Volume 2 ▪ Issue 3 ▪ August 2011

Editor-in-Chief
Professor Walid Aref

INTERNATIONAL JOURNAL OF DATA ENGINEERING (IJDE)

ISSN : 2180-1274
Publication Frequency: 6 Issues / Year

CSC PUBLISHERS
http://www.cscjournals.org

INTERNATIONAL JOURNAL OF DATA ENGINEERING (IJDE)

VOLUME 2, ISSUE 3, 2011

EDITED BY
DR. NABEEL TAHIR

ISSN (Online): 2180-1274

International Journal of Data Engineering is published both in traditional paper form and in Internet. This journal is published at the website http://www.cscjournals.org, maintained by Computer Science Journals (CSC Journals), Malaysia.

IJDE Journal is a part of CSC Publishers
Computer Science Journals
http://www.cscjournals.org

INTERNATIONAL JOURNAL OF DATA ENGINEERING (IJDE)

Book: Volume 2, Issue 3, August 2011
Publishing Date: 31-08-2011
ISSN (Online): 2180-1274

IJDE Journal is a part of CSC Publishers
http://www.cscjournals.org

Published in Malaysia

Typesetting: Camera-ready by author, data conversation by CSC Publishing Services – CSC Journals, Malaysia

CSC Publishers, 2011

EDITORIAL PREFACE

This is third issue of volume two of the International Journal of Data Engineering (IJDE). IJDE is an International refereed journal for publication of current research in Data Engineering technologies. IJDE publishes research papers dealing primarily with the technological aspects of Data Engineering in new and emerging technologies. Publications of IJDE are beneficial for researchers, academics, scholars, advanced students, practitioners, and those seeking an update on current experience, state of the art research theories and future prospects in relation to computer science in general but specific to computer security studies. Some important topics cover by IJDE is Annotation and Data Curation, Data Engineering, Data Mining and Knowledge Discovery, Query Processing in Databases and Semantic Web etc.

The initial efforts helped to shape the editorial policy and to sharpen the focus of the journal. Starting with volume 2, 2011, IJDE appears in more focused issues. Besides normal publications, IJDE intend to organized special issues on more focused topics. Each special issue will have a designated editor (editors) – either member of the editorial board or another recognized specialist in the respective field.

This journal publishes new dissertations and state of the art research to target its readership that not only includes researchers, industrialists and scientist but also advanced students and practitioners. The aim of IJDE is to publish research which is not only technically proficient, but contains innovation or information for our international readers. In order to position IJDE as one of the top International journal in Data Engineering, a group of highly valuable and senior International scholars are serving its Editorial Board who ensures that each issue must publish qualitative research articles from International research communities relevant to Data Engineering fields.

IJDE editors understand that how much it is important for authors and researchers to have their work published with a minimum delay after submission of their papers. They also strongly believe that the direct communication between the editors and authors are important for the welfare, quality and wellbeing of the Journal and its readers. Therefore, all activities from paper submission to paper publication are controlled through electronic systems that include electronic submission, editorial panel and review system that ensures rapid decision with least delays in the publication processes.

To build its international reputation, we are disseminating the publication information through Google Books, Google Scholar, Directory of Open Access Journals (DOAJ), Open J Gate, ScientificCommons, Docstoc and many more. Our International Editors are working on establishing ISI listing and a good impact factor for IJDE. We would like to remind you that the success of our journal depends directly on the number of quality articles submitted for review. Accordingly, we would like to request your participation by submitting quality manuscripts for review and encouraging your colleagues to submit quality manuscripts for review. One of the great benefits we can provide to our prospective authors is the mentoring nature of our review process. IJDE provides authors with high quality, helpful reviews that are shaped to assist authors in improving their manuscripts..

Editorial Board Members
International Journal of Data Engineering (IJDE)

EDITORIAL BOARD

Dr. Andrey Balmin
IBM Almaden Research Center
United States of America

Dr. Rishi R. Sinha
Microsoft Corporation
United States of America

Dr. Qiong Luo
Hong Kong University of Science and Technology
China

Dr. Thanaa M. Ghanem
University of St. Thomas
United States of America

Dr. Ravi Ramamurthy
Microsoft Research
United States of America

TABLE OF CONTENTS

Volume 2, Issue 3, August 2011

Pages

Comparison of Semantic and Syntactic Information Retrieval System on the Basis of Precision and Recall

Sanchika Gupta sanchigr8@gmail.com
Student/Computer Sc. & Engg.
Thapar University
Patiala, 147004, India

Dr. Deepak Garg dgarg@thapar.edu
Faculty/Computer Sc. & Engg./Asstt. professor
Thapar University
Patiala, 147004, India

Abstract

In this paper information retrieval system for local databases are discussed. The approach is to search the web both semantically and syntactically. The proposal handles the search queries related to the user who is interested in the focused results regarding a product with some specific characteristics. The objective of the work will be to find and retrieve the accurate information from the available information warehouse which contains related data having common keywords. This information retrieval system can eventually be used for accessing the internet also. Accuracy in information retrieval that is achieving both high precision and recall is difficult. So both semantic and syntactic search engine are compared for information retrieval using two parameters i.e. precision and recall.

Keywords: Information Retrieval, Precision, Recall, Semantic, Syntactic.

1. INTRODUCTION

Information Retrieval (IR) is the study of systems for searching, retrieving, clustering and classifying the data, particularly text or other unstructured forms. IR is finding material of an unstructured nature that satisfies an information need from within large storage usually from the computers [1]. IR is also used to facilitate semi structured search, clustering of documents based on their contents and classification of data. Before the retrieval process can even be initiated, it is necessary to define the text database which consists of related information from which information is to be retrieved. After the database is created a query is entered in the search space. A query is request for information from a database. These are formal statements for satisfying information needs. In information retrieval a query does not uniquely identify a single object in the collection. Instead, several objects may match the query, perhaps with different degrees of relevancy and accuracy. In general, a query is a form of questioning, in a line of inquiry. The style and format of querying might be different for both syntactic and semantic search engine.

A semantic information retrieval system attempts to make sense of search results based on context. It automatically identifies the concepts structuring the texts. For instance, if you search for “passport” a semantic information retrieval system might retrieve documents containing the words “visa”, “embassy” and “flights”. Semantic web help computers understand and interpret information and also finds additional information that might be useful. What this means is that the search engine through natural language processing will know whether you are looking for a small animal or a Chinese zodiac sign when you search for “rabbit”.
Every language has its own Syntax and Semantics. Syntax is the study of grammar. Semantics is the study of meaning. Syntax is how to say something. Semantic is the meaning behind what you

say. Different syntaxes may have the same semantic: x += y, x=x+y. Syntax and semantics are all about communication.

A web search engine or the syntactic information retrieval system is designed to search for information on the World Wide Web and FTP servers. The search results are presented in a list of results and are called hits. The information may consist of web pages, images, information and other types of files. Some search engines also mine data available in databases or open directories. Unlike web directories, which are maintained by human editors, search engines operate algorithmically or are a mixture of algorithmic and human input.

In this paper information retrieval system for local databases are discussed. The approach is to search the web both semantically and syntactically. The proposal handles the search queries related to the user who is interested in the focused results regarding a product with some specific characteristics. The objective of the work will be to find and retrieve the accurate information from the available information warehouse which contains related data having common keywords. This information retrieval system can eventually be used for accessing the internet also. Accuracy in information retrieval that is achieving both high precision and recall is difficult. So both semantic and syntactic search engine are compared for information retrieval using two parameters i.e. precision and recall.

For the syntactic information retrieval system, a local database is created which consists of related information and a simple search engine is developed to retrieve information from that database. For the semantic information retrieval system, ontology with same information as in the database is created and queries are used to extract information from that.

2. SEMANTIC INFORMATION RETRIEVAL SYSTEM

The Semantic Web proposes to help computers understand and use the Web. Metadata is added to Web pages that can make the existing syntactic web machine readable. The main purpose of the Semantic Web is driving the evolution of the current Web by allowing users to use it to its full potential, thus allowing them to find, share, and combine information more easily. This won't bestow artificial intelligence or make computers self-aware, but it will give machines tools to find, exchange and, to a limited extent, interpret information.

The Semantic Web combined with ontology can be used for visualization techniques in several different ways, but the visualization is dependent on characteristics of the ontology used. Ontology helps both people and machines communicate more effectively by providing a common definition of a domain [13]. The GUI serves as an interface between the user and the system. OWL (ontology web language) is the language used for developing ontologies. OWL Properties represent relationships. There are three types of properties-

- Object properties- Object properties depicts the relationships between two individuals.

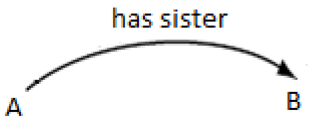


FIGURE 1: An object property linking the individual A to individual B.

- Datatype properties

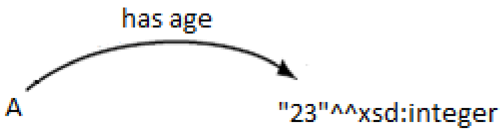


FIGURE 2: A datatype property linking the individual A to data literal ‘23’, which is a type of integer.

- Annotation properties- Annotation properties can be used to add information (metadata — data about data) to classes, individuals and object/datatype properties [4].

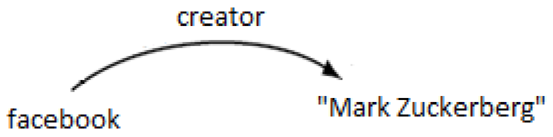


FIGURE 3: An annotation property, linking the class ‘facebook’ to the data literal (string) “Mark Zuckerberg”.

OWL	**DL Symbol**	**Manchester OWL Syntax Keyword**	**Example**
someValuesFrom	∃	some	hasChild some Man
allValuesFrom	∀	only	hasSibling only man
hasValue	∋	value	hasCountryOfOrigin value England
minCardinality	≥	min	hasChild min 3
cardinality	=	exactly	hasChild exactly 3
maxCardinality	≤	max	hasChild max 3

TABLE 1: Description logic symbols and the corresponding English language keywords [4].

The "Mediawiki Ontology" consists of information which is related. It helps to find information that is being searched and also provides the related information that might be helpful. Imagine this scenario. You want to purchase a car. You have heard about “jaguar” and want to know more about it, so you search for the term using your favourite search engine. Unfortunately, the results you’re presented with are hardly helpful. There are listings for jaguar the animal, a cat species etc. Only after sifting through multiple listings and reading through the linked pages are you able to find information about the Tata Group production “Jaguar”. On the other hand, the semantic information retrieval system can interpret and understand what is being searched for [15]. The semantic web agent helps you to find the required car and also tells you about its features, functions, price and other available options. FIGURE 4 presents the media wiki ontology graph which consists of related information.

