Minimization of the disagreements in clustering aggregation

Safia Nait Bahloul[1] Baroudi Rouba[1] Youssef Amghar[2]

[1]Computer department, Faculty of Science,
Es-Sénia , Oran University, Algeria
[2]INSA de Lyon – LIRIS FRE 2672 CNRS,
7 avenue Jean Capelle.
69621, Villeurbanne – France.
nait1@yahoo.fr

Abstract: Several experiences proved the impact of the choice of the parts of documents selected on the result of the classification and consequently on the number of requests which can answer these clusters. The process of aggregation gives a very natural method of data classification and considers then m produced classifications by them m attributes and tries to produce a classification called "optimal" which is the most close possible of m classifications. The optimization consists in minimizing the number of pairs of objects (u, v) such as a C classification place them in the same cluster whereas another C' classification place them in different clusters. This number corresponds to the concept of disagreements. We propose an approach which exploits the various elements of an XML document participating in various views to give different classifications. These classifications are then aggregated in the only one classification minimizing the number of disagreements. Our approach is divided into two steps: the first consists in applying the K-means algorithm on the collection of XML documents by considering every time a different element from the document. Second step aggregates the various classifications obtained previously to produce the one that minimizes the number of disagreements.

Keywords*: XML, classification, aggregation, disagreements.*

1 Introduction

The number of XML documents exchanged on internet increases continuously, and the necessary tools for the search for the information in documents are not sufficient enough. The tools allowing to synthesise or to classify wide collection of documents became indispensable.

The unsupervised automatic classification (or clustering) aims to regroup the similar documents. The search for a relevant information in a wide collection means then interrogating sets (classes) of reduced size. This bases itself on the idea that if a document is relevant in a request, their neighborhoods (the similar documents of the same class) have more chance to be also relevant.

Several experiences of XML documents classification of homogeneous structure were realized by [1]. These experiences showed the impact of the choice of the selected parts of documents on the result of the classification and consequently on the number of requests satisfied by these clusters. So the aggregation of these classifications allows obtaining more relevant clusters.
In this case we propose an approach allowing to optimize the aggregated clusters by minimizing the number of disagreements coming from a process of classification based on a set of attributes considered relevant.

2 The Classification of XML Documents

The classification consists in analyzing data and in affecting them according to their characteristics or attributes, to such or such class. There is an important quantity of methods of document classification. These methods can be classified generally, according to their objectives in two types: the supervised classification (classification) and the unsupervised classification (clustering).

The various presentations [2] of the methods of clustering are due, on one hand, to the fact that the classes of algorithms become covered (certain methods bases, for example, on probability models to propose partitions) and on the other hand, to the interest of the results of the clustering (hierarchy vs. Partitions, hard clustering vs. fuzzy Clustering etc.), and to the method used to reach this result (the use of the probability functions versus use of graphs, etc. …).

Several works concerning the clustering [3, 4, 5], [6, 7] and the similarity [8, 9, 10] of XML documents were realized, and this with different objectives. Some works aim to identify the part of the DTD the most use [11], the others try to identify frequent structures in a wide collection [12]. Among objectives, one finds also the need of identification of the DTD for heterogeneous collections [13], and finally to realize the clustering [14] the combination of the structure and the content of documents is taken into consideration. Certain methods of classification reduce XML documents to their purely textual part [4, 15], without taking advantage of the structure which c carries rich information.

The interest in [1] concerns the impact of the choice of the selected document's parts on the result of the classification. Two levels of selection were applied: one using the structure of the document, another at the level of the text first selected called a linguistic selection. A classification algorithm of type k-means [16, 17] builds a partition of documents, affects documents to classes and shows the list of the words which allowed the classification. So it has been proved that the quality of the classification depends strongly on selected parts of documents.

