An Overview of Web Data Clustering Practices

Athena Vakali[1], Jaroslav Pokorný[2], and Theodore Dalamagas[3]

[1] Department of Informatics,
Aristotle University of Thessaloniki,
Thessaloniki, 54124, Greece
avakali@csd.auth.gr
[2] Faculty of Mathematics and Physics,
Charles University,
118 00 Praha 1, Czech Republic
pokorny@ksi.mff.cuni.cz
[3] School of Electr. and Comp. Engineering,
National Technical University of Athens,
Zographou, 15773, Athens, Greece
dalamag@dblab.ece.ntua.gr

Abstract. Clustering is a challenging topic in the area of Web data management. Various forms of clustering are required in a wide range of applications, including finding mirrored Web pages, detecting copyright violations, and reporting search results in a structured way. Clustering can either be performed once offline, (independently to search queries), or online (on the results of search queries). Important efforts have focused on mining Web access logs and to cluster search engine results on the fly. Online methods based on link structure and text have been applied successfully to finding pages on related topics. This paper presents an overview of the most popular methodologies and implementations in terms of clustering either Web users or Web sources and presents a survey about current status and future trends in clustering employed over the Web.

1 Introduction

Nowadays, more and more people rely on the World Wide Web to acquire knowledge and information by navigating Websites. However, the exponentially growing of the Web implies difficulties in the way people interact, search, do business etc. Therefore, issues related with organizing the Web content and the structure of a Website become quite popular in recent research efforts.

A lot of previous work has focused on Web data clustering (e.g. [2, 5]). Web data clustering is the process of grouping Web data into "clusters" so that similar objects are in the same class and dissimilar objects are in different classes. Its goal is to organize data circulated over the Web into groups / collections in order to facilitate data availability and accessing, and at the same time meet user preferences. Therefore, the main benefits include: increasing Web information accessibility, understanding users' navigation behaviour, improving information retrieval and content delivery on the Web.

W. Lindner et al. (Eds.): EDBT 2004 Workshops, LNCS 3268, pp. 597–606, 2004.

We can broadly categorize Web data clustering into (I) *users' sessions-based* and (II) *link-based*. The former uses the Web log data and tries to group together a set of users' navigation sessions having similar characteristics. In this framework, Web-log data provide information about activities performed by a user from the moment the user enters a Web site to the moment the same user leaves it [6]. The records of users' actions within a Web site are stored in a log file. Each record in the log file contains the client's IP address, the date and time the request is received, the requested object and some additional information -such as protocol of request, size of the object etc. Figure 1 presents a sample of a Web access log file from an educational Web server (the

```
216.239.46.60 - - [04/Jan/2003:14:56:50 +0200] "GET
/~lpis/curriculum/C+Unix/Ergastiria/Week-7/filetype.c.txt HTTP/1.0"
304 -
216.239.46.100 - - [04/Jan/2003:14:57:33 +0200] "GET
/~oswinds/top.html HTTP/1.0" 200 869
64.68.82.70 - - [04/Jan/2003:14:58:25 +0200] "GET /~lpis/systems/r-
device/r_device_examples.html HTTP/1.0" 200 16792
216.239.46.133 - - [04/Jan/2003:14:58:27 +0200] "GET
/~lpis/publications/crc-chapter1.html HTTP/1.0" 304 -
209.237.238.161 - - [04/Jan/2003:14:59:11 +0200] "GET /robots.txt
HTTP/1.0" 404 276
209.237.238.161 - - [04/Jan/2003:14:59:12 +0200] "GET
/teachers/pitas1.html HTTP/1.0" 404 286
216.239.46.43 - - [04/Jan/2003:14:59:45 +0200] "GET
/~oswinds/publications.html HTTP/1.0" 200 48966
```

Fig. 1. A sample of Web Server Log File

Department of Computer Science in Aristotle University of Thessaloniki). Usually, we need to do some data processing, such as invalid data cleaning and session identification [8]. Data cleaning removes log entries (e.g. images, javascripts etc) that are not needed for the mining process. In order to identify unique users' sessions, heuristic methods are (mainly) used [6], based on IP, and session time-outs. In this context, it is considered that a new session is created when a new IP address is encountered or if the visiting page time exceeds a time threshold (e.g. 30 minutes) for the same IP-address. Then, the original Web logs are transferred into user access session datasets for analysis. The above process is illustrated in Figure 2. Clustering users' sessions are useful for discovering both groups of users exhibiting similar browsing patterns and groups of pages having related content based on how often URL references occur together across them. Therefore, clustering users' sessions is more important in some Web applications, such as on-line monitoring user behaviour, on-line performance analysis, and detecting traffic problems.

