Multilevel Conditional Fuzzy C-Means Clustering of XML Documents

Michal Kozielski

Silesian University of Technology, Akademicka 16, 44-100 Gliwice
michal.kozielski@polsl.pl

Abstract. XML documents are the special kind of data having hierarchical structure. Typical clustering algorithms do not meet requirements which may be stated for analysis of such data. A novel, dedicated for XML documents clustering method called *Multilevel clustering of XML documents* (*ML*) is presented in the paper. The method clusters feature vectors encoding XML documents on the different structure levels. Application of *Conditional Fuzzy C-Means* algorithm to *ML* method is proposed in the paper and the advantage of this fuzzy method over hard approach to *ML* algorithm is discussed and proved. An application of *ML* method to accelerating query execution on XML documents is discussed in the paper. The experimental results performed on two data sets having different characteristics show that the proposed method of multilevel conditional fuzzy clustering of XML documents outperforms hard multilevel clustering.

Keywords: clustering, clustering XML documents.

1 Introduction

Popularity of XML standard (*eXtensible Markup Language*) and a large number of its applications triggered an intense development of the new database systems called *native XML databases*. Also the functionality of relational systems supporting XML storage is continuously extended in order to effectively store, query and process XML documents.

Due to flexibility of the XML document structure it is possible that some queries will address only few documents in a large database. In order to accelerate execution of such selective queries on XML documents it is possible to consider a method reducing number of documents which have to be analysed. The reduced document set must contain all the documents which are addressed by a query and shall contain as little number of other documents as possible. It is possible to apply clustering of XML documents according to their structure in order to determine such groups of documents.

There are plenty of clustering algorithms which may be applied to the task presented above [4]. These algorithms however, are not dedicated to the hierarchical structure of XML documents and do not perform in acceptable way concerning the specified task. A new approach called *Multilevel clustering of XML documents* (*ML* algorithm) was proposed in [5]. *ML* algorithm is dedicated to a hierarchical structure

J.N. Kok et al. (Eds.): PKDD 2007, LNAI 4702, pp. 532–539, 2007.

of XML documents and may take advantage of any clustering algorithm. The paper presented defines formally *ML* algorithm and describes the shortcomings of using hard clustering in *ML* approach and proposes the solution of the problem by means of *Conditional Fuzzy C-Means* algorithm (CFCM) [11].

The paper is organised as follows. Section 2 describes the approach to accelerating XML queries by clustering XML documents. Section 3 presents *Multilevel clustering of XML documents* algorithm and application of *CFCM* to *ML* algorithm. Datasets which were used in the analysis and the results of the experiments which were performed are presented in section 4. The final conclusions are drawn in section 5.

2 Accelerating XML Query Execution

Elements and attributes of XML documents which are addressed in the queries are defined by means of path expressions. Flexibility of the structure of XML documents stored in a database causes that occurrence of an element or attribute may be optional and not all the documents in a collection match a path specified in a query. Assuming that an execution of a query on a subset of documents is less time consuming then querying the whole collection it is worth verifying the methods which could determine the collection subsets addressing the given queries.

Occurrence of an element or an attribute is a feature of a structure of XML document. It is possible therefore, to apply methods of clustering XML documents according to their structure to determine such document subsets. Having a cluster of documents it should be possible to calculate a signature of the cluster representing all the features (elements and attributes) existing in the cluster. It should be also possible to calculate a signature of a query representing all the features which are addressed by the query. Comparison of the two signatures (of a cluster and a query) should show whether the query addresses any documents in the cluster and whether the XML documents in a cluster should be processed by the query.

3 Clustering XML Documents

In order to determine the clusters of XML documents having similar structure within the clusters and different structure between the clusters it is needed to:

- apply one of the methods calculating similarity or distance between the XML document structures,
- apply one of the clustering algorithms determining the clusters.

In the presented work an approach calculating structural similarity or distance on the bases of feature vectors encoding the structure features of the documents was used. The analysis of two encoding methods: signal encoding [3] and bit encoding [7], [13] was performed [5] and bit encoding was chosen as the only acceptable approach. An assumption was taken in a work presented that the query paths are fully defined and indicate all the elements starting from a root element up to a target node.