Several approaches use the concept of aggregation in classification in various domains such as: machine learning [18, 19], pattern recognition [20], bioinformatics [21], and data mining [22, 23]. The aggregation supplies a very natural method for the data classification.
By considering a set of tuples T1, …, Tn characterized by a set of attributes A1 , … .Am, our idea consists in seeing every attribute as an element being able to produce a simple classification of the data set; if attribute Aj contains Kj different values then Aj regroups data in Kj clusters. The aggregation process considers then produced m classifications by them m attributes and tries to produce a classification called "optimal" which is the most possible close to m classifications, that means minimizing the number of pairs of objects (u, v) such as one C places them in the same cluster, whereas another classification C_0 place them in different clusters. This number corresponds to the concept of disagreements [24]. Our approach consists then in aggregating a set of based classifications each one a relevant attribute extracted from the DTD of documents to be classified. Every classification is arisen from the application of the k-means algorithm [16, 17]. The quality of the obtained clusters is assured on one hand by the efficiency of the k-means algorithm, as reference algorithms of classification, and on the other hand by the optimization (minimization of disagreements) assured by the aggregation concept.

The following sections describe in detail steps and concepts of our approach.

3 Description of our Approach

The proposed approach follows four steps:

Step 1: Determination of the relevant elements set (the relevance of the element is determined by its frequency of appearance in requests)
Step 2: inventory for every attribute of the representative words (the attribute's possible values)
Step 3: Application of the k-means algorithm for every attribute extracted in the first step
Step 4: Aggregation of obtained results in the step 3.

3.1 Illustrative example

We illustrate our approach through an example. Let a collection of XML documents based on the following DTD:

<! ELEMENT Film (Title, Kind, Director, Actor, Actress, Year, Budget, editor) >

First stage in our approach is to identify the most important parts of the DTD being able to produce relevant clusters. It is evident that Title and budget do not constitute elements of classification. On the other hand, films can be to regroup in classes according to their kind, their director, their actors or their editor.

The following step in the process is the choice of the representative words of every attribute. The result of this step can have the shape of the following table:

Table 1. Example of representative words of attributes

Attribute	Representative words
Kind	Comedy, action, horror…
Actor	Tom Cruz, Kevin kosner…
Realisator	Spielberg, newman…
Editor	20 Century, 3 stars…

- Third step consists in applying K-means algorithm [16, 17] on the collection by considering every time a different attribute. One will have for example clusters "Comedy", "Horror", "Action" for the attribute "Kind"
- The last step is the phase of aggregation which allows aggregating the obtained classifications during the third step (first part). This step allows to build clusters of type " All the films of Action realized by Spielberg and edited by 20Century in which played Tom Cruz "

3.2 Definitions

Certain authors [24] define the aggregation as a problem of optimization aiming to minimize the number of disagreement among the m classification.

3.2.1 Aggregation

Let CL = {C1 , …... Cm} a set of m classifications. The concept of aggregation consists in producing a C classification which realizes a compromise with the m classifications.

3.2.2 Disagreement

Let C and C0 two classifications, the disagreement is defined as being a pair of objects (u, v) such as C place them in the same cluster, whereas C0 places them in different clusters. If d (C0, C) is the number of disagreements between C and C0, the aggregation will consist then in finding a C classification which minimizes:

$$\sum_{i=1}^{m} d\left(C_i, C\right) \qquad \textbf{(1)}$$

The equation (1) allows calculating the distance between a classification C and the set of classifications. This distance represents in fact the number of couple of (Vi, Vj) objects on which the two classifications are in discord.

Example of aggregation [24]

Let C1, C2 and C3 of classifications, V1, … , V6 are objects to be classified. The Value K in entry (Vi, Cj) expresses that the Vi object belongs to the Cj cluster. The C column corresponds to the optimal classification which minimizes the number of disagreements among the C1, C2, C3 classifications.

Table 2. Example of an optimal classification

	C_1	C_2	C_3	**C**
V_1	1	1	1	**1**
V_2	1	2	2	**2**
V_3	2	1	1	**1**
V_4	2	2	2	**2**
V_5	3	3	3	**3**
V_6	3	4	3	**3**

In this example the total number of disagreement is 5: one with the C2 classification for the couple (v5; v6), and four with the C1classification for the couples (v1; v2); (v1; v3); (v2; v4); (v3; v4). It is not difficult to determine the classification which minimizes the number of disagreements corresponding in this example to the C3 classification.