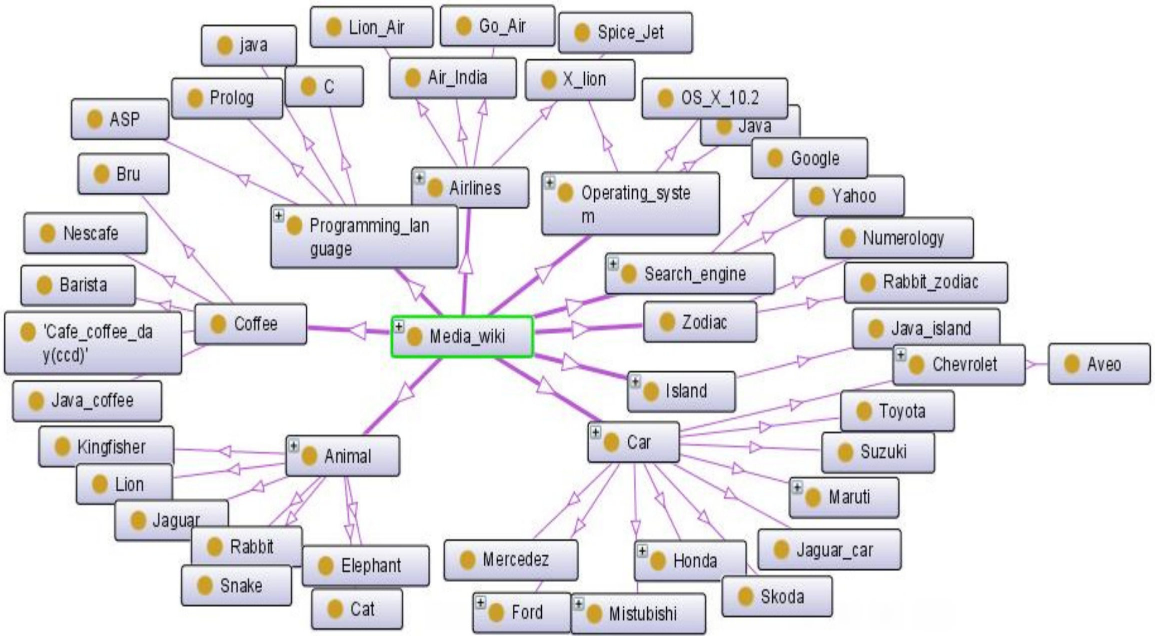


FIGURE 4: Mediawiki Ontology graph

3. SYNTACTIC INFORMATION RETRIEVAL SYSTEM

The term ”search engine” is used to indicate both crawler based search engines and manually maintained directories, although they gather their indexes in radically different ways. Here we are discussing the Crawler-based search engines, which are based upon the syntactic information retrieval system, such as Google which create their catalogues automatically: they crawl the web, then the users searches through what they have found. On the contrary, a manually maintained directory, such as the Open Directory, depends on humans: people submits a short description to the system about a certain site, or appropriate editors write a review for their assigned sites; thus, a web search looks for matches only in the submitted descriptions [12].

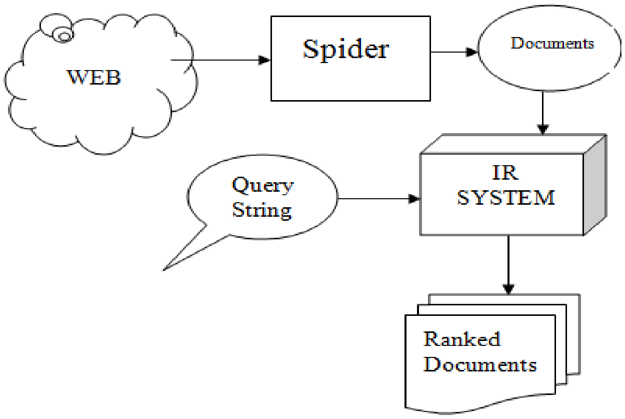


FIGURE 5: Syntactic Information Retrieval System.

FIGURE 5: depicts the syntactic information retrieval system. It consists of a spider which is a computer program that browses the web in a orderly fashion. It automatically discovers and collects resources, especially the web pages, from the Internet. This process is called spidering. Many search engine use spidering to provide up to date data. It provides a copy of all the documents which has already been visited for faster searches [14]. So when user inputs a query

string in the syntactic information retrieval system, the system then provides a list of ranked documents, ordered according to the requirement of the user to get high precision and recall.

In the syntactic information retrieval system a database is created "mediawiki" which consists of same information that is used to build the ontology, and a search engine is implemented using PHP to search from that database. This search engine retrieves every occurrence of the search item from the database. For ex. If we searched for "Lion", then every occurrence of Lion i.e. "Lion-the panther", "X Lion- Apple Mac operating system" and "Lion Air- Indonesia's largest private carrier airplane" is retrieved. Along with the search items links are also provided to get more information about them from the internet.

4. COMPARISON BASED ON FUNDAMENTAL SEARCH FACILITIES

This table gives the comparison of semantic and syntactic information retrieval system on the basis of various fundamental search facilities like symbol used, keywords used in the search queries, phrases, wildcards, prefixes etc.

Information retrieval system/ Properties	**Semantic Information Retrieval system**	**Syntactic Information Retrieval system**
Symbol	∃, ∋, ≥, =, ≤, ∀	+, -, ( )
Keywords	some, value, min, exactly, max, only	AND, OR, ANDNOT
Phrase	" ", [ ]	" "
Wildcards	*, ?, $	(*) whole word wildcard
Case sensitive	YES	NO
Prefixes	length, maxLength, minLength, totalDigits, fractionDigits	filetype, inurl

TABLE 2: Comparison of semantic and syntactic Information Retrieval system.

.

5. ESTIMATION OF PRECISION AND RECALL

To measure information retrieval effectiveness in the standard way, we need a test collection consisting of three things [1]:

1. A document collection i.e. a database from which the search is to be performed.
2. Information needs, expressible as queries.
3. A binary assessment of either relevant or non-relevant for each query-document pair.

To measure the effectiveness two parameters are defined: Precision and recall.

Precision (P) is the fraction of retrieved documents that are relevant

Precision = #(relevant items retrieved) / #(retrieved items)
= P(relevant|retrieved)
= P(sum/#)

Recall (R) is the fraction of relevant documents that are retrieved

Recall = #(relevant items retrieved) / #(relevant items)
= P(retrieved|relevant)
=P(num/#)

To measure the precision and recall, both the semantic and syntactic information retrieval systems are tested for 5 queries and based on the results which are retrieved, the estimation is made. The search- items (queries) on which estimation are done:

#1: Operating system
#2: Jaguar car
#3: Web Proxy

#4: Fly Kingfisher
#5: Rabbit Zodiac

Search Item	Syntactic Information retrieval system		Semantic Information retrieval system	
	P(sum/#)	P	P(sum/#)	P
#1	2.0/3.0	0.67	2.0/3.0	0.67
#2	1.0/4.0	0.25	1.0/1.0	1
#3	1.0/2.0	0.5	1.5/2.0	0.75
#4	1.0/3.0	0.34	1.0/5.0	0.2
#5	1.0/3.0	0.34	1.0/1.0	1
Mean P	N/A	0.42	N/A	0.72

TABLE 3: Estimation of Precision

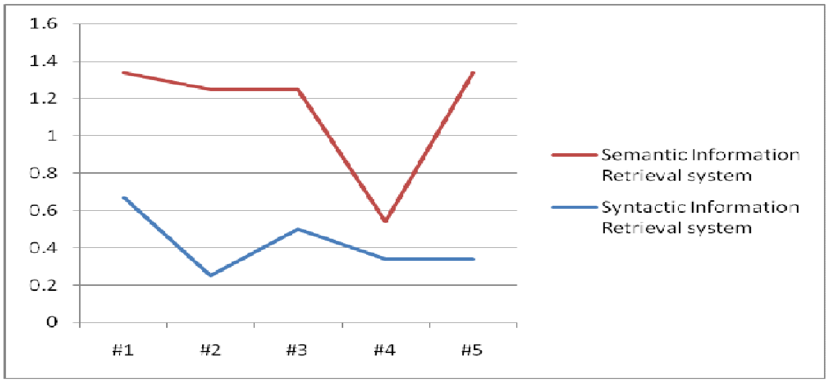


FIGURE 6: Comparison on the basis of precision

Mean precision

- Syntactic Information Retrieval system= 0.42
- Semantic Information Retrieval system= 0.72

Figure 6 gives the graphical representation of the above table. From Table 3, a graph is plotted which gives the comparison of two environments on the basis of precision. From the graph it can be inferred that the semantic information retrieval system has a higher precision for the same search items as compared to the syntactic information retrieval system.

Search Item	Syntactic Information retrieval system		Semantic Information retrieval system	
	P(num/#)	R	P(num/#)	R
#1	2.0/3.0	0.67	3.0/3.0	1
#2	1.0/2.0	0.5	1.0/1.0	1
#3	1.0/2.0	0.5	0/2.0	0
#4	1.0/2.0	0.5	1.0/2.0	0.2
#5	1.0/1.0	1	1.0/1.0	1
Mean R	N/A	0.634	N/A	0.64

Table 4: Estimation of Recall

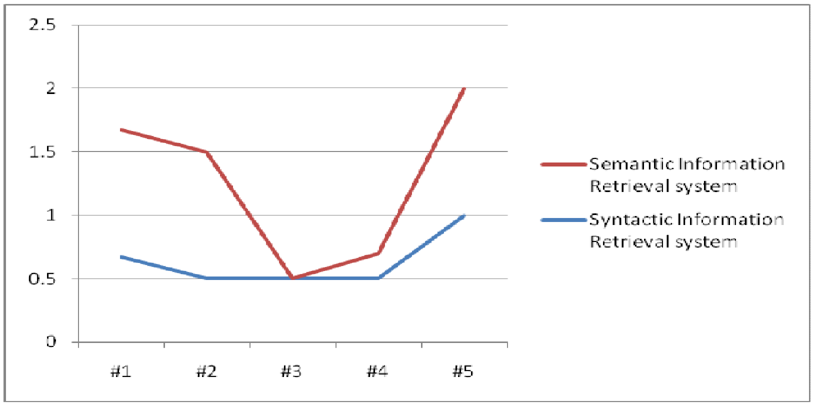


FIGURE 7: Comparison on the basis of recall

Mean recall

- Syntactic Information Retrieval system= 0.634
- Semantic Information Retrieval system= 0.64

Figure 7 gives the graphical representation of the above table. A graph is plotted between recall and the search items to give a comparison of the two search environments. As can be seen from the graph, semantic information retrieval system, shows a large diversity in the recall ratio for different search items i.e. for some search items the recall rate is very high and for others, it is nearly zero. Whereas for syntactic search retrieval system a constant recall rate can be seen. The graph shows a consistent rate and is not fluctuating.

Though the mean recall rate is approximately the same for both the semantic and syntactic retrieval systems, it can be inferred that syntactic retrieval system shows a more consistent recall rate as compared to semantic retrieval system, which has a fluctuating recall rate.

6. CONCLUSION

A competent Information Retrieval system must include the fundamental search facilities that users are familiar with, which include Boolean logic symbols, phrase searching, wild cards and use of prefixes. Because the searching capabilities of Information Retrieval system ultimately determine its performance, absence of these basic functions will severely handicap the search tool [8]. As we can see in Table 2, a comparison is done based on these search facilities. Both semantic and syntactic information retrieval system uses various search facilities but popularity of syntactic web is more as compared to semantic web as the former is widely used and accepted whereas the semantic web is new.

Retrieval performance is traditionally evaluated on two parameters: precision and recall. While the two variables can all be quantitatively measured, extra caution should be exercised when one judges the relevance of retrieved items and estimates the total number of documents relevant to a specific topic in the retrieval system [8]. FIGURE 6 gives the comparison of precision for both semantic and syntactic information retrieval system. Clearly it can be seen that semantic information retrieval system have mean precision of 0.72 which is much higher than that of syntactic information retrieval system which have a mean precision of 0.42 only. So it can be said that ratio of relevant items retrieved to total items retrieved for a search query is better in case of semantic information retrieval system.

Similarly FIGURE 7 gives the comparison on the basis of recall. As can be seen from the table the mean recall for both semantic and syntactic information retrieval system is almost the same. So the ratio of number of relevant items retrieved to the total number of relevant items present is almost same for both the retrieval system. Moreover from FIGURE 7, it can be inferred that the syntactic retrieval system has a more consistent recall rate as compared to syntactic search retrieval system which has a fluctuating recall rate.

7. REFERENCES

[1] C.D. Manning, P. Raghavan, H. Schütze. "*An Introduction to information retrieval*", Cambridge University Press Cambridge, England, Apr 1, 2009, pp. 26- 569.

[2] World Wide Web Consortium. "OWL Web Ontology Language Semantics and Abstract Syntax". W3C Recommendation 10 Feb, 2004.

[3] H. Knublauch, M. A. Musen, A. L. Rector. Medical Informatics Group, "Editing Description Logic Ontologies with the Protege OWL Plugin", Stanford University and University of Manchester, pp. 1- 9.

[4] M. Horridge, H. Knublauch, A. Rector, R. Stevens, C. Wroe. "A Practical Guide To Building OWL Ontologies Using The Prot´eg´e-OWL Plugin and CO-ODE Tools Edition 1.0", The University Of Manchester, 2004.

[5] World wide web consortium Internet: http://www.w3.org/2001, 2001.

