Yao and Zerida 2007) | 21 | 0.44 | 0.49 |
| | 4RP (Yao and Zerida 2007) | 21 | 0.42 | 0.49 |
| | SOM (Hagenbuchner et al. 2008) | 21 | 0.27 | 0.27 |
| | LSK (Tran et al. 2008) | 21 | 0.37 | 0.40 |
| *Sigmod* | XCo-Clust | 5 | 0.91 | 0.95 |
| | XC-NMF | 5 | 0.83 | 0.92 |
| | XPart | 5 | 0.58 | 0.69 |
| | XCFS (Kutty et al. 2009b) | 5 | 0.82 | 0.49 |
| | HCX (Kutty et al. 2009a) | 5 | 0.89 | 0.64 |

Table 3 for convenience. It is evident that both the micro- and macro-averaged purity attained by XC-NMF, XCo-Clust and XPart increase on the chosen XML corpora, as long as contextualized n-grams of suitable length are used as XML features. In particular, XPart exhibits a remarkable improvement with respect to the performance in Table 3. Nonetheless, by taking advantage of the interplay between XML documents and their respective XML features, XCo-Clust is the most effective instantiation of XML clustering by structure-constrained phrases, whose validity is overall confirmed by the observed performance of XC-NMF, XCo-Clust and XPart. Notably, for certain values of the length of contextualized n-grams, the effectiveness of XCo-Clust and XPart overcomes the one of all competitors in Table 3, with XCo-Clust being the best performer among all competitors on the tested XML corpora.

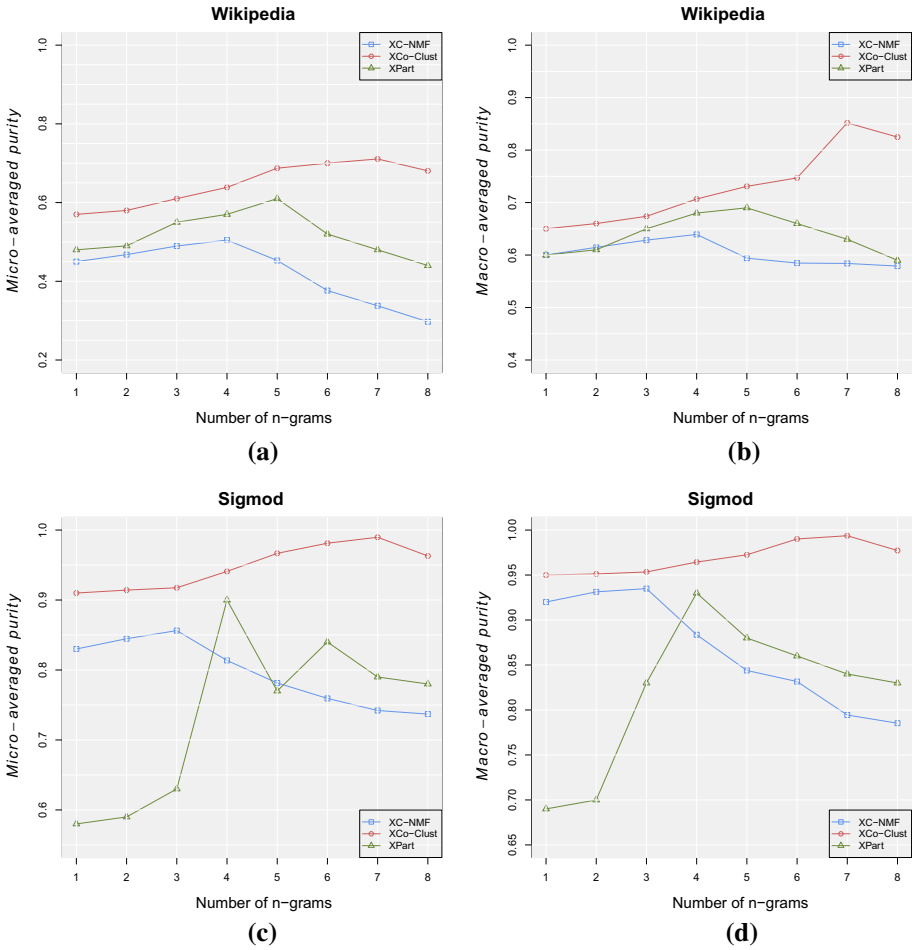


**Fig. 6** Clustering effectiveness with contextualized n-grams of length from 1 to 8. **a** Micro-averaged purity on *Wikipedia*, **b** macro-averaged purity on *Wikipedia*, **c** micro-averaged purity on *Sigmod*, **d** macro-averaged purity on *Sigmod*