3.1 Clustering Algorithms Review

There is a large number of clustering algorithms [4] which can be applied to the task of clustering bit feature vectors and supporting proposed method of accelerating XML queries execution. Algorithms of different types like hierarchical algorithms, e.g. *Complete* or *Single Link* [4] or partitional algorithms, e.g. *Hard C-Means* [4] can be used in the presented task. However, these algorithms are not dedicated to the data representing a structure of XML documents.

The algorithms mentioned above perform clustering in a full feature vector space. Bit encoding produces very long feature vectors what decreases a clustering quality [6], [8]. Additionally, the queries which may be performed on XML documents do not traverse through the whole tree structure to the leafs very often. Concerning the application presented reduction of the number of features cannot be performed by any of the known methods [6], [8] because they operate on the whole feature space and they do not differentiate features according to the document structure level.

It is also a common observation that the most general and therefore important information is enclosed nearby the root element concerning the structure of XML document. The features which are placed on the levels neighbouring a root element should have therefore, a greater influence on the clustering results then the leaf nodes what cannot be achieved by means of the algorithms mentioned above.

There are approaches to clustering XML documents concerning their structure which take under consideration tree-like structure of XML documents and the significance of the features depending on their level in this structure [3], [10]. These algorithms however, introduce methods dedicated to XML structure on a level of calculating similarity between document structures. They do not operate on bit feature vectors encoding XML document structure which were shown to be very effective in the presented method of accelerating XML queries.

There was therefore, a need to introduce a new clustering algorithm which would be dedicated to XML documents and which would address all the requirements which are not met by the clustering algorithms mentioned above. The new clustering algorithm giving promising results was called *Multilevel Clustering of XML Documents* (*ML*) [5].

3.2 Multilevel Clustering of XML Document Structure

Multilevel Clustering of XML Documents (*ML*) [5] is a method dedicated to XML documents. Multilevel approach starts clustering at a root level and continues the process at the following levels. In this way it differentiates features treating the elements placed in the neighbourhood of a root element as more significant. It is possible to stop the algorithm at a certain level of the document structure tree reducing a number of features which are processed.

Defining a set of XML documents as $D = \{ d_1, \dots , d_N \}$ and a feature vector B encoding each document as a string of bits as $B = \{ b_1^1, \dots , b_{n1}^1 , \dots , b_1^l, \dots , b_{nl}^l \}$ where $l = 1, \dots , L_T$ is a number of a level at which occurs a given feature. A hard clustering result on a level l is a partition of a set D to a set of clusters $C^l = \{ C_1^l, \dots , C_{Kl}^l \}$, where K_l is a number of clusters determined on a level l, $\bigcup_{i=1}^{K_l} C_i^l = D$, and $C_i^l \cap$

$C_j^l = \Phi$, where $i \neq j$. Each cluster C_i^l may be partitioned on a level $l+1$ giving a set of new clusters. A final clustering result is a set of clusters $C = C^L$, where L is a level of XML structure tree which is defined by a user as a stop condition. The other input parameters are a user defined final number of clusters K_L and a distribution of the features among the document structure levels. Clustering on each level can be performed by means of any clustering algorithm. The definitions presented above concern the process of hard clustering which was used in the previous analysis [5] and which may be illustrated by the figure presented below.

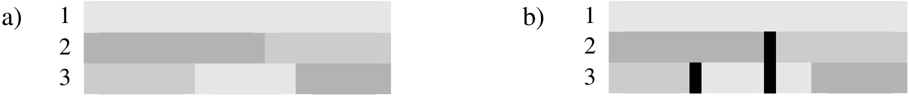

Fig. 1. Illustration of a feature space of XML documents (a) and the partition created by hard Multilevel clustering of XML documents (b)