[6] V. David, F. Miriam, C. Pablo. "An Ontology Based Information Retrieval Model" Universidad Autonoma de Madrid.

[7] J. Bar-Ilan. "On the overlap, the precision and estimated recall of search engines: A case study of the query "Erdos"". Scientometrics, 42 (2), 207-208, 1998.

[8] H. Chu, M. Rosenthal. (1996). "Search engines for the World Wide Web: a comparative study and evaluation methodology" Proceedings of the ASIS 1996 Annual Conference. [online] Available: http://www.asis.org/annual96/ElectronicProceedings/chu.html. October, 33. 127-35. Retrieved August 19, 2003.

[9] S. Clarke, P. Willett. "Estimating the recall performance of search engines". ASLIB Proceedings, 49 (7), pp. 184-189, 1997.

[10] W. Ding, G. Marchionini. "A comparative study of the Web search service performance". In: Proceedings of the ASIS 1996 Annual Conference, Oct 1996, pp.136-142.

[11] C. Oppenheiem, A. Moris, C. Mcknight, S. Lowley. "The evaluation of WWW search engines". Journal of documentation, 56 (2), pp.190-211, 2000.

[12] C. Cesarano, A. d'Acierno, A. Picariello. "An Intelligent Search Agent System for Semantic Information Retrieval on the Internet". WIDM'03, , New Orleans, Louisiana, USA. Nov 7–8, 2003.

[13] E. HyvÄonen, A. Styrman, S. Saarela. "Ontology-Based Image Retrieval", University of Helsinki, Department of Computer Science, pp.1-13.

[14] H. Sumiyoshi, I. Yamada, Y. Murasaki, Y.B. Kim, N. Yagi and M. Shibata, "Agent Search System for A New Interactive Education Broadcast Service", NHK STRL R&D No.84, Mar, 2004.

[15] Guo-Qiang Zhang, Adam D. Troy, and Keith Bourgoin. “Bootstrapping Ontology Learning for Information Retrieval Using Formal Concept Analysis and Information Anchors”, Department of Electrical Engineering and Computer Science Case Western Reserve University Cleveland, Ohio 44106, U.S.A, pp.1-14, 2008.

Optimized Access Strategies for a Distributed Database Design

Rajinder Singh tovirk@yahoo.com
Assoc. Professor
Faculty of Computer Science & Engineering
Guru Nanak Dev University, Amritsar-143001
Punjab, India.

Gurvinder Singh gsbawa71@yahoo.com
Assoc. Professor
Faculty of Computer Science & Engineering
Guru Nanak Dev University, Amritsar-143001
Punjab, India.

Varinder Pannu viki_virk@yahoo.com
Computer Engineer
Faculty of Computer Science & Engineering
Govt. Polytechnic, Amritsar-143001
Punjab, India.

Abstract

Distributed Database Query Optimization is achieved thru many complex sub operations on the Relations, Network Sites, Local Processing Facilities and the Database System itself. Many of these sub problems are NP-Hard itself, which makes Distributed Database Query Optimization a very complex and hard process. One of these NP Hard components is optimal allocation of various sub-queries to different sites of data distribution. Most of prevalent solutions take help of Exhaustive Enumeration Techniques, along with use of innovative heuristics. In this Paper we have proposed a stochastic model simulating a Distributed Database environment, and shown benefits of using innovative Genetic Algorithms (GA) for optimizing the sequence of sub-query operations allocation over the Network Sites. Also, the effect of varying Genetic Parameters on Solution's quality is analyzed.

Keywords: Distributed Query Optimization, Database Statistics, Query Execution Plan, Genetic Algorithms, Operation Allocation.

1. INTRODUCTION

Query Optimization process involves finding a near optimal query execution plan which represents the overall execution strategy for the query. The efficiency of distributed database system is significantly dependent on the extent of optimality of this execution plan. According to Ozsu and Valduriez[1],this process of generating a good query execution strategy involves three phases. First is to find a *search space* which is a set of alternative execution plans for query. Second is to build a *cost model* which can compare costs of different execution plans. Finally in third step we explore a *search strategy* to find the best possible execution plan using cost model.

Before putting any queries to a Distributed Database, one needs to design it according to the needs of an organization. Analysts have to plan data/Fragment allocation according to the nature and frequencies of the various queries at different sites. Different Sequence of operations or sub-operations needed to generate results of a query is very large e.g. as near to n! for relational join of n relations. Decisions have to be taken dynamically for allocation of intermediate relation fragments generated during the query. A Transaction Profile is build to provide this information for various queries to the database. This paper gives a Genetic Algorithm for this step of Execution Plan. It assumes that a transaction profile provides the necessary details of fragment allocation and cost profiles for various pair of sites. It gives a cost model to predict cost of various allocation

plans for intermediate relations and sub operations generated during process of Query. Finally this GA finds the best possible execution strategy in terms of finding sequence and sites for various sub operations, called Operation Allocation Problem[2].
A GA(Genetic Algorithm) has several advantages over other approaches, as it has been successfully applied to a vast set of real world applications, not only that it gives a robust solution but also a good set of alternative possible solutions to choose from [3] and is inherently parallel to achieve good response time.

2. PREVIOUS RESEARCH WORK

Distributed database systems design and query optimization has been and will remain an active area of research for a lot times to come, due to complex and intractable nature of the problem[4,5,6,7,8,9,10].Most of the work has concentrated on two aspects: Data Allocation(The plan of allocating Fragments to various sites) and Operation Allocation(How to generate a sequence of subqueries on various sites). Apers and P.M have discussed in detail the data allocation problem and their fragmentation in [11]. An integrated solution to problems of Data Fragmentation, allocation, replication in Distributed Databases, has been proposed in Tamhankar & Ram[12]. Zehai Zhou[10] propose using heuristics and genetic algorithms for large scale database query optimization.The NP Hard problem is reduced to a join ordering problem similar to a variant of a Travelling Salesman Problem.Several heuristics and a GA is proposed for solving the join order problem.

Simulation experiments for comparison of Branch & Bound, Simulated Annealing, Greedy approaches for operation allocation problem have been describes in detail by Martin & Lam[13].
Frieder and Baru [14] propose dynamic site selection strategies for distributed database design on a microcomputer. March & Rho in [2] have proposed an excellent cost model for reducing local I/O costs, CPU Costs and Communication Costs in operation allocation strategy. Johansonn & Noumann in [15] extended their work bu considering parallel processing and Load Balancing in Data and Operation Allocation.

3. Objective Function & Cost Model

Database Statistics

The main factor affecting the performance of an execution strategy is the size of the intermediate fragments produced during the execution of the sub operations of the query. As many intermediate relations or fragments will need to move over various sites, we need to estimate the size of them to determine transmission costs. This estimation is based on statistical information about base relation and formulas to predict the cardinalities of the results of operations [1].

The set of operations (sub-queries) generated in response to a query can be represented by an operator tree. Nodes of operator tree represent various operations and lines represent cost (based on size of fragment) of operation sequence. A site's Local CPU and I/O costs are proportional to the size in bytes (blocks) of data processed and communication costs depend on communication coefficients between a pair of sites and bytes of blocks moved.

The main assumptions are that Transaction Profiles are known a-priori, providing the details of frequencies of transaction at various sites, base relation sizes and allocation plan at various sites, communication coefficients giving cost of communication amongst various pair of sites and local I/O and CPU coefficients. Also query execution order is given, we emphasize on finding sub-query allocation and cost associated to it with respect to allocation to various sites.
Projection, Selection and Joins account for most of the sub-queries in a Database Query and for simplicity purposes, only these operations have been considered.

3.1 Objective Function Formulation

We start by simulating a design of distributed database by taking a set ‘S’ of data distribution sites. Set ‘R’ of relations/fragments stored on those sites. A Set ‘Q’ as set of transactions.
Let a query transaction ***‘q’*** for retrieval, be broken into a set of ***‘j’*** sub queries on the ‘R’ set of relations.

3.1.1 Decision Variables :-

(i) Data Allocation Variable A_{rs}

A_{rs} = 1 *(if site ‘S’ holds copy of relation/fragment ‘r’)*

A_{rs} = 0 *(otherwise i.e. fragment ‘r’ copy is not available at set S)*

(ii) Variables for site selection for sub query execution:

S_{ys}^{q} **:** *(Represents sequence of sub query execution at various sites in the life time of query)*

$S_{ys}^{q} = 1$ *(if subquery ‘y’ of Query ‘q’ is done at site s)*

$S_{ys}^{q} = 0$ *(otherwise)*

(iii) For Join operations a notation is proposed to handle left previous operation operation of a join operation (LPO) & right previous operation of a join(RPO) as following:

$S_{yv[p]S}$ = 1 *(for [p] = 1 for left previous operation of a Join)*

$S_{yv[p]S}$ = 1 *(for [p] = 2 for right previous operation of a Join)*

$S_{yv[p]S}$ = 0 *otherwise*

(iv) f^{q}_{ry} represents the query tree in such a way that sub query ‘*y*’ of query ‘*q*’ references the intermediate relation/fragment *r*.

$f^{q}_{ry} = 1$ *(if the base relation ‘r’ or intermediate fragment ‘r’ is used by sub query ‘y’ of ‘q’ query)*

$f^{q}_{ry} = 0$ *otherwise*

(v) For use of intermediate Relations by Join Operation

$f^{q}_{ryv[p]} = 1$ *(for lpo of join ‘y’)*

$f^{q}_{ryv[p]} = 1$ *(for rpo of join ‘y’)*

$f^{q}_{ryv[p]} = 0$ *otherwise*

By making use of above decision variables operation allocation problem formulation is represented as,

Given a input data file highlighting data allocation scheme matrix, given by S^{q}_{ys} Data Allocation Scheme Matrix: A base relation ***y*** stored at site ***s***.
i.e.

Given a Transaction Profile,a Data Allocation Sceme represented by variable $\boldsymbol{A_{rs}}$

And given $\boldsymbol{f^q_{ry}}$ *intermediate relations/fragments is used by sub query **y** of query **q***

We have an objective Function to calculate as to find $\boldsymbol{S^q_{ys}}$

3.1.2) Cost model

Given a set of fragments
R = $\{r_1,r_2,\ldots,r_n\}$

& a network of sites.
S = $\{s_1,s_2,\ldots,s_m\}$

& a set of sub queries
Q = $\{q_1,q_2,\ldots,q_q\}$

Sub Query Allocation problem involves finding the “optimal” possible distribution of R to S.
Ozsu gives a model for Total cost as Total Cost Function having two components: query processing and storage cost as

$$TOC = \sum QPC_i + \sum_{\forall s \text{€} S} \sum_{\forall fj \text{€} F} STC_{jk}$$

Where QPC_i is query processing cost of application q_i and STC_{jk} is the cost of storing fragment F_j at site S_k.

We choose Ozsu’s model of query cost as function of sum of local processing costs and transmission costs .We simplify it further by ignoring update costs and ignoring concurrency control costs as we are giving model for retrieval transactions(queries) only. Further concurrent retrievals don’t impose any more integrity control costs.
Ozsu’s formulation gives

$QPC_i = PC_i + TC_i$ *(PC: Processing Cost, TC :Transmission Cost)*

&

$PC_i = AC_i + IE_i + CC_i$ *(AC: Access Costs, IE : Integrity Enforcement Costs, CC : Concurrent Update control costs)*

In our model we discard the sum of two costs components ($IE_i + CC_i$), because as discussed in Para above, we present a simple model of retrieval queries only.

Therefore Access Costs may be represented as

$$AC_i = \sum_{\forall s \text{€} S} \sum_{\forall fj \text{€} F} (u_{ij} * UR_{ij} + r_{ij} * RR_{ij}) * x_{jk} * LPC_k$$

The summation gives total number of accesses for all the fragments referenced by $\boldsymbol{q_i}$. Now $\boldsymbol{x_{jk}}$ selects only those cost volume entries for sites where fragments are stored actually.