## 6.6 Scalability

We now turn to investigate the scalability of XC-NMF, XCo-Clust and XPart. Figure 7 shows the runtime taken by XC-NMF, XCo-Clust and XPart to find clusters in increasingly larger samples of the original *Wikipedia* corpus, when the length of the targeted contextualized n-grams is fixed to 3. XC-NMF is the most scalable on *Wikipedia* among the tested approaches to XML clustering by contextualized n-grams. Interestingly, XCo-Clust exhibits a superior scalability on *Wikipedia* with respect to XPart, despite simultaneously clustering both the XML documents and their respective XML features.

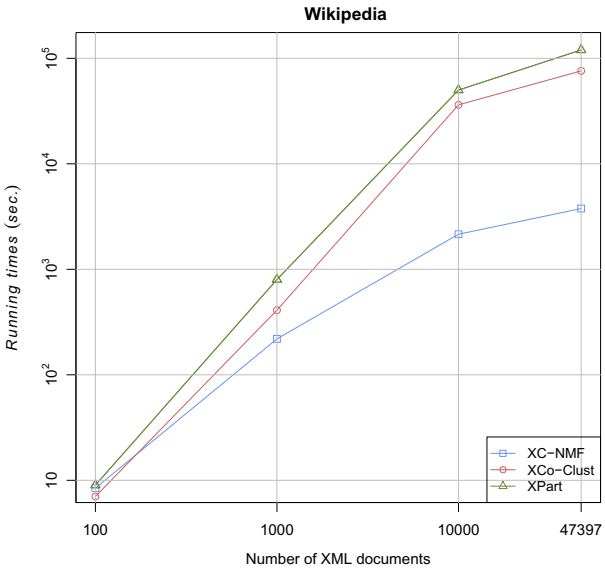


**Fig. 7** Scalability with contextualized n-grams of length 3

## 7 Related works

A selection of state-of-the-art approaches to XML clustering is reviewed hereunder. The review is by no means exhaustive and the interested reader is referred to Algergawy et al. (2011) for a more comprehensive survey.

Much of the efforts towards XML partitioning have focused only on the (sub)structures of the XML documents. Hierarchical clustering has been largely adopted (e.g. Costa et al. 2004, 2013; Dalamagas et al. 2006; Lee et al. 2002; Lian et al. 2004), because of the high quality of its results. Partitioning clustering techniques have also been investigated (e.g. Aggarwal et al. 2007; Costa and Ortale 2012b). However, one serious limitation of all structure-oriented approaches is that these cannot effectively divide XML documents with strongly matching or undifferentiated structures, despite meaningful differences in their textual contents.

A number of approaches have also been proposed to cluster XML documents by both content and structure. The approach in Tran et al. (2008) combines incremental clustering with clustering based on pairwise distance matrix, in order to effectively divide large XML corpora. The combination is meant to first reduce the dimension of the XML corpus and, then, group on the basis of pairwise distances to preserve the effectiveness of the resulting clustering. The incremental clustering progressively groups the available XML corpus into a number of clusters, by comparing each remaining XML document with the representative of the already discovered clusters. The representative of a cluster is simply the XML document that originated that cluster. At this point, a pairwise distance matrix is exploited to further reduce the number of unveiled clusters. More precisely, the similarity between the representatives of the incrementally discovered clusters is computed and the resulting pairwise distance matrix is fed into a graph clustering method. The latter merges the incrementally discovered clusters together, so that to ultimately yield a required number of clusters. The structural similarity between XML documents is measured by considering the common elements in the nested paths of XML documents. The semantic similarity between

the contents of the XML documents is computed by using a latent semantic kernel with a suitable subset of the XML corpus.