Figure 1. illustrates a feature space of a set of XML documents (Fig. 1. a)) captured at the three levels of XML document structures. Documents are placed on axis X, structure levels are placed on axis Y and the existing clusters relevant to different feature values are depicted by different shades. The result of applying hard Multilevel clustering of XML documents (Fig. 1. b)) is marked by thick black lines separating the clusters. The final number of clusters was set to three. Figure 1. illustrates also a problem which may be encountered when applying hard clustering (e.g. *HCM* algorithm) to *ML* method. The hard partitioning on the second level of the document trees is directly transferred to the next iteration and determines the clusters on the third level making one of them inconsistent. In order to solve the presented problem it is needed to introduce an algorithm affecting a clustering on a level $l+1$ by the results of a clustering performed on a level l but not determining them in a hard way. Therefore, *Conditional Fuzzy C-Means* [11] algorithm is proposed and applied into *ML* method in the presented work.

Conditional Fuzzy C-Means algorithm was introduced in [11] and generalised in [9]. It extends *Fuzzy C-Means* [9], [11] algorithm by introducing conditional variable f_k. Conditional variable f_k specifies what is the impact of a data object x_k on the created partition. The result of an algorithm is a pair of iteratively modified matrices: a fuzzy partition matrix $U = [u_{ik}]$ defining what is the membership value of each data object x_k to each cluster C_i and a prototype matrix $V = [v_i]$.

3.3 Multilevel Conditional Fuzzy C-Means

ML algorithm where *CFCM* algorithm is applied to perform partitioning on each level of XML document structure trees was named *Multilevel Conditional Fuzzy C-Means* (*MLCFCM*). The clustering results at the level l defined in paragraph 3.2 for hard clustering must be modified and defined as a fuzzy partition having a form of a matrix $U^l = [u_{ik}^l]$ of size $K_l \times N$, where K_l was defined as a number of clusters determined on a level l, N is a number of all documents which are analysed. The created clusters are not separated and partitioned in the consecutive iterations as it was performed when hard clustering was used. The partition matrix U^{l-1} calculated on a previous level $l-1$

of the document structure trees becomes a bases of condition matrix $F_{Kl}{}^{l}$ containing the values of condition parameter f_k used in *CFCM* algorithm. Condition parameters impact a new fuzzy partition U^l on a level l. Condition matrix $F_{Kl}{}^{l} = g(U^{l-1})$ may be calculated by means of different forms of function g. A final partition U^L is binarized what gives a hard partition defining a set of clusters C^L as defined in paragraph 3.2.

The approach presented above ensures that clustering on a level closer to a root element will impact the partition of the features placed further in the XML document tree. At the same time, a direct transfer of cluster borders, which is performed when hard clustering is used, is avoided what enables the algorithm to produce a correct results for the case presented on figure 1.

Another advantage of this method is a possibility of detection of the documents having a strongly different structure comparing to the cluster prototypes. Concerning acceleration of the XML queries it is profitable to receive as compact clusters as possible. The characteristic feature of the documents of this type is a very small variance of the membership values in a partition matrix U^l. The proposed *MLCFCM* algorithm assigns documents strongly distinct from the cluster prototypes on each level to the "others" cluster.

4 Experiments

Performance of *ML* algorithm which was presented in the previous sections was analysed on two sets of XML documents.

Level3 dataset [2] consists of 112 XML documents which were generated by means of ToXgene tool [1]. This dataset was used in order to verify the differences between hard version of *ML* algorithm and *MLCFCM* method. Therefore, the characteristics of the generated documents refer to the data illustrated on figure 1. Additionally, twelve documents having distinct structure from the others were added to the dataset what should enable to verify how fuzzy approach can deal with the data of very different characteristic. Encoding the documents creating *Level3* dataset gave a feature vector containing 48 bits.

Wiki dataset consists of 989 XML documents randomly chosen from a large extract of Wikipedia to XML format [12]. This dataset is a real life dataset of unknown structure and it is used in order to verify how a new *MLCFCM* algorithm performs in general conditions. Encoding the documents from *Wiki* dataset gave a feature vector containing 6557 bits distributed among 36 levels of XML document structure trees.