3.1.3) Local Processing Costs

For Simple selection & projections

$$LOPC^{q}_{y} = \sum_{s} S^{q}_{ys}(I\ PO_{s} \sum_{r} I^{q}_{ry}M^{q}_{ry} + CPC_{s} \sum_{r} I^{q}_{ry}M^{q}_{ry}) \qquad (1)$$

Where M^{q}_{ry} = No. of memory blocks of relations 'r' accessed by sub-query y of q.

IPO_{s} = *Input Output Cost Coefficient of site s in msec per 8k bytes*

CPC_{s} = *CPU Cost coefficient of site s.*

So equation (1) represents local processing costs of transforming input relation from disk to memory and CPU time for processing a Selection or Projection at sites ***s***.

Ozsu's model ignores join's local processing cost details. For that we have extended this model to add local join costs details as following.

Local processing costs for a join

$$LOPC^{q}_{y} = \sum_{s}S^{q}_{ys}IPO_{s}\sum_{p}\sum_{r}p_{v}I^{q}_{ryv[p]}M_{q} \qquad 2(a)$$

$$+$$

$$\sum_{s}S^{q}_{ys}(IPO_{t}\prod_{r}I^{q}_{ry}M^{q}_{ry} + CPC_{s}\prod_{r}I^{q}_{ry}M^{q}_{ry}) \qquad 2(b)$$

Where p_{v} is Selectivity Factor & is referred as the ratio of possible different values of a field to the domain of that field.($0 <= p_{v} <= 1$)

$M_{ryv[p]}$ is the size of intermediate relation

where v[p] represents p=1 for left previous operation of a join &
p=2 for right previous operation of a join.

Equation 2(a) represents

Input Output costs in storing intermediate results of previous operations to the site of current join operation.

Equation 2(b) represents

CPU & I/O costs for performing current join operations at site 's'.

3.1.4) Communication Costs:

An involved in case of join operations only as we have assured that selections & projections of retrievals an to be done only at sites which hold a copy of that base relations. Join may be performed at any of possible sites.

$$COMM^{q}_{y} = \sum_{p}\sum_{s}\sum_{t} S^{q}_{yv[m]s} * S^{q}_{yt}C_{st} (\sum_{r}^{q} I_{ryv[p]}\ M^{q}_{ryv[p]})$$

Where

C_{st} *(is the communication cost coefficient taken from input data matrix)*

$C_{st} = 0$ if $(s = t)$ *(i.e. previous operations and join operation on same site)*

If final operation is not done at the query destination site then a Communication component is added separately for sending the final query result that site.

4. THE GENETIC ALGORITHM (GA_OA)

GA_OA (Genetic Algorithm for Operation Allocation) starts by generating an initial pool of solutions by random generation of operation sequence at given number of sites. An improvement in this previous prevalent approach in [9][10] has been done in this GA is to use transaction profile statistics to generate it.

Each chromosome is evaluated according to objective function and assigned a Fitness value accordingly. Next populations are generated using principles of GA as in [3] i.e. applying SELECTION,CROSSOVER & MUTATION, The fitter a member is more chance it gets to enter the mating pool to generate next population.

Crossover is used so that off-spring shares features of both parents and possibly improves over them. Mutation operator is applied with very small probability like.02 ,so as some important features of parent population of chromosomes are not lost .Elitism is also applied, which ensure that best chromosome of previous population enters next population by 1oo percent probability.

A sequence of integers like 2 1 3 3 4 1 is used to represent a chromosome such that it represents the sub-query allocation plan in the way that, that sub-query 1 is done at site 2, sub-query 2 is done at site1, sub-query 3 is done at site3, sub-query 4 is done at site 3, sub-query 5 is done at site 4, sub-query 6 is done at site 1.

A Structured English representation of GA_OA is outlined below:

1. *Generate an initial population pool of chromosomes, based on half the members of the pool generated from transaction profiles, data allocation profiles and others selecting randomly over no. of sites .*

2. *Evaluate the fitness of each member based on objective function to reduce the total cost of a query.*

3. *Based on Stochastic remainder method select and give more chance to fitter members to enter a mating pool according to probability proportional to their fitness value.*

4. *To enable Elitism, enter the most fit member of previous generation in mating pool, by replacing it with least fit.*

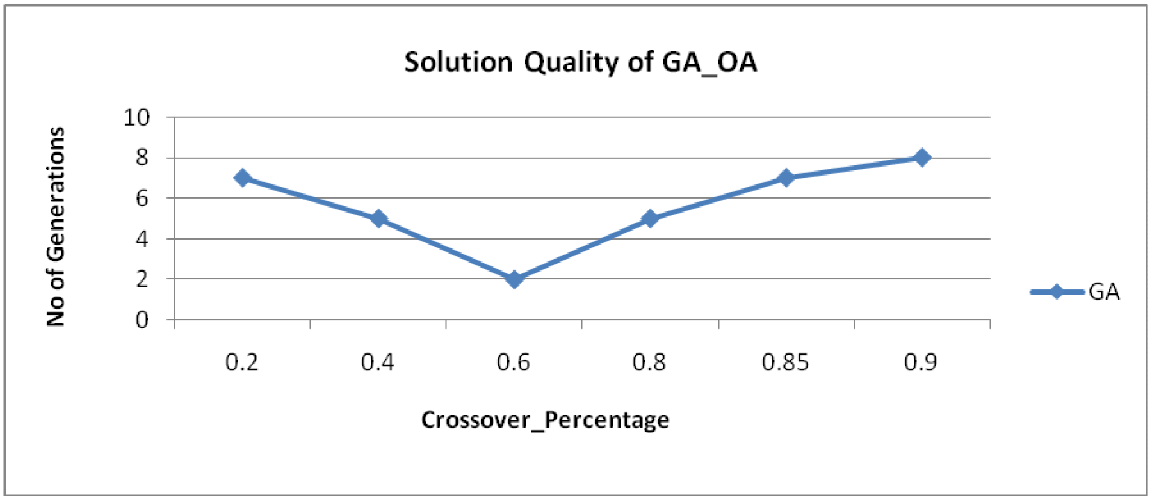
5. *Apply crossover (probability=0.7) and mutation (0.2) to generate a new population pool.*

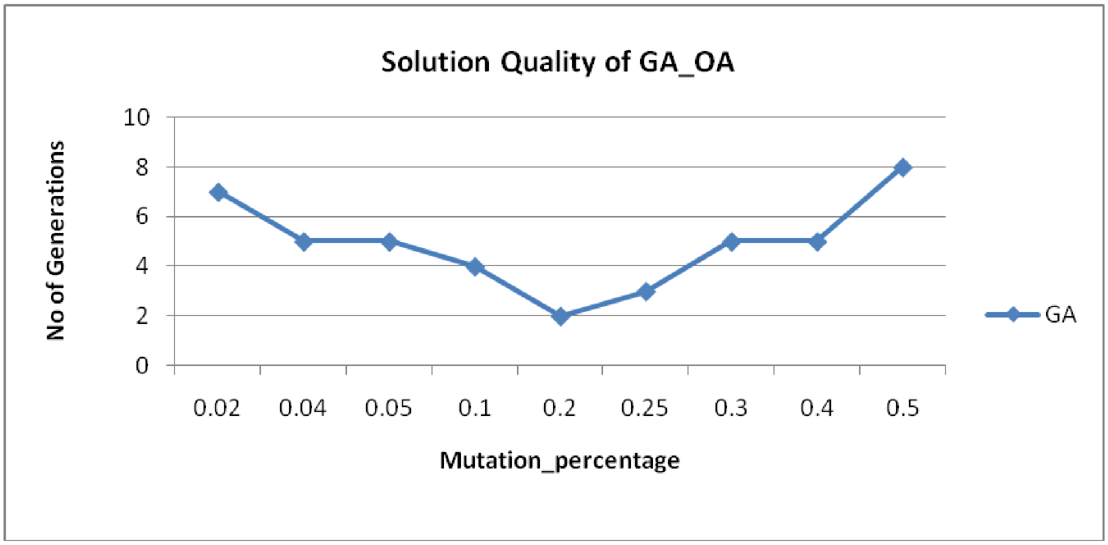
6. *Repeat steps 2 to 5 until maximum no of generations are generated.*

5. EXPERIMENTS & RESULTS

Experiments were conducted after coding the GA_OA simulator on an Intel® core™ 2 6420 @ 2.13 GHz machine with 1.00 GB RAM on WINDOWS-XP platform. Exhaustive Enumeration simulator program was developed with varying all possible permutations of sub-query allocation sequence. It was observed that Exhaustive Enumeration run time rises exponentially as compared to GA when we increase no. of joins or no. of sites. When no. of joins are increased from 4 to 10 Exhaustive_ Enumeration chokes very quickly but GA Run Time rises slowly. In case of increasing the number of sites ,Run Time for GA increases linearly whereas Exhaustive_ Enumeration rises exponentially and very quickly becomes almost intractable.

The performance of simulator varied as we vary genetic operators Crossover and Mutation as highlighted by the subsequent graphs, A crossover value of 0.6 was found giving optimal results and mutation parameter of 0.2 achieved optimal solution in least number of generations.





6. CONSLUSION & FUTURE WORK

The aim of this research paper is limited to proposing a stochastic solution to the operation allocation problem of Distributed Database Design. Most of the commercial vendors of Distributed DBMS to date use exhaustive enumeration procedures along with different heuristics. They also incorporate solutions based on other algorithm design techniques like Dynamic Programming, Backtracking etc. This paper highlights that exhaustive procedures quickly go intractable when No. of sites, or, No. of joins are increased suddenly. Exhaustive Enumerations along with heuristics guarantee an optimal solution but total time of query is too large to be practically viable.GA_OA does not guarantee the most optimal solution but provides very near to the best solution in a very short span of time.

In future efforts should be done to incorporate Genetic Based Solutions to allocation problems of Distributed Database. More work needed to be done to ensure that an optimal solution is guaranteed in most of situations by GA's. Furthermore Fragmentation, operation allocation and Data Allocation and Load Balancing need to be integrated in one robust Genetic Solution.

2. REFERENCES

[1] Ozsu & Valduriez. *"Principles of Distributed Database Systems" Pearson Education 2nd Edition,pp. 228-298.*

[2] March,Rho,"Characterisation and Analysis of a Nested Genetic Algorithm for Distributed Database Design".,Seoul Journal of Business pp 85-121 vol2,Number 1. 1995.

[3] Goldberg David.E "Genetic Algorithms in search, Optimization & Learning" *Pearson Education 2nd Edition,pp. 1-55.*

[4] Sacco,G. & Yao"Query Optimisation in Distributed Database Systems"1982,Advances in Computers,21,225-53.

[5] Yu,C.T,Chang" Distributed Query Processing " ACM Computing Surveys,16,399-433.

[6] Graefe,G"Query Evalution Texhniques for a large Database" ACM Computing Surveys,25,73-90, .1993.

[7] March,S.T.,Rho "Allocating Data and Operations to nodes in a distributed database design". IEEE Trans. On knowledge and Data Engg.,7(2). 1995.

[8] Kossman,D. "The state of the art in Distributed Query Processing".,ACM Computing Surveys.,32(4),422-469. 2000.

[9] Cheng,C.H.Lee,W-K,Wong,K-F, "A Genetic Agorithm based clustering approach for database partitioning " IEEE Transactions on System,Man,Cybernetics,32(3),215-230. 2002.

[10] Zehai Zhou,"Using Heuristics and Genetic Algorithms for Large Scale Database Query Optimization," Journal of Information and Computing Sciences,Acadeamim Press-2007.

[11] Apers,P.M.G,1988"Data Allocation in Distributed Database Systems",ACM Trans. On Database Syatems,.13(3),263-304

[12] Tamhankar,A.M & Ram"Database Fragmentation & Allocation: An Integrated Methodolgy and case study." IEEE Transactions on System,Man,Cybernetics,28(3),288-305.

[13] Martin,T,Lam& Russel"An Evaluation of Site Selection Algorithms for Distributed Query Processing"'The Compuer Journal,33(1),61-70,1990.

[14] Frieder, O. Baru"Site and Quey Sechduling policies in Microcomputer Database Systems" IEEE Trans. On knowledge and Data Engg.,6(4).1994.

[15] Johansson,JM,March,ST,Naumann"Modelling Network Latency Paralell Processing In Distributed Database design",Decision Sciences,34(4) 677-706 2003.