The determination of the classification can be defined as a problem of optimization aiming to minimize the number of disagreements.

We realize our approach by the **Clust-Agregat** algorithm which we describe in the following section.

4 Clust-Agregat Algorithm

In what follows, we present an algorithm which summarizes various steps of our approach. Algorithm accepts as entry a set of V objects. Each object is characterized by a set of A attributes. The Algorithm builds a C set of classifications by taking into account every time a different attribute.

General Process of Clust-Agregat:

- In entry, we have V, a set of objects to be classified;
- Let A= {A1,….Am}, a set of attributes such as:
  - For every attribute Ai:
    - To apply the algorithm of k-means;
    - To add the classification obtained Ci in the set of classifications;
- Application of the function 'to aggregate' on the set of the obtained classifications;
- In exit, we shall have a set of clusters forming the optimal classification

```
Algorithm: Clust-Agregat
Entry: V {the set of objects to be classified}
Exit: Cf {A set of clusters the optimal classification"}
A= {A1,….Am,} {A set f attributes}
Begin
C: =∅;{the set of classifications to be optimized}
For i from 1 to m do
Ci:=K-means/A;{Apply K-means by considering the attribute A}
 C:=C∪Ci;
End For
  Cf: =Aggregate(C,V);
End
   Function Aggregate(C, V) {return one Classification
u,v : two objects to V.}
Begin
 For i from 1 to m-1 Do
   For j from i+1 to m Do
   Dv(Ci,Cj):=0;
     For each (u,v) ∈V²
```

$$d_{u,v}(C_i,C_j)=\begin{cases}1 & \text{if } C_i(u)=C_i(v) \text{ et } C_j(u)\neq C_j(v)\\ & or\ \ C_i(u)\neq C_i(v) \text{ et } C_j(u)=C_j(v)\\ 0 & \text{else}\end{cases}$$

End for

$d_v(C_i,C_j) := d_v(C_i,C_j) + d_{u,v}(C_i,C_j)$; {distance between the two classifications C_i,C_j}

```
End For
End For
```

$$D(C)=\sum_{i=1}^{m} d_V(C_i,C)$$

```
Cf:= Min (D(C));
Return (Cf):
End
```

The function Aggregate returns an optimal classification. The optimum criterion of the result corresponds to the minimization of the number of disagreements; on the other term this function returns the classification which is in agreement with all the classifications of the C set.

5 Complexity of Clust-Agregat algorithm

Global complexity of Clust-Agregat algorithm: The complexity of Clust-Agregat depends in one hand both of the complexity of k-means algorithm, and that of the function Aggregate. The complexity of a variant of k-means algorithm (k-mediode fuzzy) has been evaluated to O (n2) [25]. One the other hand, the process of aggregation is in nature NP-complete [24], but after it has been demonstrated that is easy to find 2-approximation and consequently reducing the complexity of aggregation process to O(mn) (m is the number of classifications to be aggregated) (see the details of the BESTCLUSTERING algorithm in [24]). In general, we can say that the complexity of the proposed Clust-Agregat algorithm can exceed O (n^2).

Practical Aspects: The implanting of the algorithm Clust-Agregat is in progress makes on database Iris [26]. The purpose of this practice is the comparison of results with existing algorithms to prove efficiency and advantageous difference of our algorithm with regard to K-means. In the same frame we envisage a study aiming to aggregate algorithms of classification such as k-means and POBOC [25].

6 Conclusion

In this paper we exploited the fact that various elements of a XML document participate in various view and lead to different classifications. This method produces clusters which constitute partial views in the data set. We proposed an algorithm aiming to improve the quality of the obtained clusters by exploiting the notion of aggregation. Our approach is based on a optimization process minimizing the disagreement among obtained classifications by the application of the k-means algorithm. The quality of the obtained clusters is guaranteed in one hand by the optimization process and one other hand by the reference of the k-means algorithm.