4.1 Partition Consistency Verification

Section 3.2 presented a problem which may occur when hard clustering is implemented in multilevel method and which may lead to inconsistent clusters. In order to confirm the expected advantage (described in section 3.3) of *MLCFCM* algorithm over hard multilevel approach, *Level3* dataset was clustered by both kinds of algorithms. *Hard C-Means* algorithm was implemented in *ML* method as a hard multilevel algorithm (*MLHCM*). The figures presented below show the structure of *Level3* dataset (fig. 2.) and the partitions created by both algorithms (fig. 3.).

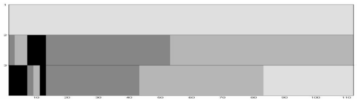

Fig. 2. Designed structure of *Level3* dataset

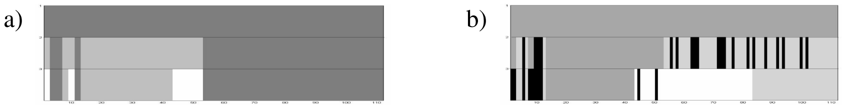

Fig. 3. Partition created by *MLHCM* algorithm (a) and by *MLCFCM* algorithm (b)

Each figure presents the results of clustering of 112 XML documents placed on *X* axis on three levels of XML structure placed on *Y* axis. Different clusters are marked by different shades. The colour of a cluster is not important (except black colour) and was used in order to distinguish the clusters. Documents having a strongly different structure and assigned to "others" cluster were marked in black. Figure 2. presents the expected result of clustering according to designed structure of XML documents.

In this experiment the following function *g* transforming partition matrix U^{l-1} to condition matrix F_{Kl}^{l} was used in *MLCFCM* algorithm:

$$g = 0.5 \cdot U^{l-1} + 0.5 \qquad (1)$$

The figures presented above show that only *MLCFCM* algorithm revealed the clusters existing on the third level of document structures correctly. Hard multilevel clustering assigned documents numbered from 53 to 112 into one inconsistent cluster.

4.2 Dataset Reduction Verification

Another experiment shows the results of accelerating XML query execution by means of *ML* method. As it was presented in the section 2, it is assumed that a query should be faster executed on a reduced set of documents containing all the documents addressed by the query. Therefore, a reduction degree of the datasets which was received by means of clustering algorithm was compared in the experiment.

In case of *Level3* dataset an average degree of reduction was calculated for all possible query paths. Comparison of fuzzy (*MLCFCM*) and hard (*MLHCM*) implementation of *ML* method was performed and the results of the analysis are presented in table 1.

Table 1. Average reduction degree of *Level3* dataset

MLHCM	MLCFCM (*v=0*)	MLCFCM (*v=0.01*)
59.3	62.9	72.4

Two values of parameter *v* determining the maximal value of variance of partition matrix which assigns a document as "others" were used in the implementation of

MLCFCM algorithm. A value of *v=0* means that no document was assigned to “others” cluster. Function *g* was defined in the same way as in equation (1).

Presented in table 1. average numbers of documents which were reduced as not being addressed by a query show that *MLCFCM* algorithm performs better concerning the presented application to acceleration of queries on XML documents. The difference between the reduction values for *MLHCM* and *MLCFCM* where *v=0* is not very large but the analysis of the particular paths addressing the documents which were incorrectly clustered by *MLHCM* algorithm (documents numbered from 53 to 83 on fig. 3.) show that the difference in reduction degree for these paths may be significant (71 reduced documents in case of *MLCFCM* instead of 30 documents reduced in case of *MLHCM*). It is also visible that the documents having strongly different structure may decrease the quality of clustering. It is important therefore, to assign that kind of documents to a separate cluster what may be performed by means of *MLCFCM* algorithm.

In case of *Wiki* dataset a reduction degree for four queries was compared. Table 2. presents what is a number of documents which are addressed and which are not addressed by the queries. Table 3. presents the results of the analysis performed on different levels of the document structure trees. In this experiment, a value of parameter *v* was set to *v = 10^{-13}* and *g* function was defined as $g = U^{l-1}$.