A Novel preprocessing Algorithm for Frequent Pattern Mining in Multidatasets

Dr.K.Duraiswamy *kduraiswamy@yahoo.co.in*
K.S.Rangasamy College of Terchnology,
Tiruchengode -637 209, Tamilnadu, India

B.Jayanthi (Corresponding Author) *sjaihere@gmail.com*
P.G.Department of Computer Science,
Kongu Arts and Science College,
Erode – 638 107, Tamilnadu, India

Abstract

In many database applications, information stored in a database has a built-in hierarchy consisting of multiple levels of concepts. In such a database users may want to find out association rules among items only at the same levels. This task is called multiple-level association rule mining. However, mining frequent patterns at multiple levels may lead to the discovery of more specific and concrete knowledge from data. Initial step to find frequent pattern is to preprocess the multidataset to find the large 1 frequent pattern for all levels. In this research paper, we introduce a new algorithm, called CCB-tree i.e., Category-Content-Brand tree is developed to mine Large 1 Frequent pattern for all levels of abstraction. The proposed algorithm is a tree based structure and it first constructs the tree in CCB order for entire database and second, it searches for frequent pattern in CCB order. This method is using concept of reduced support and it reduces the time complexity.

Keywords: Frequent Patterns, Multiple-level, Association Rule, CCB-tree, Minimum Support.

1. INTRODUCTION

Association rule mining is an important research subject put forward by Agrawal in reference [1]. Association Rule mining techniques can be used to discover unknown or hidden correlation between items found in the database of transactions. The problem of mining association rule could be decomposed into two sub problems, the mining of frequent itemsets/Patterns and the generation of association rules. [1][3].Finding frequent itemsets becomes the main work of mining association rules [2] many applications at mining associations require that mining be performed at multiple levels of abstraction [6].For example; a transaction in the database consists of a set of items. An example of such an association rule might be “80% of customers who buy itemset X also buy itemset Y”. The support count of an itemset is the number of transactions containing an itemset and support of an itemset is the fraction of those transactions besides, finding 80 percent of customers that purchase milk may also buy purchase bread, it is interesting to allow users to drill-down and show that 75 percent of people buy wheat bread if they buy 2 percent milk [10]. The association relationship in the latter statement is expressed at a lower level of abstraction but carries more specific and concrete information than in the former. Therefore a data mining should provide efficient methods for mining multiple-level association rules. To explore multiple-level association rule mining, one needs to provide: 1) data at multiple levels of abstraction, and 2) efficient methods for multiple-level rule mining. In many applications, taxonomy information is either stored implicitly in the database. Therefore, in this study, we generate category-content-brand tree i.e., CCB-tree to find frequent pattern at all levels of abstraction. The proposed algorithm has the following advantages. 1) It generates a frequent pattern at all levels. 2) If follows Top-down deepening Search method. So that searching time is reduced for lower level tree if ancestors are not at minimum support count. It also reduces the execution time.

The rest of the paper is organized as follows. Section gives the basic concept related to multiple level association rules. Section 3 gives the view of the related works. Section4 gives the statement of problem. Section presents the Apriori Algorithm Section6 presents the frequent pattern generation algorithm. Section7 gives the example of the proposed algorithm. Section8 shows the experimental results of the performance of the algorithm. Section9 Concluding remarks of the proposed research work.

2. MULTIPLE-LEVEL ASSOCIATION RULES

We assume that the database contain 1) an item dataset which contain the description of each item in **I** in the form of (A_i, description), where A_i € **I** and 2) a transaction dataset, T, which consist of a set of transaction (T_i { A_p,…. A_q,}), where T_i is a transaction identifier and A_i € **I** for (for **I** = p….q).

To find relatively frequent occurring patterns and reasonably strong rule implications, a user or an expert may specify two thresholds: minimum support, σ' and minimum confidence, φ. For finding multiple-level association rule, different minimum support and/or minimum confidence can be specified at different levels.

Definition 1: The support of an item A in a set S, $\sigma(A/S)$, is the number of transactions(in S) which contain A versus the total number of Transactions in S.

Definition 2: The confidence of A→B in S, $\varphi(A \rightarrow B/S)$, is the ratio of $\sigma(AUB/S)$ versus $\sigma(A/S)$, i.e., the probability that item B occurs in S when item A occurs in S.

The definition implies a filtering process which confines the pattern to be examined at lower level to be only those with large support at their corresponding high level. Based on this definition, the idea of mining multiple- level association rules is illustrated below.

TABLE1: A sales transaction table

transaction_id	Bar_code_set
351428	{17325, 92108, 55349…}
982510	{92458, 77451, 60395…}
----	----

Example 1: Let the query to be to find multiple-level association rule in the database in Table 1 for the purchase patterns related to Category, Content and Brand of the food which can only be stored for less than three weeks.

TABLE 2: A sales_item (description) relation

Bar_code	Category	Brand	Content	Size	Storage_pd	price
17325	Milk	Foremost	2%	1(ga)	14(days)	$3.89
----	----	----	---	----	----	----

TABLE 3 : A generalized sales_item description table

GID	Bar_Code_Set	Category	Content	Brand
112	{17325, 31414, 91265}	Milk	2%	Foremost
----	----	----	---	----

The relevant part of the sales item description relation in Table 2 is fetched and generalized into a generalized Sales_item description table, as shown in Table 3, in which is tuple represent a generalized item which is the merge of a group of a tuples which share the same values in the interested attributes. For example, the tuple with the same category, content and brand in Table 2 are merged into one, with their bar codes replace by a bar-code set. Each group is then treated as an atomic item in the generation of lowest level association rules. For example, the association rule generated regarding to milk will be only in relevance to (at the low concept levels) brand (such as Dairyland) and Content (such as 2%) but not to size, producer, etc.

The taxonomy information is provided in table 3. Let Category (such as “milk”) represent the first-level concept, content (such as “2%”) for the second level one and brand (such as “Foremost”) for the third level one. The table implies a concept tree like Fig.1.

The process of mining Multiple-level association rules is actually will be starting from top-most concept level. Let the minimum support at this level be 5% and the minimum confidence is 50%. One may fine the Large 1-itemset: “bread (25%), meat (10%), and milk (20%), Vegetable (30%).

At the second level, only the transactions which contain the large items at the first level are examined. Let the minimum support at this level be 2% and the minimum confidence is 40%. One may find frequent 1-itemsets: “lettuce (10%), Wheat bread (15%), white bread (10%, 2% milk (10%)...”The process repeats at even lower concept level until no large patterns can be found.

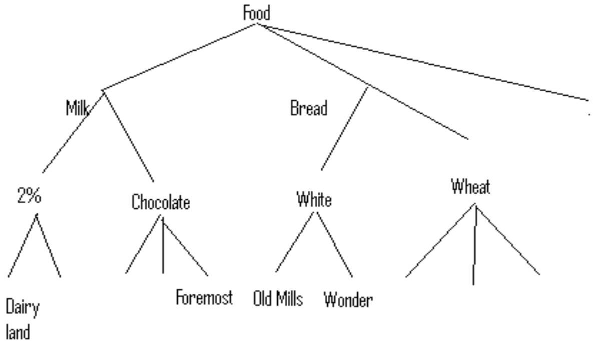


FIGURE 1: taxonomy for the relevant data items.

2. RELATED WORK

Since it was introduced in [1](R.Agrawal,T.Imielinski and A.N.Swami,1993). The problem of frequent itemset mining has been studied extensively by many researchers. As a result, a large number of algorithms have been developed in order to efficiently solve the problem [2][3](R.Agrawal, R.Srikant, 1994, J.Han, J.Pel, Y.Yin, 2000).In practice; the number of works has been focused on mining association rules at single concept level. Thus there has been recent interest in discovering Multiple Level Association rule. A new approach to Find Frequent pattern for multi-level datasets has to be considered. Work has been done in adopting approaches originally made for single level datasets into techniques usable on multi-level datasets. The paper in [4] Han & Fu (1995) shows one of the earliest approaches proposed to find frequent itemsets in multi-level datasets and later revisited in [5] Han & Fu (1999). This work primarily focused on finding frequent itemsets at each level in the dataset. The paper in [11] (Thakur, Jain & Pardasani 2006) proposed to find cross-level frequent itemsets. The paper in (8) (Pratima Gautham & K.R. Pardasani 2010) proposed efficient version of Apriori approach to find large 1 frequent pattern. The paper in [9] (Popescu, Daniela.E, Mirela Pater 2008) proposed AFOPT algorithm. The paper in [12] (Yinbo Wan, Yong Liang, Liya Ding 2009) proposed a novel method to extract multilevel rules based on different hierarchical levels by organizing and extracting frequent itemsets mined from primitive data items. The paper in [7](Mohamed Salah Gouider, Amine Farhat 2010) proposed a technique for modeling and interpretation of constraints in a context of use of concept hierarchies. However, even with all this work the focus has been on finding the large 1 frequent pattern using Apriori algorithm method. This work attempts to find the Large 1 frequent pattern for all levels with new approach i.e., CCB-tree using reduced support.

3. PROBLEM STATEMENT

The problem of mining multiple-level association rules was introduced in [4](Han & Fu (1995)), [5]Han & Fu(1999), [11](Thakur, Jain & Pardasani 2006), [8](Pratima Gautham & K.R. Pardasani 2010), [9] (Popescu, Daniela.E, Mirela Pater 2008), [12] (Yinbo Wan, Yong Liang, Liya Ding 2009), [7](Mohamed Salah Gouider, Amine Farhat 2010). There are two steps in association rule mining. First step is to find Large 1 frequent patterns for all level and then Large2...LargeK frequent pattern and Second step is to generate Association rules. We focus on first step i.e., finding large 1 Frequent Patterns at all levels. The objective of this work is to construct category-content-Brand tree (CCB-tree) in depth first order and it search for the large 1 frequent pattern in the same order so that it reduces the searching time. In this work, an algorithm CCB-tree is proposed, to find the frequent patterns for different levels. More specifically, given a transaction database TD, a different minimum Support for each level.

4. PROPOSED ALGORITHM

```
Algorithm CCB-tree construction and mining:
Input:
1. Transaction Database TD, minimum support (min_sup) for all levels
Output:
  Large 1 Frequent pattern for all levels.
Steps:
1. Create the root of the CCB-tree T with label “Null”
2. For each transaction Trans in TD do the following
3. Select items in Trans
4. Let item list in Trans be [p/P], where p is the first element and each element has a
   dimension d and P is the remaining list
5. Call Insertion ([p/P], T)
6. Call mining(T)
7. End for
8. Function Insertion ([p/P],T)
9. //Search a tree T for Key Value $P^1$,.. $P^d$. It is assumed that branching is determined by
     the dimension d of the key value//
10. For i = 1 to d by 1 do
11. If T has a child $N^i$ such that $N^i$.itemName = $p^i$.itemName
12. Then $N^i$.Count = $N^i$.Count + 1 and Trans_id = TID
13. Else
14. If i <d Create a new node with 3 fields i.e., item.name, Count, Trans_id
15. Then $N^i$.itemName = $p^i$.itemName , $N^i$.Count = $N^i$.Count + 1 and Trans_id =
    TID
16. Else Create a new node with 2 fields i.e., item.name, Count
17. Then $N^i$.itemName = $p^i$.itemName , $N^i$.Count = $N^i$.Count + 1
18. End If
19. Increment i and perform steps from 9 to 16.
20. End For.
21. Function mining (T)
22. Put the initial node in T on a list search
23. If initial node. count>=min_sup print its item.name, count and
24. Move towards its descendents i.e., next level by level of the same parent and
25. Print its item.name, count
26. Else move to the successors of initial node
27. End If
28. End For
```

5. EXAMPLE

This Section shows the example to demonstrate the proposed algorithm to mine Large 1 frequent pattern in multidatasets, which uses a hierarchy information encoded transaction table [5]. This based on the following consideration, first a data mining is usually in relevance to only a portion of the transaction database, such as food instead of all the items. It is beneficial to collect the

relevant set of data and then work repeatedly on the task-relevant set. Second, encoding can be performed during the collection of task-relevant data and thus there is no extra “encoding pass” required. Third, an encoding string, which represents a position in a hierarchy, required fewer bits than the corresponding object identifier or bar-code.
An abstract example, which simulates the real life example of Example 1, is analyzed as follows:

Example 2: The taxonomy information for each (grouped) item in Example 1 is encoded as a sequence of digits in the transaction table4. For example, the item ‘2% Foremost milk’ is encoded as ‘112’ in which digit, ‘1’ represents ‘milk’ at level-1, the second, ‘1’, for ‘2%(milk)’ at level-2 and the third,’2’, for the brand ‘Foremost’ at level-3. Similar to Agrawal and Srikant [2], repeated items at any level will be treated as one item in one transaction.The derivation of large 1 itemsets at all levels proceed as follows.