The approach proposed in Tran et al. (2008) clusters XML documents by separately computing the similarity of their content and structures (which are assumed to have different relevance based on the nature of the underlying XML corpus and pursued applicative purposes) and then combines both contributions through a weighted sum. A hierarchical clustering method is used to partition the XML documents based on their similarities. In particular, the semantic relationships between the contents of two XML documents are caught by means of a latent semantic kernel. Structural similarity is instead computed through the Euclidean distance between the occurrence frequencies within the two XML documents of the root-to-leaf paths observed in the whole XML corpus. A general criticism for those approaches, in which the relative importance of structure and content must be specified by the user, is that this explicitly involves a (potentially intensive) parameter tuning.

Frequent subtree mining is used in Kutty et al. (2009a, b) to represent the structural similarity of groups of XML documents with common substructures in terms of frequent subtrees. The content constrained by such subtrees within the individual XML documents is then extracted and represented in the vector space model (Salton et al. 1975), wherein the original XML corpus is then clustered through a bisection partitioning method. A critical aspect of both approaches concerns the enumeration of frequent subtrees, which is a time-expensive operation especially when performed over very large corpora of XML documents with complex structures. Additionally, when the available XML documents share overlapping structures, focusing on the content constrained by the frequent subtrees does not reduce text dimensionality, which is the main motivation for its adoption.

A two-stage hierarchical partitioning approach is presented in Doucet and Lehtonen (2006). At the first stage, the XML documents are divided with respect to their tags by a well known partitioning clustering algorithm, i.e., k-means. Only the most cohesive of the resulting clusters are retained at the second stage, whereas the remaining ones are further split with respect to their content again through the k-means algorithm. An inconvenient of this technique is that the user is required to provide a threshold for the structural homogeneity. Also, clustering XML documents by their tags at the first stage might prevent an effective grouping by their contents at the second stage.

The idea behind the study in Yao and Zerida (2007) is to project the XML documents into a space of root-based text path descriptors combined with filtered word descriptors and, then, use a constrained agglomerative clustering algorithm for partitioning.

The approach in Kutty (2011) is meant to overcome a drawback arising from the adoption of the vector space model (Salton et al. 1975) in the context of XML partitioning, i.e., the lost of the actual mapping between structure and content. In particular, a novel method of representing the XML documents based on the vector space model is exploited to preserve such a mapping. Furthermore, a randomized tensor decomposition technique is developed to expedite the analysis of large size tensors into memory. The XML trees are clustered by applying the k-means algorithm to the left singular matrix for the document order. A limitation of the tensor representation is that generally XML corpora with large and dense tensor representation cannot be directly analyzed through traditional decomposition algorithms, thus requiring the design of suitable techniques.

To the best of our knowledge, none of the existing approaches to XML clustering focuses on contextualized n-grams. The study in Costa and Ortale (2017) is a recent advance along this line of research. However, it is focused on the investigation of XPart (Costa and Ortale 2015b) alone with representations of the XML documents over both fixed- and mixed-length sequences of contextualized textual items.

This manuscript is the first research effort, in which XML clustering by structure-constrained n-grams is explored and instantiated from different and unexplored perspectives, i.e., XC-NMF, XCo-Clust and XPart. In such approaches, XML partitioning is performed by looking at the contextualized n-grams within the flattened representations of the XML documents, rather than by explicitly computing the pair-wise similarity of their tree-like structures and nested contents.

## 8 Conclusions and future research

We identified a novel family of approaches to XML clustering by structure and nested content. These can be understood as instantiations of a new method, proposed in this manuscript as an original cutting-edge research effort consisting in partitioning XML documents by structure-constrained phrases. The effectiveness of the devised method was studied over real-world benchmark XML corpora, by experimenting with three state-of-the-art machine-learning approaches along previously-unexplored research directions in the XML domain, i.e., non-negative matrix (tri-)factorization, co-clustering and automatic transactional clustering. These approaches were used to partition flattened representations of the XML documents over a set of (suitably filtered) XML features, which capture approximate phrases under the form of word n-grams in the context of root-to-leaf paths. The empirical evidence revealed that the effectiveness of the three approaches at partitioning the chosen XML corpora actually improves, as long as contextualized n-grams of appropriate length are used as XML features. This confirms the validity of XML clustering by structure-constrained phrases from different clustering perspectives. Furthermore, the exploitation of contextualized n-grams was shown to result into a superior effectiveness of both the co-clustering and transactional-clustering approaches with respect to several state-of-the-art competitors for XML clustering. Lastly, the relative scalability of the non-negative matrix (tri-)factorization, co-clustering and transactional-clustering approaches was studied on a large-scale corpus of text-centric XML documents.

