An Online Analytical Processing Framework for Large Hypertext Collections

Mandar R. Mutalikdesai
International Institute of Information Technology
Bangalore 560100, INDIA
mandar@iiitb.ac.in

Srinath Srinivasa
International Institute of Information Technology
Bangalore 560100, INDIA
sri@iiitb.ac.in

ABSTRACT

Hypertext collections abound in various contexts. In such collections, the combination of content and hyperlink structures reflect several interesting facts. Presently, standalone analyses have to be implemented to infer them. However, it is cumbersome to script separate experiments for each individual analysis. In the wake of growing amounts of hypertext data, we propose a *unified framework for online analytical processing* for such collections. Using this framework, a user will be able to conveniently provide various *analytical queries* over desired portions of hypertext collections. To implement this seamlessly, we have identified four major challenges to address: (1) A data model with support for handling user-defined search-spaces, (2) Storage structures supporting quick creation and retrieval of various views of data, (3) A query model for providing complex analytical queries, and (4) A query processor for online execution of analytical queries using indexes and summaries.

1. BACKGROUND AND MOTIVATION

Growing amounts of hypertext data can be found in various contexts like weblogs and online journals, intranet webs, the World Wide Web (WWW), online communities, intra-organizational wikis and other collaborative content management platforms. The combination of content and linkage structure of a hypertext collection encloses interesting information about various phenomena [7]. For example, the existence of cybercommunities (c.f. [15]), the hierarchical structure of an organization, the documents similar to a given document (c.f. [11]), the popularity and importance of documents (c.f. [5]), the probability of reaching a document from any other document by following a sequence of hyperlinks, the trends in the growth of the hypertext network, etc. can all be determined by analyzing a hypertext web. Graph-theoretic analysis yields several useful insights into the dynamics of hypertext webs.

However, separate experiments need to be scripted for each individual analysis presently. It is cumbersome to write and manage a large number of standalone scripts every time a hypertext collection is analyzed for some phenomenon. Moreover, many of these analyses require large amounts of time owing to the complexity of operations as well as the size of the underlying hypertext collection. Hence, we propose a unified framework for online analytical processing (OLAP) in large hypertext collections.

Using the proposed OLAP framework, the user will be able to provide various kinds of *analytical queries*. Presently, we have identified four categories of analytical queries, which we discuss in section 2. These queries are termed "analytical" because they address the *aggregate properties* of hypertext webs. They cannot be answered by simple data extraction and reporting models. They need to sift through large hypertext graphs before returning their answers.

The user can query not only the entire hypertext collection, but also desired *subsets* of it. Such user-defined abstractions of search-spaces reflect domain knowledge. The queries that are answered within such a context are likely to be more useful than those that blindly search the entire data set. We also envisage support for index structures and pre-computed summaries in our framework, so that the queries execute in an *online* fashion, i.e. in much lesser time compared to brute-force computations.

In order to realize the proposed OLAP model for hypertext, we have identified the following as major challenges: (1) Designing a data model that supports handling of user-defined views of data, (2) Designing a storage model that supports quick creation, updation, storage and retrieval of various views of the underlying data, (3) Designing a query interface for constructing complex analytical queries, and (4) Designing a query processing engine for online execution of analytical queries using indexes, materialized and/or virtualized views, schemata and pre-computed summaries. We explore each of these challenges in section 3.

The first author of this paper is a first-year PhD student. Hence, this work is in its formative stages, and is largely exploratory in nature. The expected duration of this thesis work is 4 years.

2. TAXONOMY OF QUERIES

In this section, we present a taxonomy of the queries that are envisaged for our OLAP model for hypertext collections. We have presently identified four categories of queries.

2.1 Chronological Analysis

Hypertext collections evolve over time. This evolution can occur in two ways: (1) Evolution of document content, and (2) Evolution of the hyperlink graph. Analysis of the dynamics of a hypertext collection over intervals of time yields useful insights into the patterns of its evolution. For example, it is interesting to study the change in the *local*

Proceedings of the VLDB2006 Ph.D. Workshop
Seoul, Rep of Korea, 2006

PageRank of a given document in a given hypertext subset over time, since it represents the change in the "temporal importance" of the document in the community represented by the hypertext subset. Such analyses, which are conducted across time, are defined as chronological analyses.