TABLE4: Sample Data

TID	Items
T1	{111, 121, 211, 211}
T2	{111, 211, 222, 323}
T3	{112, 122, 221, 411}
T4	{111, 121}
T5	{111, 122, 211, 221, 413}
T6	{113, 323, 524}
T7	{131, 231}
T8	{323, 411, 524, 713}

CCB-Tree Construction:
Let T1 = {111, 121, 211, 211} and p be a data with 3 dimensions, i.e., 1-category, 2-content and 3-Brand.Consider level 1(dimension 1 of first item) search a tree for key value. It is assured that level is determined by the dimensions d of p. If key values are not in tree, create a node with item.name, count and transaction id.

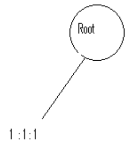


FIGURE 1: First level 1: item.name 1 : count and 1: trans_id

Consider level 2 (dimension 2 of first item) searches a tree for key value. If key values are not in tree, create a node with item.name, count and transaction id.

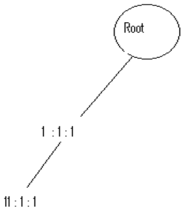


FIGURE 2: Second level

Consider level 3 (dimension 3 of first item) searches a tree for key value. If key values are not in tree, create a node with item.name, count.

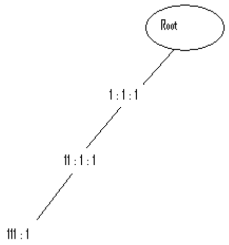


FIGURE 3: Third level

After T1 is over the appearance of CCB- Tree:

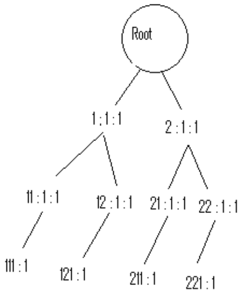


FIGURE 4: CCB-tree for T1

After the complete construction of CCB-Tree for the Table4:

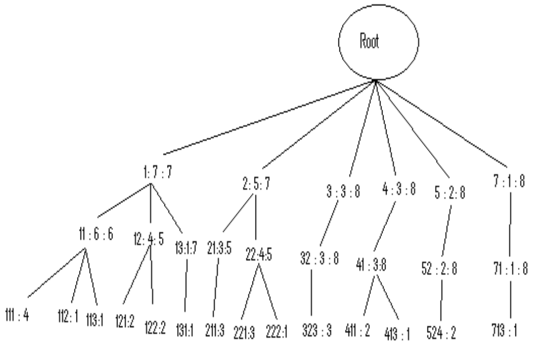


FIGURE 5: CCB-tree for Table4

CCB-Tree Mining Process:

Minimum support for all levels is 4, 3, and 3:

Mining starts from the left most initial node i.e., from 1**: 7 > min_sup and its descendents 11*:6>3 and 111>3. But 112,113<3 so it's considered to be a large 1 frequent pattern.

Finally frequent pattern for level 1: 1**, 2** Level 2: 11*, 12*, 21*, 22* Level 3:111,211,221.

6. EXPERIMENTAL ANALYSIS

Here, we study the experimental analysis of CCB-tree algorithm to mine large-1 frequent pattern. As far as we know, the Apriori algorithm [1 – 5, 11,14] is the only other algorithm that has been designed to mine large-1 frequent pattern. So the first set of experiments we conduct is to compare our algorithm CCB-tree with Apriori.

We also provide the following results for CCB-tree with different choices of the Threshold for different levels; the performance as database size scales.

Finally, we examine the performance of CCB-tree with respect to a synthetic transactional database generated by IBM Quest Market-Basket Synthetic data generator [13]. We used 5000 datasets with three levels; top level of tree has 10 items.

The algorithms were implemented in C language and executed on a Windows machine with Intel CPU.

Threshold	Minimum support thresholds
1	[50, 40, 30]
2	[40, 30, 30]
3	[30, 20, 20]

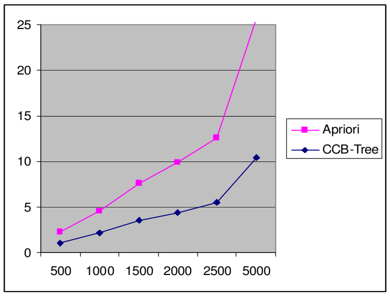


FIGURE 6: Threshold 1

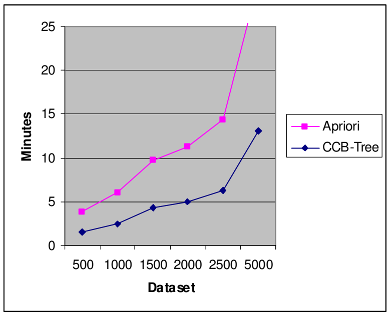


FIGURE 7: Threshold 2

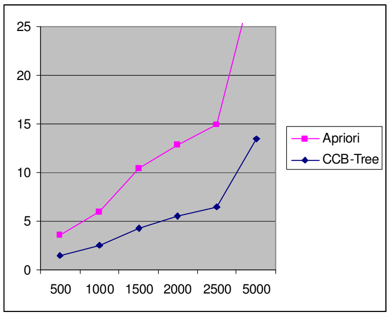


FIGURE 8: Threshold 3

Fig 6 - 8 shows performance measurements for mining large-1 frequent pattern using CCB-tree and Apriori algorithm. The running time and the number of transactions are shown to different minimum support thresholds for different levels ranging from 50 to 20.The above three figures shows two interesting features. First, the relative performance of the two algorithms under any setting is relatively independent of the number of transactions used in the testing, which indicates that the performance is highly relevant to threshold setting. Second, the CCB-tree algorithm have relatively good 'scale-up' behavior since the increase of the number of the transactions in the database will lead to approximately the linear growth of the processing of large transaction databases.

7. CONCLUSION AND FUTURE WORK

Transaction databases in many applications contain data that has built-in hierarchy information. In such databases, uses may be interested in finding association rules among items only at the same level or association rules that span over multiple levels in the hierarchy. In this paper, we presented an efficient preprocessing algorithm for Frequent Pattern Mining in Multidatasets. This algorithm can be used as initial processing step to find frequent pattern generation. As a result, its execution time is much smaller than that of Apriori-based algorithm so that overall time complexity for frequent pattern generation can be reduced.. We conducted extensive experiments and the results confirmed our analysis. In future an efficient algorithm can be generated for frequent pattern mining in multidatasets based on transaction reduction concept.

REFERENCES

[1] Agrawal R,Imienlinski T,Swami A,(1993).Mining association rules between sets of items in large databases. In Proc. Of the ACM SIGMOD Int. Conf. on Management of Data, Pages 207-216.

[2] Agrawal R, and Srikant R, (1994). Fast algorithms for mining association rules. In Proc. Of the 20th Int. Conf. on very Large Databases. Pages 487-499.

[3] Han .J ,Pei .J, and Yin .Y,(2000) Mining Frequent patterns without candidate generation. In Proc. Of ACM-SIGMOD Int. Conf. on Management of Data, pages 1-12.

[4] Han, J., Fu, Y., Discovery of Multiple-Level Association Rules from Large Databases, in Proceedings of the 21st Very Large Data Bases Conference, Morgan Kaufmann, P. 420-431, 1995.

[5] Han, J., Fu, Y., Mining Multiple-Level Association Rules in Large Databases, in IEEE Transactions on Knowledge and Data Engineering, Vol. 11, No. 5, September/October 1999.

[6] Mehmet Kaya, Reda Alhajj, " Mining Multi-Cross-Level Fuzzy Weighted Association rules", Second IEEE International Conference on Intelligent Systems.Vol.1,pp.225-230, 2004

[7] Mohamed Salah Gouider, Amine Farhat, "Mining Multi-level Frequent Itemsets under Constraints", International Journal of Database Theory and Application Vol. 3, No. 4, December, 2010

[8] Pratima Gautham, Pardasani, K. R., "Algorithm for Efficient Multilevel Association Rule Mining", International Journal of Computer Science and Engineering, Vol.2 pp. 1700-1704, 2010.

[9] Popescu, Daniela.E, Mirela Pater, "Multi-Level Database using AFOPT Data Structure and Adaptive Support Constraints", Int. J. of Computers, Comm. & Control, Vol.3,2008.

[10] Rajkumar.N, Karthik.M.R, Sivanada.S.N, "Fast Algorithm for mining multilevel Association Rules,"IEEE Trans. Knowledge and Data Engg., Vol.2 pp. 688-692, 2003.

[11] Thakur, R. S., Jain, R. C., Pardasani, K. R., Mining Level-Crossing Association Rules from Large Databases, in the Journal of Computer Science 2(1), P. 76-81, 2006.

[12] Yinbo WAN, Yong LIANG, Liya DING, "Mining Multilevel Association Rules from Primitive Frequent Itemsets", Journal of Macau University of Science and Technology, Vol.3 No.1, 2009

[13] Synthetic Data generation Code for Associations and Sequential Patterns (IBM Almaden Research center). http://www.almaden.ibm.com/software/quest/Resources/datasets/syndata.html.

[14] Gavin Shaw, 'Discovery & Effective use of Quality Association Rules in Multi-Level Datasets ", Ph.D-Thesis, Queensland University of Technology, Brisbane, Australia,2010.

An Efficient Algorithm for Mining Frequent Itemsets Within Large Windows Over Data Streams

Mahmood Deypir mdeypir@cse.shirazu.ac.ir
School of Engineering, Computer Science and Engineering
Shiraz University
Shiraz, 7134851154, Iran

Mohammad Hadi Sadreddini sadredin@shirazu.ac.ir
School of Engineering, Computer Science and Engineering
Shiraz University
Shiraz, 7134851154, Iran

Abstract

Sliding window is an interesting model for frequent pattern mining over data stream due to handling concept change by considering recent data. In this study, a novel approximate algorithm for frequent itemset mining is proposed which operates in both transactional and time sensitive sliding window model. This algorithm divides the current window into a set of partitions and estimates the support of newly appeared itemsets within the previous partitions of the window. By monitoring essential set of itemsets within incoming data, this algorithm does not waste processing power for itemsets which are not frequent in the current window. Experimental evaluations using both synthetic and real datasets shows the superiority of the proposed algorithm with respect to previously proposed algorithms.

Keywords: Data Stream Mining, Frequent Itemsets, Sliding Window, Support Estimation.

1. INTRODUCTION

A data stream is an infinite amount of data elements which receive at a rapid rate. By the emergence of the application of data stream in business, science and industry, mining this type of data becomes an attractive field in data mining community. Frequent patterns mining [1] over data streams is a challenging problem since it must be solved using minimum resources of main memory and processing power. In a data stream mining algorithm, data elements should be scanned only once due to the rapid data arrival rate [2]. Handling the concept change is another issue. Concept change in the frequent itemset mining problem is changes that occur in the set of frequent itemsets during a data stream mining. Although monitoring previous frequent itemsets in the newly arrived data is an easy task, it is hard to detect new frequent itemsets and computing their supports. Sliding window model is a widely used model to perform frequent itemset mining since it considers only recent transactions and forgets obsolete ones. Due to this reason, a large number of sliding window based algorithms have been devised [3-10]. However, a subset of these studies adaptively maintain and update the set of frequent itemsets [6-10] and others [3-5] only store sliding window transactions in an efficient way and perform the mining task when the user requests. In this study, a novel approach for mining frequent itemsets over data streams is proposed which operate under sliding window model. Experimental evaluations on real and synthetic datasets show the superiority of the proposed approach with respect to previous algorithms. The rest of the paper is organized as follows. The next section introduces some preliminaries and also states the problem. In section 3, some previous related studies are reviewed. Section 4 presents the proposed approach and section 5 empirically compares the approach to its competitors. Finally, section 6 concludes the paper.