Clustering of Web documents helps to discover groups of pages having related content. In general, a Web document can be considered as a collection of Web Pages (a set of related Web resources, such as HTML files, XML files, images, applets, multimedia resources etc.). The main contributions of clustering the Web documents are to improve both the Web information retrieval (e.g. search engines) and content delivery on the Web. In this framework, the Web topology can be regarded as a directed graph, where the nodes represent the Web pages with URL addresses and the edges among nodes represent the hyperlinks among Web pages. Therefore, new techniques are used in order to recognize

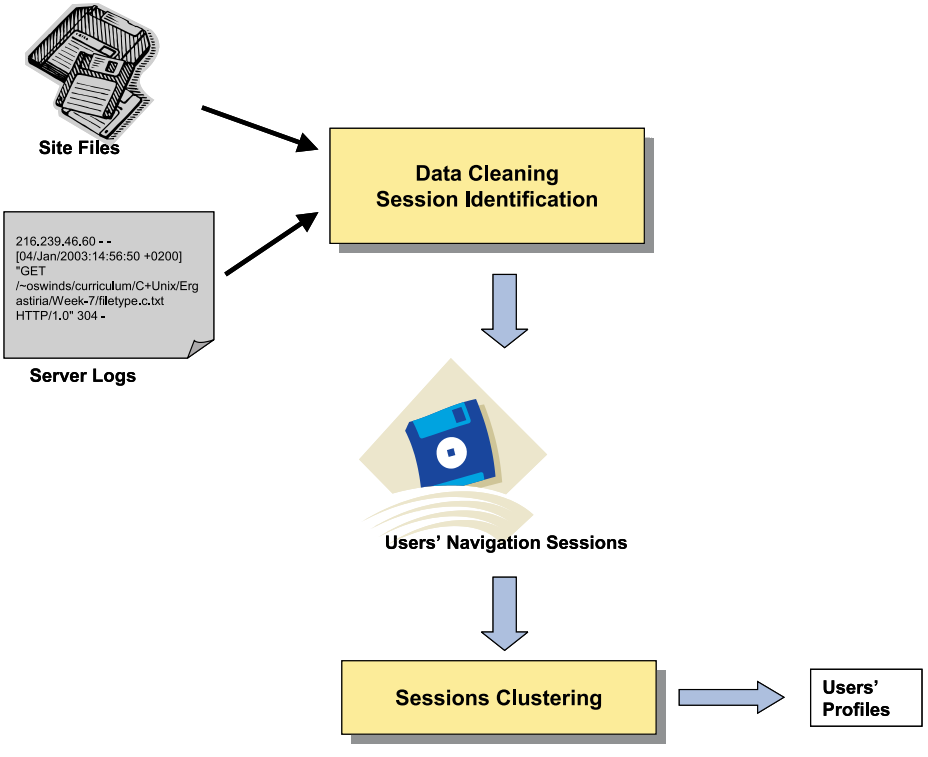

Fig. 2. Clustering users navigation sessions: An overview

and group hypertext nodes into cohesive documents. In this context, the idea of compound documents [11] and logical information units [29] has been evolving recently. A compound document is a set of Web pages that contains at least a tree embedded within the document. A necessary condition for a set of Web pages to form a compound document is that their link graph should contain a vertex that has a path to every other part of the document. Moreover, the notion of Web page communities [18] has gain ground lately in order to organize Web sources and meet Web user requirements. More specifically, a Web community is defined as a set of Web pages that link to more Web pages in the community than to pages outside of the community. A Web community enables Web crawlers to effectively focus on narrow but topically related subsets of the Web.

Much of previous work has focused on understanding the Web user needs and on organizing Web data sources (e.g. pages, documents) [3, 4, 5, 28]. Clustering methodologies have been proven beneficial in terms of grouping Web users in clusters such that the various information circulation activities can be facilitated. In this framework, the XML language is nowadays the standard Web data exchange format. Using XML, one can annotate pages or data exchanged in the Web using tags, providing interoperability and enabling automatic processing of Web resources. Clustering of XML documents brings new challenges, since an XML document encodes not only data but also structure, in one entity [1].