Future research in the field of XML clustering by structure-constrained n-grams involves the development and comparative analysis of further clustering techniques. In particular, constrained agglomerative clustering and nearest-neighbor clustering are two appealing techniques. The former would allow the incorporation of the advantages of both the partitioning and hierarchical schemes (Zhao and Karypis 2005), while the latter would allow clustering very large corpora of XML documents through hash-based indexing (Costa et al. 2010). Moreover, it is interesting to study whether ensemble XML clustering (Costa and Ortale 2013a) can be exploited to improve the effectiveness of the input XML clusterings by structure-constrained phrases. Finally, the development of suitable topic models for XML corpora would enable partitioning at a high (thematic) level by latent-topics. Therein, a first step was recently proposed in Costa and Ortale (2015b). However, although empirically shown to be effective for XML clustering, the generative semantics of the XML topic model in Costa and Ortale (2015b) accounts for XML features, that still reflect the nesting of unigrams into root-to-leaf paths. Further improvements in the effectiveness of thematic XML clustering are expected by redefining the generative process of the XML topic model to account for the nesting of phrases into the logical structure of the XML documents.

## References


Abiteboul, S. (1997). Querying semistructured data. In *Proceedings of the international conference on database theory* (pp. 1–18).

Abiteboul, S., Buneman, P., & Suciu, D. (2000). *Data on the Web: From relations to semistructured data and XML*. Burlington: Morgan Kaufmann.

Aggarwal, C. C., Ta, N., Wang, J., Feng, J., & Zaki, M. (2007). Xproj: A framework for projected structural clustering of xml documents. In *Proceedings of the international conference on knowledge discovery and data mining* (pp. 46–55).

Albright, R., Cox, J., Duling amd, D., Langville, A. N., & Meyer, C. D. (2006). Algorithms, initializations, and convergence for the nonnegative matrix factorization. Technical Report Math 81706, North Carolina State University.

Algergawy, A., Mesiti, M., Nayak, R., & Saake, G. (2011). Xml data clustering: An overview. *ACM Computing Surveys*, *43*(4), 25:1–25:41.

Bratko, A., & Filipič, B. (2006). Exploiting structural information for semi-structured document categorization. *Information Processing and Management*, *42*(3), 679–694.

Cesario, E., Manco, G., & Ortale, R. (2007). Top-down parameter-free clustering of high-dimensional categorical data. *IEEE Transactions on Knowledge and Data Engineering*, *19*(12), 1607–1624.

Cho, H., Dhillon, I. S., Guan, Y., & Sra. S. (2004). Minimum sum-squared residue co-clustering of gene expression data. In *Proceedings of the SIAM international conference on data mining* (pp. 114–125).

Cichocki, A., Zdunek, R., Phan, A. H., & Amari, S. (2009). *Nonnegative matrix and tensor factorizations*. London: Wiley.

Connolly, T., & Begg, C. (2002). *Database systems: A practical approach to design, implementation, and management*. Reading: Addison Wesley.

Costa, G., Manco, G., & Ortale, R. (2008). A hierarchical model-based approach to co-clustering high-dimensional data. In *Proceedings of ACM symposium on applied computing* (pp. 886–890)

Costa, G., Manco, G., & Ortale, R. (2010). An incremental clustering scheme for data deduplication. *Data Mining and Knowledge Discovery*, *20*(1), 152–187.

Costa, G., Manco, G., Ortale, R., & Ritacco, E. (2013). Hierarchical clustering of xml documents focused on structural components. *Data and Knowledge Engineering*, *84*, 26–46.

Costa, G., Manco, G., Ortale, R., & Tagarelli, A. (2004). A tree-based approach to clustering xml documents by structure. In *Proceedings of the international conference on principles and practice of knowledge discovery in databases* (pp. 137–148).

Costa, G., & Ortale, R. (2012a). On effective xml clustering by path commonality: An efficient and scalable algorithm. In *IEEE international conference on tools with artificial intelligence* (pp. 389–396).

Costa, G., & Ortale, R. (2012b). Structure-oriented clustering of xml documents: A transactional approach. In *IEEE international conference on intelligent systems* (pp. 188–193).