Some examples of chronological analysis queries are: (1) *Plot the degree distributions of an organization's intranet web over the last 6 months at 1 month intervals, with distinct legends for each interval*, and (2) *What has been the topic of most discussion in January 2006 among the domains* `blogspot.com`, `typepad.com` *and* `livejournal.com` *put together in India?*

2.2 Classification Analysis

The broad category of classification analysis addresses this question: *Can I classify groups of hypertext documents according to some theme, based on similarities in their graph-theoretic properties?* It would be interesting to classify document groups according to several "themes" such as *content*, *size*, *activity*, *collaboration*, *organizational structure*, etc. For example, suppose the degree distributions for the websites of the department of Computer Science and the department of Electrical Engineering at some university are similar. Let us assume that these distributions are only for "non-nepotistic" links, i.e. only for incoming and outgoing links outside the concerned website. Similarity in such distributions for the two websites may indicate the similar extents to which the two departments *collaborate* with external agencies. Here, "extent of collaboration" is a notion that is conceived by the user. However, this notion is captured using non-nepotistic degree distributions. Such an analysis is interesting because it allows classification of "document clusters" based on their graph-theoretic properties, as per a user-conceived notion.

Some examples of classification analysis queries are: (1) *Classify the academic webs of various countries (the domains* `ac.in`, `edu`, `edu.np`, `edu.sg`, *etc.) based on the diameters of their respective largest strongly connected components, as closely-knit (<8), medium-knit (9-19) and loosely-knit (>20)*, and (2) *Cluster the pages of a research lab's internal wiki as belonging to "Project A", "Project B" or "Other Projects."*

2.3 Structural Analysis

Structural analysis deals with the discovery of structural elements such as subgraphs, co-citations, bibliographic couplings, cycles, bipartite-cores, cliques, strongly connected components, shortest paths and other structural motifs in a given search-space. Such structural analyses yield meaningful insights into the semantics of the underlying web. For example, it is interesting to mine bipartite-cores in a given hypertext collection, because bipartite-cores indicate the existence of cybercommunities [15].

Some examples of structural analysis queries are: (1) *List all webpages in the* `ac.in` *domain, which have been co-cited the most number of times by the top-500 webpages in the result set obtained from the search for the phrase "best universities India"*, and (2) *Check for the existence of bipartite-cores in the top-100 result set obtained from the search for the phrase "Indian students organization in USA."*

2.4 Correlation Analysis

Suppose a user identifies an interesting trend or phenomenon in some hypertext collection. It may interest her to determine the manner in which that phenomenon occurs in other hypertext collections. For example, it may be useful to correlate the density distributions of document adjacency in the blogspace and Wikipedia, since both are collections of autonomously created documents. Such analyses correlating phenomena across various hypertext webs give a useful perspective of their comparative dynamics.

Some examples of correlation analysis queries are: (1) *Plot the PageRank distributions of the top-100 result sets obtained from the search on the contemporary topics "FIFA world cup" and "Wimbledon championship", with distinct legends for each topic*, and (2) *Plot the frequency distributions of the occurrence of the phrase "information retrieval" in the domains* `ac.in`, `ac.uk` *and* `edu`, *with distinct legends for each domain.*

3. MAJOR CHALLENGES

In section 2, we have identified four categories of analytical queries. We now discuss the challenges involved in implementing them.

3.1 Data Model

In our model, when the user begins a query session, her default search space is the entire hypertext collection. However, as shown in the examples in section 2, the user is allowed to abstract her own search-space for executing queries. The question that needs to be addressed here is: *In what way can a user define a search-space?* We explore the multidimensional data model to address this question.

In traditional OLAP systems (c.f. [8]), the user can build *data cubes* using various *dimensions* and query the *facts*. A data cube is an abstraction of a user-defined search-space. The dimension-tables contain records relating to the facts. In our OLAP model too, we propose using pre-defined facts and dimensions for building data cubes. The dimensions can be hierarchies or simple data. Each fact and its associated dimensions can be modeled as a *star schema*.

There are two facets to an OLAP model for hypertext collections: *Document Text* and *Hyperlink Graph*.

Multidimensional models have been employed in the context of text collections, mainly for information retrieval [16, 18]. In a text collection, examples of dimensions are time of creation, location, subject category, author, search key, etc., while term-occurrence in documents is an example of a fact. A sample analytical query on document text is: *Select the top-10 relevant documents containing a given search key, which have been created in 2004 in USA*. However, in an OLAP model for hypertext, facts and dimensions pertain to hyperlinks as well.

Introducing the concept of hyperlink graph into a multidimensional model is a challenge. Some of the questions that arise in this regard are as follows:

1. What do we mean by facts about hyperlinks?
2. What analytical queries can be issued for hyperlink-based facts, and how?
3. What dimensions can be defined in terms of hyperlinks?