The rest of the paper is organized as follows. Sections 2 and 3 survey the most popular methodologies for Web data clustering. Section 4 presents the XML data clustering perspectives and future trends. Finally, Section 5 concludes the paper and gives some future remarks.

2 Users' Sessions-Based Clustering

The algorithms for users' sessions clustering may be classified into two approaches: similarity-based and model-based (or probabilistic).

2.1 Similarity-Based Clustering Approach

Similarity measures have been proposed towards capturing Web users' common practices whereas effective Web users' logs processing has resulted in the definition of users' session patterns. The first step is to determine the attributes that should be used to estimate similarity between users' sessions (in other words, we determine the users' session representation). Then, it is determined the "strength" of the relationships between the attributes (similarity measures/correlation distance). Finally, clustering algorithms (hierarchical or partitional) are applied in order to determine the classes/clusters to which each user session will be assigned. The hierarchical algorithms define a hierarchy of clustering, merging always the most similar clusters. On the other hand, the partitional approaches (e.g. k-means) define a "flat" clustering into a pre-determined number of clusters (with minimal costs).

Originally, sessions clustering efforts considered sessions as unordered sets of "clicks", where the number of common pages visited was a similarity indication between sessions. The most popular measures that are used are euclidean distance, cosine measure, and Jaccard coefficient. Later on, it was recognized that the order of visiting pages is important, since for example visiting a page A after a page B is not the same information as knowing that both A and B belong to the same session. In this context, the most indicative similarity-based clustering approaches, which have been proposed in the past, can be summarized as follows:

- **Sequence Alignment Method** (SAM) [20], where sessions are chronologically ordered sequences of page accesses. SAM measures similarities between sessions, taking into account the sequential order of elements in a session. SAM distance measure between two sessions is defined as the number of operations that are required in order to equalize the sessions (dynamic programming method to match related sessions).
- **Generalization-Based Clustering** [16] uses page URLs to construct a hierarchy, for categorizing the pages (partial ordering of Web pages, leaf is the Web page file, non-leaf nodes are the general pages). Then, the pages in each user session are replaced by the corresponding general pages and clustered using the BIRCH algorithm [34].
- **Clickstream (Sessions) Analysis** [25] evaluates the similarities between two clickstreams. More specifically, the similarity between two clickstreams requires finding similarity / distance between two page views. Since semantic analysis is not possi-

ble, the degree of similarity between two page views is proportional to their relative frequency of co-occurrence. In this context, authors in [3] cluster two clickstreams using as criterion the length of the largest subsequence common (LCS) between two clickstreams.

2.2 Model-Based (or Probabilistic) Clustering Approach

Model-based clustering techniques have been widely used and have shown promising results in many applications involving Web data [2, 4]. More specifically, in the model-based approach the users' sessions clusters are generated as follows:

1. A user arrives at the Web site in a particular time and is assigned to one of a predetermined number of clusters with some probability. The number of clusters is determined by using several probabilistic methods, such as BIC (Bayesian Information Criterion), bayesian approximations, or bootstrap methods [14].
2. The behaviour of each cluster is governed by a statistical model and the user's behavior is generated from this model to that cluster.

Each cluster has a data-generating model with different parameters for each cluster. Therefore, this model can be well defined, if only we learn the parameters of each model component, which are the probability distribution used to assign users to the various clusters and the number of components. The model structure can be determined by model selection techniques and parameters estimated using maximum likelihood algorithms, e.g., the EM (Expectation-Maximization) algorithm [10]. Markov models (e.g. first order Markov models, or Hidden Markov models) [2, 4] are the most indicative models that are used for users' sessions. Once the model is learned, we can use it to assign each user to a cluster or fractionally to the set of clusters. Compared to similarity-based methods, model-based methods offer better interpretability since the resulting model for each cluster directly characterizes that cluster. Model-based clustering algorithms often have a computational complexity that is "linear" in the number of data objects under certain practical assumptions.

3 Link-Based Clustering

Due to the high heterogeneity of Web documents, the information seeking on the Web has many difficulties. Recently, researchers suggested to apply clustering to Web documents in order to improve the Web searching process [5]. In this approach the Web is treated as a directed graph. Previous researches have shown that the Web presents strong connectivity, which means that the Web pages with similar topical content have "dense" links between them. Therefore the goal is to cluster in the same group the Web pages with similar content and this can be achieved by eliminating arcs between dissimilar pages. The advantage of this approach than the previous one (users' sessions-based clustering) is that the similarity/dissimilarity of pages is determined by the structure of Website. Another interesting feature of this approach is that it does not need to specify the number of clusters as a separate parameter. On the other hand, the users' sessions-based algorithms have several tuneable parameters (such as the number of clusters) that may affect significantly the clustering method.