Costa, G., & Ortale, R. (2013a). Developments in partitioning xml documents by content and structure based on combining multiple clusterings. In *IEEE international conference on tools with artificial intelligence* (pp. 477–482).

Costa, G., & Ortale, R. (2013b). A latent semantic approach to xml clustering by content and structure based on non-negative matrix factorization. In *IEEE international conference on machine learning applications* (pp. 179–184).

Costa, G., & Ortale, R. (2014). Xml document co-clustering via non-negative matrix tri-factorization. In *International conference on tools with artificial intelligence* (pp. 607–614).

Costa, G., & Ortale, R. (2015a). Fully-automatic xml clustering by structure-constrained phrases. In *International conference on tools with artificial intelligence* (pp. 146–153).

Costa, G., & Ortale, R. (2015b). Mining clusters in xml corpora based on Bayesian generative topic modeling. In *International conference on machine learning applications* (pp. 515–520).

Costa, G., & Ortale, R. (2017). XML clustering by structure-constrained phrases: A fully-automatic approach using contextualized n-grams. *International Journal on Artificial Intelligence Tools*, *26*(1), 1–24.

Costa, G., Ortale, R., & Ritacco, E. (2011). Effective xml classification using content and structural information via rule learning. In *International conference on tools with artificial intelligence* (pp. 102–109).

Costa, G., Ortale, R., & Ritacco, E. (2013). X-class: Associative classification of xml documents by structure. *ACM Transactions on Information Systems*, *31*(1), 3:1–3:40.

Dalamagas, T., Cheng, T., Winkel, K.-J., & Sellis, T. (2006). A methodology for clustering xml documents by structure. *Information Systems*, *31*(3), 187–228.

Denoyer, L., & Gallinari, P. (2007). Report on the xml mining track at INEX 2005 and INEX 2006. *ACM SIGIR Forum*, *41*(1), 79–90.

Denoyer, L., & Gallinari, P. (2008). Report on the xml mining track at INEX 2007. *ACM SIGIR Forum*, *42*(1), 22–28.

Dhillon, I. S. (2001). Co-clustering documents and words using bipartite spectral graph partitioning. In *Proceedings of ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 269–274).

Dhillon, I. S., Mallela, S., & Modha, D. S. (2003). Information-theoretic co-clustering. In *Proceedings of ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 89–98).

De Francesca, F., Gordano, G., Ortale, R., & Tagarelli, A. (2003). Distance-based clustering of xml documents. In *International ECML/PKDD workshop on mining graphs, trees and sequences* (pp. 75–78).

Ding, C. H. Q., Li, T., Peng, W., & Park. H. (2006). Orthogonal nonnegative matrix tri-factorizations for clustering. In *Proceedings of ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 126–135).

Doucet, A., & Lehtonen, M. (2006). Unsupervised classification of text-centric xml document collections. In *Proceedings of the workshop of the initiative for the evaluation of XML retrieval* (pp. 497–509).

Fox, C. (1992). *Lexical analysis and stoplists*. Upper Saddle River: Prentice Hall.

Hagenbuchner, M., Tsoi, A. C., Sperduti, A., & Kc, M. (2008). Efficient clustering of structured documents using graph self-organizing maps. In *Focused access to XML Documents* (pp. 207–221)

Joshi, S., Agrawal, N., Krishnapuram, R., & Negi. S. (2003). A bag of paths model for measuring structural similarity in web documents. In *ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 577–582).

Kutty, S., Nayak, R., & Li, Y. (2009a). Hcx: An efficient hybrid clustering approach for xml documents. In *Proceedings of ACM symposium on document engineering* (pp. 94–97).

Kutty, S., Nayak, R., & Li, Y. (2009b). Xcfs: A novel approach for clustering xml documents using both the structure and the content. In *Proceedings of ACM conference on information and knowledge management* (pp. 1729 – 1732).

Kutty, S., Nayak, R., & Li, Y. (2011). Xml documents clustering using a tensor space model. In *Pacific-Asia conference on advances in knowledge discovery and data mining* (pp. 488–499).

Lee, D. D., & Seung, H. S. (2001). Algorithms for non-negative matrix factorization. In *Advances in neural information processing systems* (pp. 556–562).