Table 2. Characteristics of the queries defined on *Wiki* dataset

Query	Query path	Path length (no. of levels)	Documents addressed by a query	Documents which may be reduced
q1	/article/body/section/title	4	620	369
q2	/article/body/definitionlist	3	3	986
q3	/article/body/normallist	3	56	933
q4	/article/body/figure	3	155	834

Table 3. Number of documents reduced by means of *ML* method

Query	Level 3		Level 4	
	MLHCM	MLCFCM	MLHCM	MLCFCM
q1	355	0	319	0
q2	347	553	211	508
q3	14	93	0	0
q4	0	13	0	127

The results presented in table 3. show that *MLCFCM* algorithm performed better concerning queries *q2*, *q3* and *q4*, where the difference received for query *q2* is significant. *MLHCM* algorithm however, performed significantly better concerning query *q1*.

5 Conclusions

The new method of multilevel clustering of XML documents (*ML*) is presented in the paper. The discussion presented in the paper and the results of the experiments which were performed show that application of conditional fuzzy clustering to *ML* method (*MLCFCM*) is able to produce more consistent clusters comparing to hard clustering algorithm (*MLHCM*).

An application of clustering XML documents according to their structure to accelerating query execution on XML document set was also presented in the paper. An analysis of the methods encoding a structure of XML document which could be applied to this task showed that only bit encoding meets the requirements which were stated. The experiments comparing fuzzy (*MLCFCM*) and hard (*MLHCM*) implementations of *ML* method show that multilevel conditional fuzzy c-means clustering gives better results then *HCM* implementation on both generated and real life data.

References

1. Barbosa, D., Keenleyside, J., Lyons, K., Mendelzon, A.: ToXgene - the ToX XML Data Generator (2007), http://www.cs.toronto.edu/tox/toxgene
2. Dataset used in the experiments: http://dydaktyka.polsl.pl/ZTiPSK/ IndywidualnePlanyZajec/ Michal_Kozielski_dane/mkoz_www.html
3. Flesca, S., et al.: Fast Detection of XML Structural Similarity. IEEE Transactions on Knowledge and Data Engineering 17(2) (2004)
4. Jain, A.K., Murty, M.N., Flynn, P.J.: Data Clustering: A review. ACM Computing Surveys 31(3) (1999)
5. Kozielski, M.: Przyspieszanie realizacji zapytań na dokumentach XML z wykorzystaniem grupowania względem ich struktury, Bazy Danych, Nowe Technologie: Architektura, metody formalne i zaawansowana analiza danych, WKŁ, pp. 305–314 (2007)
6. Kozielski, M.: Improving the Results and Performance of Clustering Bit-encoded XML Documents. In: Proc. of ICDM Workshops 2006, pp. 60–64. IEEE Computer Society Press, Los Alamitos (2006)
7. Lian, W., et al.: An Efficient and Scalable Algorithm for Clustering XML Documents by Structure. IEEE Transactions on Knowledge and Data Engineering 16(1) (2004)
8. Liu, J., et al.: XML Clustering by Principal Component Analysis. In: Proceedings of the 16th IEEE International Conference on Tools with Artificial Intelligence (ICTAI 2004), IEEE Computer Society Press, Los Alamitos (2004)
9. Łęski, J.: Generalized Weighted Conditional Fuzzy Clustering. IEEE Transactions on Fuzzy Systems 11(6) (2003)
10. Nayak, R.: Fast and Effective Clustering of XML Data Utilizing their Structural Information, Under publication in KAIS: Knowledge and Information Systems - An International Journal
11. Pedrycz, W.: Conditional Fuzzy C-Means. Pattern Recognition Letters 17, 625–631 (1996)
12. (2006), http://xmlmining.lip6.fr
13. Yoon, J.P., Raghavan, V., Chakilam, V.: Bitmap Indexing-based Clustering and Retrieval of XML Documents. In: Proceedings of ACM SIGIR Workshop on Mathematical/Formal Methods in Information Retrieval, New Orleans, LA, ACM Press, New York (2001)