In this context, various approaches for clustering of Web documents using the Website topology have been proposed in the literature. The most indicative of them are the following:

- **Web Communities** were proposed [18] on the basis of the evolution of an initial set of hubs (pages that points to many relevant ones) and authorities (relevant pages that pointed to by many hubs), such that the behavior of users is captured with respect to the popularity of existing pages for the topic of interest [21]. More specifically, a Web graph consists of several hundred thousand of sub-graphs, the majority of which correspond to communities with a definite topic of interest. In this framework, several approaches have been proposed (e.g. Maximum Flow and Minimal cuts, graph cuts and partitions, PageRank algorithm etc.) in order to identify them [12].
- **Compound Documents** are represented as Web graphs, which are either strongly connected or nearly so. In graph theory, a directed graph is strongly connected if there is a path from every vertex to every other vertex. Authors in [11] present new techniques for identifying and working with such compound documents. In this work, the compound documents are identified if they contain at least one of the following graph structures within their hyperlink graph:
  - Linear paths: There is a single ordered path through the document, and navigation to other parts of the document are usually secondary (e.g. news sites with next link at the bottom)
  - Fully connected: These types of documents have on each page, links to all other pages of the document (e.g. short technical documents and presentations)
  - Wheel documents: They contain a table of contents (toc) and have links from this single toc to the individual sections of the document (toc is a kind of hub for the document)
  - Multi-level documents: Complex documents that may contain irregular link structures such as multilevel table of contents

4 XML Data Clustering Perspectives and Future Trends

The XML language is becoming the standard web data exchange format, providing interoperability and enabling automatic processing of web resources. Using XML, one can annotate pages or data exchanged in the Web using tags. Tags can be exploited by web scripts or programs to identify data easier, since they give meaning and structure to data. To this extend, an XML document encodes data and structure in one entity, perfectly suited for describing semistructured data [1], that is schema-less and self-describing pieces of information.

Processing and management of XML documents have already become popular research issues [1]. Clustering XML documents refers to the application of clustering algorithms to detect groups of XML documents that share similar characteristics. The estimation of *similarity* is closely related to the *distance metric* exploited by the clustering algorithm. We consider the clustering of XML documents as a problem with two dimensions: *content* and *structure*. The content dimension needs distances that estimate similarity in terms of the textual content inside elements in XML documents, while the

structure dimension needs distances that estimate similarity in terms of the structural relationships of the elements in XML documents. We next discuss each one of these two dimensions.

4.1 Clustering XML Documents: The Content Dimension

Clustering XML documents by content is mainly based on the application of traditional IR techniques [31] to define distance metrics that capture the content similarity for pieces of text. A new requirement for such a task arises from the need to support *granularity of indexes* in XML documents. Applications may restrict the context of interest for the clustering procedure to certain XML elements instead of the whole document. Flexible models to manipulate structured documents, taking into consideration their granularity, have been examined in older works for SGML document management [23] and structured text databases retrieval [32]. The main issues to consider in the case of content dimension in the clustering procedure are:

1. the generation of *dynamic statistics*: these statistics include statistical information (for example frequencies) for the terms inside tags, for various parts of the XML documents,
2. the design of *hierarchical indexes*: these indexes should calculate efficiently the distance metrics required by the clustering procedure for various parts of the XML documents, and should be easily maintained to reflect changes in statistics.

Current work examples where such issues are explored include XML retrieval systems like JuruXML [22], XXL [30], XIRQL [15], and hierarchical indexing methodologies, like the flexible indexes [9], and the dynamic generation of vector spaces [17].

Another interesting issue arises from viewing XML documents under a data-centric approach. Treating elements as *categorical attributes* (e.g. values "red", "green","blue" for the element color as categorical attribute) or the values of elements as *market basket data* (e.g. values of the element price) brings a data mining perspective in the task of grouping XML documents by content. The challenge is the application of data mining techniques (like for example the ROCK algorithm for clustering categorical attributes [19]) in the context of XML documents, under the requirement of granularity.