Lee, M. L., Yang, L. H., Hsu, W., & Yang, X. (2002). Xclust: Clustering xml schemas for effective integration. In *Proceedings of international conference on information and knowledge management* (pp. 292–299).

Li, T., Sindhwani, V., Ding, C. H. Q., & Zhang, Y. (2010). Bridging domains with words: Opinion analysis with matrix tri-factorizations. In *Proceedings of SIAM international conference on data mining* (pp. 293–302).

Lian, W., Cheung, D. W., Mamoulis, N., & Yiu, S.-M. (2004). An efficient and scalable algorithm for clustering xml documents by structure. *IEEE Transactions on Knowledge and Data Engineering*, *16*(1), 82–96.

Long, B., Wu, X., Zhang, Z., & Yu, P. S. (2006). Unsupervised learning on k-partite graphs. In *Proceedings of ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 317–326).

Menahem, E., Schclar, A., Rokach, L., & Elovici, Y. (2016). Xml-ad: Detecting anomalous patterns in xml documents. *Information Sciences*, *326*, 71–88.

Pauca, V. P., Shahnaz, F., Berry, M. W., & Plemmons, R. J. (2004). Text mining using non-negative matrix factorizations. In *SIAM international conference on data mining* (pp. 452–456).

Piernik, M., Brzezinski, D., Morzy, T., & Lesniewska, A. (2015). Xml clustering: A review of structural approaches. *The Knowledge Engineering Review*, *30*(3), 297–323.

Porter, M. F. (1980). An algorithm for suffix stripping. *Program*, *14*(3), 130–137.

Salton, G. (1991). Developments in automatic text retrieval. *Science*, *253*, 974–980.

Salton, G., Wong, A., & Yang, C. S. (1975). A vector space model for information retrieval. *Communications of the ACM*, *18*, 613–620.

Shan, H., & Banerjee, A. (2008). Bayesian co-clustering. In *Proceedings of international conference on data mining* (pp. 530–539).

Song, Y., Pan, S., Liu, S., Wei, F., Zhou, M. X., & Qian, W. (2010). Constrained co-clustering for textual documents. In *Proceedings of AAAI conference on artificial intelligence* (pp. 581–586).

Tang, B., Shepherd, M., Milios, E., & Heywood, M. I. (2005). Comparing and combining dimension reduction techniques for efficient text clustering. In *Canadian conference on artificial intelligence* (pp. 292–296).

Tran, T., Nayak, R., & Bruza, P. (2008). Combining structure and content similarities for xml document clustering. In *Australasian conference on data mining* (pp. 219–226).

Tran, T., Nayak, R., & Bruza, P. (2008). Document clustering using incremental and pairwise approaches. In *Focused access to XML documents* (pp. 222–233).

W3C. Extensible markup language (xml) 1.0 (fifth edition) W3C Recommendation. 2008. http://www.w3c.org.

Wang, H., Nie, F., Huang, H., & Makedon, F. (2011). Fast nonnegative matrix tri-factorization for large-scale data co-clustering. In *Proceedings of international joint conference on artificial intelligence* (pp. 1553–1558).

Wang, P., Domeniconi, C., & Laskey, K. B. (2009). Latent Dirichlet Bayesian co-clustering. In *Proceedings of European conference on machine learning and principles and practice of knowledge discovery in databases* (pp. 522–537).

Wilde, E., & Glushko, R. J. (2008). Xml fever. *Communications of the ACM*, *51*(7), 40–46.

Xu, W., Liu, X., & Gong, Y., (2003). Document clustering based on non-negative matrix factorization. In *ACM SIGIR conference on research and development in information retrieval* (pp. 267–273).

Yao, J., & Zerida, N. (2007). Rare patterns to improve path-based clustering of Wikipedia articles. In *Pre-proceedings of the initiative for the evaluation of XML retrieval* (pp. 224–231).

Zaki, M. J., & Aggarwal, C. C. (2003). Xrules: An effective structural classifier for xml data. In *Proceedings of SIGKDD international conference on knowledge discovery and data mining (KDD)* (pp. 316–325).

Zhao, Y., & Karypis, G. (2005). Hierarchical clustering algorithms for document datasets. *Data Mining and Knowledge Discovery*, *10*(2), 141–168.