Some of the facts with respect to the hyperlink graph are: PageRank of documents, in-degrees and out-degrees of

documents, centrality measures of documents, diameter of a hyperlink subgraph, etc. The question that arises here is: *How can we view these properties in aggregations?*

Consider the following example: Usually, in a hypertext graph, nodes represent documents and edges represent hyperlinks. However, we can model a hypertext graph such that nodes represent entire *websites* instead of individual documents, and edges represent hyperlink connectivity between websites instead of documents. That is, an edge from a node A to node B represents all the hyperlinks between pages in website A to pages in website B.

PageRank is a hyperlink-based property of a single document. However, in the modified hypertext graph as above, it can be aggregated as a property of a website instead of individual documents. Such *graph aggregations* can take place at various levels of granularity. For instance, at the level of directories in a website, at the level of websites, at the level of sub-domains (e.g., `.ac.in`), at the level of domains (e.g., `.com`), etc. The challenge here is to develop efficient techniques for enabling such aggregations in on online fashion.

Dimensions can be defined in terms of hyperlink-based properties too. Consider the following query: *Select pairs of documents containing a given search key, such that the shortest path between them is less than 5, and which have been created after 2002 in Asia.* Here, *shortest path* is a dimension based on a link-based property. Similarly, several other graph properties can be projected as dimensions. It is challenging to enable slicing and dicing using hyperlink properties in addition to document properties.

3.2 Storage Model

The query processor needs to know *where* and *how* the entire hypertext web is stored. The following are some questions that arise in this regard.

1. How should documents be stored?
2. How should the hyperlink graph be stored?
3. How should data-store updates be handled?

A hypertext collection typically contains a large number of documents. The query processor should be able to quickly locate the contents of the documents required to construct the view of a data cube. To derive a scheme of file structures for storing the documents such that view construction and query answering are optimized, is challenging.

Typically, a hypertext collection contains a large number of hyperlinks too. Along with document content, hyperlink information is also required to build views and answer queries. Storing the hyperlink graph such that accessing the information about any set of hyperlinks is optimized, is challenging.

Chronological analysis queries are executed over "historical" data. Therefore, when the data-store is updated with fresh crawls of hypertext, the OLAP system will have to make snapshots of the "stale" data, so that it's properties are not overwritten. The storage model has to be able to support content updates as well as hyperlink updates without losing information about the previous crawl. It is challenging to design and optimize such an updation scheme.

3.3 Query Model

We intend to export a query interface to the user to enable interaction with the OLAP model. Using well-defined query constructs, the user will be able to execute analytical queries.

We have presently identified four categories of analyses to support in our model. Each of these categories represents a wide range of queries, as exemplified in section 2. However, the thesis might become overly ambitious in trying to capture all forms of queries in a single high-level query language. The difficulty in mapping aggregate reasoning and analysis tasks to a high-level query language is evident from the relatively slow rate of progress in supporting data mining in database systems [17]. Hence, presently, we intend to identify exactly what queries will be supported by our querying system, and export a query-specific interface to the user, instead of defining a generic high-level query algebra.

The query interface should consist of two modules: Data Definition Module (DDM) and Data Manipulation Module (DMM). DDM constructs can be used for creating, updating, storing, loading and deleting views, schemata, indexes, historical snapshots and pre-computed summaries. The DMM can be used for posing the analytical queries supported by the system.

3.4 Online Query Processing

Views of data cubes can be either materialized or computed on demand, in order to process a query. Materialized views should be well-known to the query processor. If a materialized view that can be used to answer a query exists, it should be loaded into memory and the query should be answered. This reduces query-response time.

The analyses planned to be addressed by this framework involve time-consuming operations over large data sets. We envisage the use of appropriate index structures to answer queries quickly and efficiently. Index structures need to be built not only over the entire hypertext collection, but also over the user-defined views. Using these index structures, analytical queries can be answered in an *online* fashion.

Certain queries involve the computation of "standard values" like diameters, shortest paths, average in-degrees, etc. We propose to identify a class of such values that can be pre-computed and stored as summaries along with their respective materialized views. These summaries can be used to speed up the query processing.