4.2 Clustering XML Documents: The Structure Dimension

Modeling XML documents with tree models [1], we can face the 'clustering XML documents by structure' problem as a 'tree clustering' problem, and exploit *tree edit distances* to define metrics that capture structural similarity [26]. Assuming a set of tree operations (e.g. insert, delete, replace node) and a cost model to assign costs for each one, the tree edit distance between two trees T_1 and T_2 is the minimum cost among the costs of all possible tree edit sequences that transform T_1 to T_2. The tree edit distance can estimate the structural similarity between trees that represent XML documents, and can be included in clustering procedures to identify clusters of structurally similar XML documents.

However, since tree edit distance calculations are quite intensive, vector-based approaches that capture the hierarchical relationships of tree structures should be also explored as a basis to design appropriate efficient indexes.

The main issues to consider in the case of structure dimension in the clustering procedure are:

1. the need or not for ordering in the elements of XML documents,
2. the difference in the importance of elements as structural primitives in the hierarchy imposed by the XML document: the deletion of a top element, e.g. vehicles, might be more important than the deletion of a bottom element, e.g. three-wheel-bicycles, in an XML document.
3. semantic dissimilarities: different tags might refer to semantically similar elements, e.g. elements price and cost.

Current work examples where such issues are explored include change detection methodologies [7], clustering methodologies like [24], indexes that used for time series management, and bitmaps to model tree-like structures [13, 33].

5 Conclusions

It seems that both Web modelling and Web searching need to be improved. An emphasis is put on increasing expressiveness of modelling tools and Web content capturing. New research directions include:

- developing techniques to efficiently cluster the entire web based e.g. on similarity searches in high dimensional spaces,
- developing scalable robust fuzzy techniques to model noisy data sets containing an unknown number of overlapping categories,
- developing techniques like e.g. locality sensitive hashing, in which web pages are hashed in such a way that similar pages have a much higher probability of collision than dissimilar pages,
- exploring new techniques to handle linguistic and textual features.

Another sources of new research directions appear in considering so called deep Web. Many of its sources are structured (stored in relational DBMSs) according to a specified schema. Such schemas define the object domain of a source (e.g., goods, movies) and its query capabilities (e.g., by price, actor). Clustering sources by their query schemas (i.e., attributes in query interfaces) is possible. This approach is essentially clustering categorical data. Clusters are often governed by statistical distributions.

The last but not least is a dynamics of the Web. The methods mentioned usually work on a Web samples that are static, i.e. they represent only a snapshot of the real Web. It is a challenge to model a dynamic Web and to develop methods for an efficient implementation of its structure and content.

References

1. S. Abiteboul, P. Buneman, and D. Suciu. *Data on the Web.* Morgan Kaufmann, 2000.
2. P. Baldi, P. Frasconi, and P. Smyth. *Modeling the Internet and the Web* Wiley, 2003.