7 References

1. Despeyroux, T., Lechavellier, Y., Trousse, B., Vercoustre, A. : Expériences de classification d'une collection de documents XML de structure homogène. Washington, DC, USA, 2002. IEEE Computer Society, (2004)
2. Mitchell, T.M.: Machine Learning. New York, McGraw Hill, (1997)
3. Guillaume, D., Murtagh, F.: Clustering of XML documents. Computer Physics Communications, 127(2-3):215–227, (2000)
4. Denoyer L., Vittaut J.-N., Allinari P., Brunessaux S.: Structured Multimedia Document Classification. In DocEng'03, Grenoble, France, pp 153-160, (2003)
5. Despeyroux, T., Lechavellier, Y., Trousse, B., Vercoustre, A.: Experiments in clustering homogeneous xml documents to validate an existing typology. In Proceedings of the 5th International Conference on Knowledge Management (I-Know), Vienna, Autriche, July (2005)

6. Lee, M.L., Liang Huai Yang, L.H., Wynne Hsu, W., Yang, X.: XClust: Clustering XML schemas for effective integration. In CIKM '02: Proceedings of the eleventh international conference on Information and knowledge, (2002)
7. M. Steinbach, M., Karypis, G., Kumar, V.: A comparison of document clustering techniques. In KDD Workshop on Text Mining, (2000)
8. Bertino, E., Guerrini, G., Mesiti, M.: Measuring the Structural Similarity among XML Documents and DTDs. Technical report, DISI-TR-02-02, (2001)
9. Flesca, S., Manco, G., Masciari, E., Pontieri, L., Pugliese, A.: Detecting Structural Similarities between XML Documents. In WebDB, pages 55–60, (2002)
10. Nierman, A., Jagadish, H.V.: Evaluating Structural Similarity in XML Documents. In Proceedings of the Fifth International Workshop on the Web and Databases (WebDB 2002), Madison, Wisconsin, USA, June (2002)
11. Lian, W., Cheung DW-L.: An Efficient and Scalable Algorithm for Clustering XML Documents by Structure. IEEE Trans. Knowl. Data Eng., 16(1):82–96, (2004)
12. Termier, A., Rousset, M.C, Sebag, M.: TreeFinder: A First Step towards XML Data Mining. In ICDM '02: Proceedings of the 2002 IEEE International Conference on Data Mining (ICDM'02), page 450, (2004)
13. J.McQueen: Some methods for classification and analysis of multivariate observations. In the Fifth Berkeley Symposium on Mathematical Statistics and Probability, volume 1, pages 281–297, (1967)
14. Doucet, A., Ahonen-Myka, H.: Naive Clustering of a large XML Document Collection. In INEX Workshop, pages 81–87, (2002)
15. Yi J., Sundaresan N.: A classifier for semi-structured documents. In Proc. of the 6th International Conference on Knowledge Discovery and Data mining, pp340-344, (2000)
16. Duda, R.O., Hart, P.E.: Pattern Classification and Scene Analysis. John Wiley & Sons, (1973)
17. Fukunaga, K.: Introduction to Statistical Pattern Recognition. Academic Press, San Diego, CA, (1990)
18. Strehl, A., Ghosh, J.: Cluster ensembles: A knowledge reuse framework for combining multiple partitions. Journal of Machine Learning Research, (2002)
19. Fern, X.Z., Brodley, C.E.: Random projection for high dimensional data clustering: A cluster ensemble approach. In ICML, (2003)
20. Fred, A.L.N., Jain, A.K.: Data Clustering using evidence accumulation. In ICPR, (2002).
21. Filkov, V., Skeina, S.: Integrating microarray data by consensus clustering. In International Conference on tools with Artificial Intelligence, PP: 418 – 426 (2003)
22. Topchy, A., Jain, A.K., Punch, W.: A mixture model of clustering ensembles. In SDM, (2004)
23. Boulis, C., Ostendorf, M.: Combining multiple clustering systems. In PKDD, (2004)
24. Gionis, A., Mannila, H., Tsaparas, P..: Clustering Aggregation. International Conference on Data Engineering (ICDE), (2005)
25. Cleuziou, G.: Une méthode de classification non-supervisée pour l'apprentissage de règles et la recherche d'information Thèse de doctorat, Université d'Orléan, (2004)
26. Merz, C. J., Murphy, P. M.: UCI repository of machine learning databases. (1998)