2. PRELIMINARIES

Let I={i_1,i_2,…,i_m} be a set of items. Suppose that, DS be a stream of transactions received in sequential order. Each transaction of DS is a subset of I. For an itemset X, which is also a subset of I, a transaction T in DS is said to contain the itemset X if $X \subseteq T$. A transactional sliding window W over data stream DS contains |W| recent transactions in the stream, where |W| is the size of the window. The window slides forward by inserting a new transaction into the window and deleting the oldest transaction from the window. Due to efficiency issues, instead of a single transaction, the unit of insertion and deletion can be a partition (or batch) of transactions. In fact the window contains the n most recent partitions of transactions of the input stream. The first transaction id (Tid) of each partition is regarded as partition id (Pid) of that partition and first Pid of the window is named window id (Wid). An itemset X is said to be frequent in W if Freq(X) ≥ n×|P|×s, where Freq(X), n, |P| and s are frequency of X in W, number of the partitions in the window, partition size and the minimum support threshold, respectively. The number of transactions in each partition, i.e., partition size and number of partitions in each window are fixed during a data stream mining and are the parameters of the mining algorithm. Thus, having a partitioned transactional window W and a minimum support threshold s specified by the user, the problem is defined as mining all frequent itemsets that exists in window W. The results should be continuously updated when the window advances. Due to the rapid arrival rate of transactions, an approximate result of frequent itemsets is acceptable.

3. RELATED WORKS

There are a large number of studies related to frequent itemset mining over data streams. They are mainly belonging to different models of data stream processing including sliding window [3-10], landmark [11-13] and damped models [14, 15]. DSTree [3] and CPS-Tree [4] are two algorithms that use the prefix tree to store raw transactions of sliding window. DSTree uses a fixed tree structure in canonical order of branches while in CPS-Tree, the prefix tree structure is maintaining in support descending order of items to control the amount of memory requirement. Both of [3] and [4] perform the mining task using FP-Growth [16] algorithm that was proposed for static databases. In [5], an algorithm namely MFI-TransSW was proposed which is based on the Apriori algorithm [2]. MFI-TransSW uses a bit string to store the occurrence information of an item within sliding window. Moreover, it mines all frequent itemsets over recent window of transactions. All of [3-5] perform the mining task on the current window when a user requests and don’t adaptively maintain and update the mining result. Therefore, after the mining, when new transactions are arrived from the stream, obtained result becomes invalid for the user and thus the mining task need to be re-executed. Lin et al. [6] proposed a method for mining frequent patterns over time sensitive sliding window. In their method the window is divided into a number of batches for which itemset mining is performed separately. In this algorithm at each timestamp a couple of transactions namely a block are received from input stream. The sliding window contains fixed number of blocks. However, since each batch contains a different number of transactions, different windows over a data stream have various number of transactions. The Moment algorithm [7] finds closed frequent itemsets by maintaining a boundary between frequent closed itemset and other itemsets. In [9] the authors devised an algorithm for mining non-derivable frequent itemsets over data streams. This algorithm continuously maintains non-derivable frequent itemsets of the sliding window. Algorithm of [7] and [9] adaptively mine the concise representation of frequent patterns which are a subset of all set of frequent patterns. The SWIM [8] is a partition based algorithm in which frequent itemsets in one partition of the window are considered for further analysis to find frequent itemsets in whole of the window. It keeps the union of frequent patterns of all partitions and incrementally updates their supports and prunes infrequent ones. Chang and Lee proposed the estWin algorithm [10] that finds recent frequent patterns adaptively over transactional data streams using sliding window model. It uses a reduced minimum support to early monitoring of new itemsets.

DSM-FI [11] is a landmark based algorithm. In this algorithm, every transaction is converted into smaller transactions and inserted into a summary data structure called item-suffix frequent itemset forest which is based on prefix-tree. In [12] the authors used the Chernoff Bound to produce an approximate result of frequent patterns over the landmark window. Zhi-Jun et al. [13] used a lattice structure, referred to as a frequent enumerate tree, which is divided into several equivalent classes of stored patterns with the same transaction-ids in a single class. Frequent patterns are divided into equivalent classes, and only those frequent patterns that represent the two borders of each class are maintained; other frequent patterns are pruned. Chang and Lee proposed an algorithm called estDec based on damped model in which each transaction has a weight decreasing with age [14]. In this method, in order to reduce the effect of old transactions in the set of frequent patterns, a decay rate is defined. In [15] an algorithm similar to the estMax is proposed for mining maximal frequent itemsets over data streams based on damped model.

4. THE PROPOSED ALGORITHM

In [6, 8], frequent itemsets of a new partition are mined using the FP-Growth [16] algorithm and since then the found frequent itemsets are checked against new partitions to update their support. The idea of these algorithms is based on the fact that each frequent itemset of the window are frequent in at least one partition of the window. The idea is inspired by the partitioning algorithm [17] for static databases in which, it is proved that a frequent itemset is frequent in at least one partition of a database. However, exploiting this criterion in data stream mining, increases the number of frequent itemsets that required to be monitored in the incoming transactions. The reason is that the reverse of this criterion is not correct. That is:

Theorem. An itemset that is frequent in a partition of the window might be infrequent in whole window.

Proof. An itemset which is frequent in a partition of the window might have low support in other partitions of the window. Therefore, its overall support in whole window might be smaller than the minimum support threshold. □

Based on the above statement, in our algorithm we don’t monitor each frequent itemset of a new partition. Instead, for such frequent itemset we estimate their support in the previous partitions individually using their subsets. If the sum of estimated support and actual support of the itemset is greater than minimum support threshold, the itemset is inserted to the monitoring prefix tree and their support becomes verified in subsequent new partitions. Moreover, by expiring each partition of the window, support of itemsets in the prefix tree are updated. For each itemset, the process of updating continues until the itemset is frequent in the window. In our approach, the estimation of support is partition based estimation. That is, the support of an itemset is estimated in each partition of the window. Therefore, estimated support of an itemset is equal to sum of all estimated values. For an itemset, its actual support and estimated support in different partitions of the window are stored in the corresponding node of the prefix tree. When an itemset identified as frequent in a new partition, its support is estimated in previous partitions of the window. For an n-itemset its longest subsets have length of n-1 or smaller. For each previous partition first, among the supports of their subsets, minimum value is selected. Longer subsets are checked first since longer subsets have closer value to the actual support of the itemset. If actual values of long subsets are not contained in a partition, shorter ones are tested. Therefore, a high quality estimated value for the itemset in each partition is computed. An itemset in the new partition is inserted into the prefix tree if the following conditions are hold:

$$\begin{cases} F_n \geq Sup \\ \sum_{i=1}^{n-1} EF_i + F_n \geq Sup \end{cases} \quad (1)$$

Where EF_i and F_n are estimated support in each partition i and actual support of the itemset in the newly received partition. The window contains n partitions of transactions. Considering the estimated supports of previous partitions to insert and monitor the support of the itemset reduces the size of prefix tree and also enhances the processing time. Estimating support using actual counts of individual partitions improves the mining quality since more realistic value is obtained.

For a newly inserted node, actual support of the partition and estimated support of previous partitions are stored separately. For this node, its actual supports in subsequent partitions are stored since it is monitored in newly arrived partitions. This information of individual partitions of the window is used to remove the oldest partitions efficiently. Table 1 summarizes elements that are stored in the prefix tree nodes. In this prefix tree each node represents an itemset which can be induced in the path from the root to the node. Prefix sharing among the itemset in the tree, reduces the amount of memory requirement.

Element	Purpose
ID	Item ID
Cs	Actual Count of the itemset in partitions of the window
ECs	Estimated support of the itemset in partitions of the window
Children	Set of pointers to the children of the node

TABLE 1: Information contained in each node of prefix tree

After adding a newly arrived partition, to complete the window sliding phase, the oldest partition of transactions should be removed. For each node of the prefix tree, if corresponding itemset has estimated support in this partition, the value is removed. Otherwise, the actual support value in this partition is neglected. Therefore, the oldest partition removal process does not need FP-Tree of the oldest partition as in SWIM [8] or current transactions of the window as in estWin [10] algorithm. As result, removing obsolete information is performed using smaller memory and processing time. In the partition removal process, infrequent nodes and their descendents are also deleted. When information of the oldest partition is removed from a node, its support is reduced and if the support falls below the threshold the node and their descendent are removed recursively from the prefix tree. The reason for deletion of descendents is due to the Apriori principle which states that all supersets of an infrequent node are also infrequent.

A high level pseudo code of the proposed algorithm is shown in Figure 1. As shown in this Figure, for each itemset of the prefix tree (PT), its support is updated using the newly inserted partition (P). Frequent patterns of the new partition are found by applying the FP-Growth algorithm.

```
$\forall e \in PT\ update\ e.sup$
F = FP-Growth(P)
$\forall e \in F\ e.supp + estimated(e) \geq sup$
    $Insert\ e\ into\ PT$
$if\ |W| > n \times |P|$
    $\forall e \in PT\ remove\ oldest\ pane$
        If $e.supp < supp$
            $Erase\ e\ and\ all\ of\ its\ descendents\ recursively\ from\ PT$
```

FIGURE 1: The Proposed Algorithm

For each itemset of the new partition, if its support in the new partition in addition to its estimated support in the previous partitions is greater than or equal to minimum support threshold, it is inserted to the prefix tree. If the window size is greater than its specified number of the partition, the oldest partition must be removed from the window to preserve fixed size window. Hence, for each itemset in the prefix tree, its support information of the oldest partition containing estimated or actual support is deleted from the tree. By removing this information form corresponding arrays, if the itemset becomes infrequent, it and its subsets are erased from the tree.

In the proposed algorithm, information of equal sized partitions is separately stored in the prefix tree. The proposed approach can be also used in time sensitive window where at each timestamp a number of transactions are received from a stream. These transactions can be regarded as a new partition and processed according to the above described approach. However, since at each timestamp, different number of transactions is arrived, the partition size is not fixed during the stream mining process. Using the proposed approach, extracting old transactions from the window is performed efficiently. In [6], information of the partitions of the window are stored in tables which does not benefits from prefix sharing and requires large amount of memory and processing time. Moreover, all frequent itemsets of a new partition are monitored and their previous supports are estimated imprecisely.

5. EXPERIMENTAL EVALUATION

The proposed algorithm is experimentally evaluated with respect to previously proposed algorithms. The estWin and SWIM are selected for comparison since they similarly mine frequent itemsets adaptively over data streams. We have implemented all algorithm using C++ and STL template library. All experiments were conducted on P4 Intel CPU running Windows XP with 2 GB of RAM. We have compared the algorithms in terms of runtime since it is an important factor of every data stream mining algorithm. Two datasets are selected for experimentation. First dataset is a real dataset named BMS-POS and second dataset is a synthetic dataset generated using synthetic data generator [1]. Specifications of these datasets are summarized in the Table 1.

Dataset	#trans	#items	Max. length	Avg. length
BMS-POS	515,597	1657	164	6.53
T40I10D100K	100,000	942	77	39.61

TABLE 2: Datasets specifications

Since the value of minimum support threshold has direct effect on the runtime, the first experiment compares the algorithms using different values of this parameter on BMS-POS dataset. The results are shown in Figure 2.