3. A. Banerjee and J. Ghosh. Clickstream clustering using weighted longest common Subsequences. In *Proceedings of Workshop on Web Mining, SIAM Conference on Data Mining*, pages 33-40, Chicago, USA, April 2001.
4. I. V. Cadez, D. Heckerman, C. Meek, P. Smyth, and S. White. *Model-based clustering and visualization of navigation patterns on a Web site.* Data Mining and Knowledge Discovery, 7(4):399-424, 2003.
5. S. Chakrabarti. *Mining the Web.* Morgan Kaufmann, 2003.
6. Z. Chen, A. Wai-Chee Fu, and F. Chi-Hung Tong. Optimal algorithms for finding user access sessions from very large Web logs. *World Wide Web: Internet and Information Systems*, 6:259-279, 2003.
7. G. Cobena, T. Abdessalem and Y. Hinnach. *A comparative study for XML change detection.* Technical Report, INRIA, France, 2000.
8. R. Cooley, B. Mobasher, and J. Srivastava. Data preparation for mining World Wide Web browsing patterns *Knowledge Information Systems*, 1:5-32, 1999.
9. H. Cui and J.-R. Wen. Hierarchical indexing and flexible element retrieval for structured document. In *Proceedings of ECIR*, 2003.
10. A. P. Dempster, N. M. Laird, and D. B. Rubin. *Maximum likelihood from incomplete data via the EM algorithm.* Statistics Society B, 39:1-22, 1997.
11. N. Eiron and K. S. McCurley. Untangling compound documents on the Web. In *Proceedings of ACM Hypertext,*, pages 85-94, 2003.
12. G. W. Flake, S. Lawrence, C. Lee Giles, and Frans Coetzee. *Self-organization and identification of Web Communities* IEEE Computer,35(3), 2002.
13. S. Flesca, G. Manco, E. Masciari, L. Pontieri and A. Pugliese. Detecting similarities between XML documents. In *Proceedings of WebDB Workshop*, 2002.
14. C. Fraley and A. Raftery. *How many clusters? Which clustering method? Answers via model-based cluster analysis.* Computer Journal, 41, 1998.
15. N. Fuhr and K. Groβjohann. XIRQL: a query language for information retrieval in XML documents. In *Proceedings of ACM SIGIR*, 2001.
16. Y. Fu, K. Sandhu, and M-Y Shih. Clustering of Web users based on access patterns. In *Proceedings of WEBKDD*, 1999.
17. T. Grabs and H.-J.Org Schek. Generating vector spaces on-the-fly for flexible XML retrieval. In *Proceedings of XML and IR Workshop*, 2002.
18. G. Greco, S. Greco, and E. Zumpano. *Web communities:models and algorithms.* World Wide Web, 7(1):58-82, 2004.
19. S. Guha, R. Rastogi and K. Shim *ROCK*: *A robust clustering algorithm for categorical attributes.* ACM SIGMOD Record, 25(5), 2000.
20. B. Hay, K Vanhoof, and G. Wetsr Clustering navigation patterns on a Website using a sequence alignment method. In *Proceedings of 17th International Joint Conference on Artificial Intelligence*,Seattle, Washington, USA, August, 2001.
21. J. M. Kleinberg. Authoritative sources in a hyper-linked environment. In *Proceedings of the 9th ACM-SIAM Symposium on Discrete Algorithm*, 1998.
22. Y. Mass, Matan Mandelbrod, E. Amitay, Yoelle Maarek, and Aya Soffer Juru XML - an XML retrieval system at INEX'02 In *Proceedings of INEX*,Dagstuhl, Germany, December 2002.
23. S. H. Myaeng and D-H Jang. A flexible model for retrieval of SGML documents. In *Proceedings of ACM SIGIR*, 1998.
24. A. Nierman and H. V. Jagadish Evaluating structural similarity in XML documents. In *Proceedings of the WebDB Workshop*, Madison, Wisconsin, USA, June, 2002.
25. R. Kothari, P. A. Mittal, V. Jain, and M. K. Mohania. On using page cooccurrences for computing clickstream similarity. In *Proceedings of the 3rd SIAM International Conference on Data Mining.*, San Francisco, USA, May 2003.

26. D. Sankoff and J. Kruskal. *Time warps, string edits and macromolecules, the theory and practice of sequence comparison.* CSLI Publications, 1999.
27. R. R. Sarukkai. *Link prediction and path analysis using Markov chains.* Computer Networks, 33:377-386, 2000.
28. Z. Su, Q. Yang, H. H. Zhang, X. Xu, and Y. Hu Correlation-based document clustering using Web logs. In *Proceedings of the 34th Annual Hawaii International Conference on System Sciences (HICSS-34)*, Maui, Hawaii, January, 2001.
29. K. Tajima, K. Hatano, T. Matsukura, R. Sano, and K. Tanaka. Discovery and retrieval of logical information units in Web. In *Proceedings of the Workshop on Organizing Web Space (WOWS 99)*, pages 13-23, Berkeley, USA, August 1999.
30. A. Theobald and G. Weikum. The Index-Based XXL Search engine for querying XML data with relevance ranking. In *Proceedings of the EBDT Conference*, 2002.
31. R. Baeza-Yates and B. Ribiero-Neto *Modern information retrieval.* Addison-Wesley, 1999.
32. R. Baeza-Yates and G. Navarro. *Integrating contents and structure in text retrieval.* ACM SIGMOD Record, 25(1), 1996.
33. J. Yoon and V. Raghavan and V. Chakilam and L. Kerschberg. *BitCube*: *A three-dimensional bitmap indexing for XML documents.* Journal of Intelligent Information Systems, 17, 2001.
34. T. Zhang, R. Ramakrishnan, and M. Livny. BIRCH: An efficient data clustering method for very large databases. In *Proceedings of the International Conference Management of Data (ACM-SIGMOD)*, pages 103-114, Montreal, Canada, June, 1996.