4. RELATED WORK

Several individual graph-theoretic analyses have been conducted based on the link structure of the Web. Broder, et al. used generalizations of the Breadth-First Search algorithm to traverse a web-crawl of around 200 million pages, and discovered that the macroscopic structure in the Web is in the form of a "bowtie" [6]. Using the notion of hubs and authorities (c.f. [13]), Gibson, et al. inferred web communities from a natural type of hierarchical generalization formed by cores of authoritative pages linked to by hub pages [12]. Kumar, et al. inferred web communities by identifying bipartite-cores in the link topology [15]. Bharat, et al. computed neighborhoods of webpages and used them for a fast browsing and searching experience [4]. Several studies have also investigated the power-law distributions on the Web [6, 14].

Graph-theoretic techniques and machine learning algorithms have been employed for content analysis of hypertext webs (c.f. [2, 19]); for example: clustering and classification of web documents based on their textual content.

In this work, we propose a *unified* model for analysis of

the content as well as the link structure of various kinds of hypertext networks like intranets, the WWW, wikis, etc. We propose to provide an OLAP tool with a query model, such that the user can execute various kinds of analytical queries on hypertext networks of choice. We have identified four different categories of queries to support in our model.

Several network analysis tools are available for analyzing large networks. SocSciBot [20], Pajek [3] and IKNOW [9] are examples of such tools. Of these, SocSciBot and IKNOW are prominently used for network analysis in the context of the Web. SocSciBot supports operations on link structure like counting in-links and out-links between sites, reporting the most frequent link targets, removing internal site links to ensure non-nepotism, calculating PageRank statistics, calculating topological components as in the bowtie model (c.f. [6]) and calculating diameters over various collections of webpages [20]. The SocSciBot crawler builds these collections by accepting the homepage of the URL to be crawled, and crawling the website online. In our OLAP model, we propose to abstract user-defined search-spaces over stored collections of hypertext data. The user is able to conduct various kinds of aggregate analyses over the content as well as link structure of the search-spaces.

IKNOW provides a mapping, visualization and measurement system that can help organizations in studying the patterns of knowledge and information flow through the organization's internal network. It provides for identifying critical patterns of knowledge distribution and information flow [9]. In comparison, our model supports a broader range of analytical queries. We envisage queries not only for chronological identification of interesting patterns, but also for graph-theoretic classification of documents based on user-defined notions, structural analysis and comparative analysis across various collections of hypertext.

Google Trends[1] analyzes a portion of Google web searches to compute how many searches have been made for the terms entered by the user relative to the total number of searches made on Google over time. Analogous to this, we propose to support various kinds of queries for historical and correlation analysis of hypertext collections in our model.

Several graph database models have addressed the challenge of answering structural queries ranging from finding simple paths to detecting structural similarity and subgraph isomorphism. A survey of graph database models can be found in [1]. In our model also, we are faced with the challenge of answering structural queries over desired search-spaces. We intend to overcome this challenge by using appropriate index structures.

Traditional OLAP systems over relational datawarehouses (c.f. [8]) aggregate and analyze large groups of diverse data involved in complex relationships. They provide the user the ability to perform trend analysis, comparative analysis, time-series analysis, etc. in different dimensions such as time, region, product, etc. Traditional OLAP systems support various analytical operations through aggregation, drill-down, and slicing and dicing of data. Our OLAP model for hypertext collections is also envisaged to have these capabilities.

OLAP techniques have been applied in the context of information retrieval (IR). McCabe, et al. used the hierarchical information within documents for searching [16]. For a given text collection, they defined several dimensions as well as a fact-table for term occurrence. They conceptualized a star-schema model using the dimensions and the fact, and used the multidimensional database model for IR. Priebe and Pernul designed an enterprise knowledge portal that integrated OLAP and IR functionalities to access the structured data stored in a datawarehouse as well as the unstructured data stored in document collections [18]. They defined dimensions and fact-tables based on the underlying data set and represented them using RDF and RDF Schema. In our model, we intend to use the multidimensional data model for various kinds of hypertext analysis.

MapReduce is a programming model for processing large data sets [10]. Programs written in the functional style are automatically parallelized and executed on a large cluster of machines. Operations like distributed grep, distributed sort, count of URL access frequency, term-vectors per host, inverted index, reverse web-link graph, etc. are reduced to MapReduce computations. In our model also, it might be desirable to have parallelizable query processing on the lines of MapReduce. However, we list this as work for the future.

5. CONCLUSION

With growing amounts of hypertext data around us, there is a need to analyze it to discover interesting patterns, trends and phenomena. Particularly with the Web 2.0 paradigm bringing in technologies like blogs, Wikis, and other knowledge sharing and collaboration tools, the need for analyzing hypertext data for understanding the dynamics of societies and organizations is significant. In this thesis work, we propose a unified model for various kinds of online analytical processing in hypertext collections. We have identified four categories of analytical queries to be supported in our model. In order to implement the OLAP model, we have identified four major challenges. We have also discussed initial ideas for addressing them.