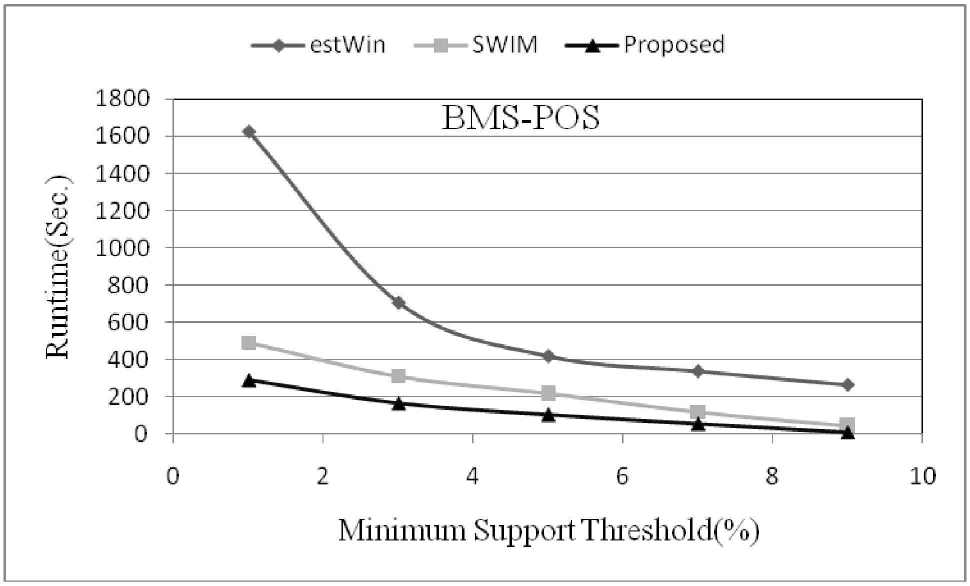


FIGURE 2: Runtime Comparison on BMS-POS

As shown in this figure, the proposed algorithm has the better runtime for different minimum support values. As the minimum support threshold decreases, the performance gap of our algorithm with respect to the other methods increases. The reason is, for lower minimum support thresholds, the number of frequent itemsets is increased. In this situation, SWIM requires to verify the support of a large number of patterns in different panes of the window. On the other hand, since the estWin uses a reduced minimum support value, i.e., significance instead of the actual threshold, the number of generated itemsets becomes prohibitively large. Therefore, the proposed algorithm operates better than both SWIM and estWin especially in lower minimum support thresholds.

The second experiment mesures the runtime for different values of minimum support thresholds on T40I10D100K synthetic dataset. The result is plotted on Figure 3.

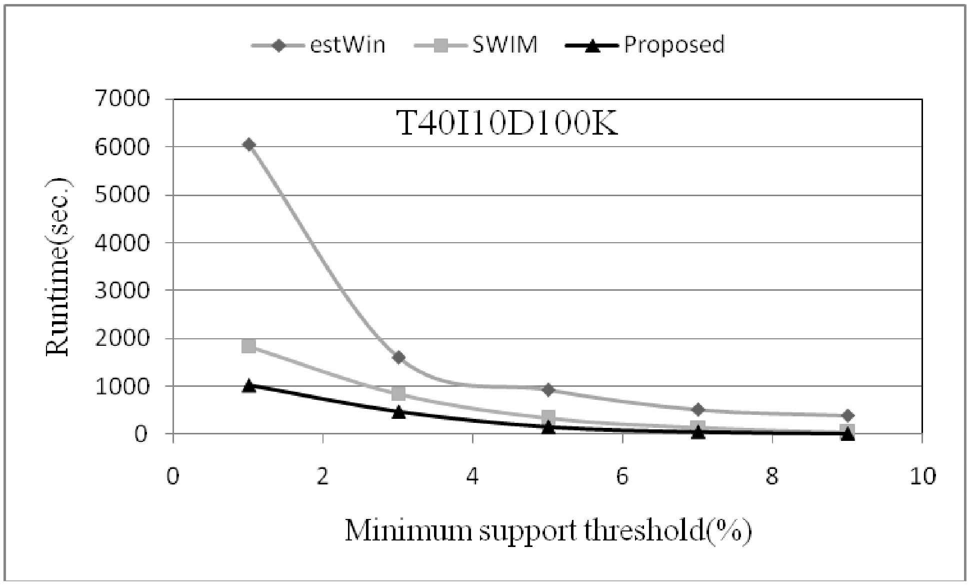


FIGURE 3: Runtime comparison on T40I10D100K

As shown in this figure, the proposed algorithm has lowest runtime. However, its runtime is closed to the SWIM algorithm. In addition to the above mentioned reasons, in both the SWIM and the proposed algorithms, the incoming transactions are batch processed while the estWin algorithm processes a single transaction at each sliding. Hence, both of them have better runtime. The SWIM stores transactional FP-Tree of each partition of the window to verify support value of new incoming partitions. On the other hand the proposed algorithm throws away the old transactions and estimates the support values of the new itemsets within previous partitions. Support estimation is faster than verifying and thus the proposed algorithm is faster than the SWIM.

6. CONCLUSION

In this study, a new algorithm for online frequent itemset mining over data streams is proposed. This algorithm has better runtime with respect to previously proposed estWin and SWIM algorithms. A prefix tree is only data structure used by the proposed algorithm while in the estWin and SWIM transactions of the window are also stored in addition to frequent itemsets of the current window. Therefore, the proposed algorithm has lower memory requirements. Although, the algorithm is proposed for operating in transactional window, it is also suitable for time sensitive window in which at any timestamp a different number of transactions are received from a stream.

7. RFERENCES

[1] R. Agrawal and R. Srikant, “Fast algorithms for mining association rules” in Proc. Int. Conf. on Very Large Databases, pp. 487–499, 1994.

[2] J. Han, H. Cheng, D. Xin, & X. Yan. “Frequent pattern mining: current status and future directions”, *Data Mining and Knowledge Discovery*,vol. 15(1), pp. 55–86, 2007.

[3] C.K.-S. Leung, & Q.I. Khan. “DSTree: a tree structure for the mining of frequent sets from data streams”, Proc. ICDM, 928–932, 2006.

[4] S.K.Tanbeer, C. F. Ahmed, B.-S. Jeong, & Y.-K. Lee. “Sliding window-based frequent pattern mining over data streams”, *Information Sciences*, vol. 179(22), pp. 3843-3865, 2009.

[5] H.-F. Li, S.-Y. Lee. “Mining frequent itemsets over data streams using efficient window sliding techniques”, *Expert Systems with Applications*, vol. 36(2), pp. 1466–1477, 2009.

[6] C.-H. Lin, D.-Y. Chiu, Y.-H. Wu, & A.L.P. Chen. “Mining frequent itemsets from data streams with a time-sensitive sliding window”, Proc. SIAM Int. Conf. Data Mining, 2005.

[7] Y. Chi, H. Wang, P.S. Yu, & R.R. Muntz. “Catch the moment: maintaining closed frequent itemsets over a data stream sliding window” *Knowledge and Information Systems*, 10(3), pp. 265–294, 2006.

[8] B. Mozafari, H. Thakkar, & C. Zaniolo. “Verifying and mining frequent patterns from large windows over data streams”, Proc. Int. Conf. ICDE, pp.179–188, 2008.

[9] H. Li, & H. Chen. “Mining non-derivable frequent itemsets over data stream”, *Data & Knowledge Engineering*, vol. 68(5), pp. 481-498, 2009.

[10] J.H. Chang, & W.S. Lee. “estWin: Online data stream mining of recent frequent itemsets by sliding window method” *Journal of Information Science*, vol. 31(2), pp. 76–90, 2005.

[11] H. F. Li, S. Y. Lee, & M. K. Shan “An efficient algorithm for mining frequent itemsets over the entire history of data streams” Proc. Int. Workshop on Knowledge Discovery in Data Streams, 2004.

[12] J.X. Yu, Z. Chong, H. Lu, Z. Zhang, Z., & A. Zhou. “A false negative approach to mining frequent itemsets from high speed transactional data streams” *Information Sciences*, vol. 176(14), pp. 1986–2015, 2006.

[13] X. Zhi-Jun, C. Hong, & C. Li. “An efficient algorithm for frequent itemset mining on data streams” Proc. ICDM, 474–491, 2006.

[14] J. Chang, W. Lee, “Finding recently frequent itemsets adaptively over online transactional data streams”, *Information Systems*, vol. 31 (8), pp. 849-869, 2006.

[15] J.H. Chang, W.S. Lee, “estMax: Tracing Maximal Frequent Itemsets Instantly over Online Transactional Data Streams”, *IEEE Transactions on Knowledge and Data Engineering*, vol. 21 (10) pp.1418-1431, 2009.

[16] J. Han, J. Pei, Y. Yin, & R. Mao. “Mining Frequent Patterns without Candidate Generation: A Frequent-Pattern Tree Approach”, *Data Mining and Knowledge Discovery*, vol. 8(1), pp. 53-87, 2004.

[17] A. Savasere, E. Omiecinski, and S. Navathe, “An efficient algorithm for mining association in large databases”, in Proceeding of the VLDB International Conference on Very Large Databases, pp. 432–444, 1995.

INSTRUCTIONS TO CONTRIBUTORS

Data Engineering refers to the use of data engineering techniques and methodologies in the design, development and assessment of computer systems for different computing platforms and application environments. With the proliferation of the different forms of data and its rich semantics, the need for sophisticated techniques has resulted an in-depth content processing, engineering analysis, indexing, learning, mining, searching, management, and retrieval of data.

International Journal of Data Engineering (IJDE) is a peer reviewed scientific journal for sharing and exchanging research and results to problems encountered in today's data engineering societies. IJDE especially encourage submissions that make efforts (1) to expose practitioners to the most recent research results, tools, and practices in data engineering topics; (2) to raise awareness in the research community of the data engineering problems that arise in practice; (3) to promote the exchange of data & information engineering technologies and experiences among researchers and practitioners; and (4) to identify new issues and directions for future research and development in the data & information engineering fields. IJDE is a peer review journal that targets researchers and practitioners working on data engineering and data management.

To build its International reputation, we are disseminating the publication information through Google Books, Google Scholar, Directory of Open Access Journals (DOAJ), Open J Gate, ScientificCommons, Docstoc and many more. Our International Editors are working on establishing ISI listing and a good impact factor for IJDE.

The initial efforts helped to shape the editorial policy and to sharpen the focus of the journal. Starting with volume 2, 2011, IJDE appears in more focused issues. Besides normal publications, IJDE intend to organized special issues on more focused topics. Each special issue will have a designated editor (editors) – either member of the editorial board or another recognized specialist in the respective field.

We are open to contributions, proposals for any topic as well as for editors and reviewers. We understand that it is through the effort of volunteers that CSC Journals continues to grow and flourish.

IJDE LIST OF TOPICS

The realm of International Journal of Data Engineering (IJDE) extends, but not limited, to the following:

- Approximation and Uncertainty in Databases and Pro
- Autonomic Databases
- Data Engineering
- Data Engineering Algorithms
- Data Engineering for Ubiquitous Mobile Distributed
- Data Engineering Models
- Data Integration
- Data Mining and Knowledge Discovery
- Data Ontologies
- Data Privacy and Security
- Data Query Optimization in Databases
- Data Streams and Sensor Networks
- Data Warehousing
- Database Tuning
- Database User Interfaces and Information Visualiza
- Knowledge Technologies
- Metadata Management and Semantic Interoperability
- OLAP and Data Grids
- Personalized Databases
- Query Processing in Databases
- Scientific Biomedical and Other Advanced Database
- Semantic Web
- Social Information Management
- Spatial Temporal

CALL FOR PAPERS

Volume: 3 - **Issue:** 1 - February 2012

i. Paper Submission: November 30, 2011 **ii. Author Notification:** January 01, 2012

iii. Issue Publication: January / February 2012

CONTACT INFORMATION

Computer Science Journals Sdn BhD
B-5-8 Plaza Mont Kiara, Mont Kiara
50480, Kuala Lumpur, MALAYSIA

Phone: 006 03 6207 1607
006 03 2782 6991

Fax: 006 03 6207 1697

Email: cscpress@cscjournals.org

CSC PUBLISHERS © 2011
COMPUTER SCIENCE JOURNALS SDN BHD
M-3-19, PLAZA DAMAS
SRI HARTAMAS
50480, KUALA LUMPUR
MALAYSIA

PHONE: 006 03 6207 1607
006 03 2782 6991

FAX: 006 03 6207 1697
EMAIL: cscpress@cscjournals.org