Tasks for the near future include addressing the issue of data and storage structures. After addressing all the challenges discussed in this paper, we propose to add a visualization engine to the model to visualize query results.

Intuitively, it seems that the techniques developed as part of this thesis may be general enough to apply in other contexts too, where large amounts of disk-bound graph data are analyzed. Examples of such contexts could be networks of social relationships, bibliographic citation graphs, protein interaction networks, UML diagrams of software design, etc.

6. REFERENCES

[1] R. Angles and C. Gutierrez. Survey of Graph Database Models. *Technical Report Number TR/DCC-2005-10, Computer Science Department, Universidad de Chile.*

[2] A. Antonacopoulos and J. Hu (eds.). *Web Document Analysis.* World Scientific, 2004.

[3] V. Batagelj and A. Mrvar. Pajek: Analysis and Visualization of Large Networks. M. Junger and P. Mutzel (eds.), *Graph Drawing Software*, 2003.

[4] K. Bharat, A. Broder, M. Henzinger, P. Kumar and S. Venkatasubramanian. The Connectivity Server: Fast Access to Linkage Information on the Web. *Proc. of the 7th International World Wide Web Conference*, 1998.

[1] http://www.google.com/trends

[5] S. Brin and L. Page. The Anatomy of a Large-Scale Hypertextual Web Search Engine. *Proc. of the 7th International World Wide Web Conference*, 1998.

[6] A. Broder, R. Kumar, F. Maghoul, P. Raghavan, S. Rajagopalan, R. Stata, A. Tomkins and J. Wiener. Graph Structure in the Web: Experiments and Models. *Proc. of the 9th International World Wide Web Conference*, 2000.

[7] S. Chakrabarti. *Mining the Web: Discovering Knowledge from Hypertext Data*. Morgan Kaufmann, 2002.

[8] S. Chaudhuri and U. Dayal. An Overview of Data Warehousing and OLAP Technology. *ACM SIGMOD Record, 26(1)*, 1997.

[9] N. Contractor, D. Zink and M. Chan. IKNOW: A Tool to Assist and Study the Creation, Maintenance and Dissolution of Knowledge Networks. *In Toru Ishida (ed.), Community Computing and Support Systems, Lecture Notes in Computer Science*, 1998.

[10] J. Dean and S. Ghemawat. MapReduce: Simplified Data Processing on Large Clusters. *Proc. of the 6th Symposium on Operating System Design and Implementation*, 2004.

[11] J. Dean and M. Henzinger. Finding Related Web Pages in the World Wide Web. *Proc. of the 8th International World Wide Web Conference*, 1999.

[12] D. Gibson, J. Kleinberg and P. Raghavan. Inferring Web Communities from Link Topology. *Proc. of the 9th ACM Conference on Hypertext and Hypermedia*, 1998.

[13] J. Kleinberg. Authoritative Sources in a Hyperlinked Environment. *Journal of the ACM, 46(5)*, 1999.

[14] J. Kleinberg, R. Kumar, P. Raghavan, S. Rajagopalan and A. Tomkins. The Web as a Graph: Measurements, Models and Methods. *Proc. of the 5th International Computing and Combinatorics Conference*, 1999.

[15] R. Kumar, P. Raghavan, S. Rajagopalan and A. Tomkins. Trawling Emerging Cyber Communities Automatically. *Proc. of the 8th International World Wide Web Conference*, 1999.

[16] M. C. McCabe, J. Lee, A. Chowdhury, D. Grossman and O. Frieder. On the Design and Evaluation of a Multidimensional Approach to Information Retrieval. *Proc. of the 23rd ACM SIGIR Conference on Research and Development on Information Retrieval*, 2000.

[17] G. Piatetsky-Shapiro. Knowledge Discovery in Databases: 10 Years After. *SIGKDD Explorations, 1(2)*, 2000.

[18] T. Priebe and G. Pernul. Ontology-based Integration of OLAP and Information Retrieval. *Proc. of the 14th International Workshop on Database and Expert Systems Applications*, 2003

[19] A. Schenker, H. Bunke, M. Last and A. Kandel. *Graph-Theoretic Techniques for Web Content Mining*. World Scientific, 2005.

[20] M. Thelwall. *Link Analysis: An Information Science Approach*. Elsevier, 2004.