An Improved Approach to Find Frequent Web Access Patterns from Web Logs

Asim Munshi[1], Dr Shyamal Tanna[2]

[1] Student, Computer Engineering, LJ Institute of Engineering, Gujarat, India
[2]Associate Professor, Computer Engineering, LJ Institute of Engineering, Gujarat, India

ABSTRACT

Web usage mining refers to programmed revelation of examples and related information, gathered or created as an after effect of client connections with one or more Web destinations. Principle objective is to investigate the behavioral examples and profiles of clients interfacing with a Web page. The found examples are represented as collections of pages, objects, or resources that are frequently accessed by groups of users with common interests. Web usage mining consists of three phases, namely pre-processing, pattern discovery, and pattern analysis. . In the pattern discovery phase, frequent pattern discovery algorithms applied on raw data. In the pattern analysis phase interesting knowledge is extracted from frequent patterns and these results can be used further. By our Literature Survey we have found that most of the frequent pattern mining algorithms are either not scalable or too complex so we propose a new an effective method for mining frequent item sets which is comparatively more scalable as well as simple and will provide better mining performance.

Keyword - *Frequent Items, Web Mining, Web Usage Mining, Frequent Pattern Mining, Association rule Mining*

1. Introduction

The internet features have influenced virtually every part of our universe. Since the number of web sites along with website pages are increasing and their features are improved rapidly, discovering and understanding web users surfing behavior are essential for the development of successful web monitoring and recommendation systems. One of the ways to achieve the above is through mining of Web Logs Web Usage mining [2] is the process of applying data mining techniques to the discovery of usage patterns from Web data, targeted towards various practical applications such as personalized web search and surfing, web recommendation systems. Data mining efforts associated with the Web, called Web mining, can be broadly divided into three classes, i.e. web content mining, web structure mining, and web usage mining. It attempts to discover useful knowledge from the secondary data, especially those contained in Web log files. Other sources can be browser logs, user profiles, user sessions, bookmarks, folders and scrolls. These data are obtained from the interactions of the users with the Web.Web mining can be broadly divided into three distinct categories, based on which type of data is to be mined the main categories can be given as follows..

1.1 Web content mining

Web content mining is the process of extracting useful information from the contents of web documents. Content data is the collection of facts a web page is designed to contain. It may consist of text, images, audio, video, or structured records such as lists and tables. Application of text mining to web content has been the most widely researched. Issues addressed in text mining include topic discovery and tracking, extracting association patterns, clustering of web documents and classification of web pages.

1.2 Web Structure Mining

The structure of a typical web graph consists of web pages as nodes, and hyperlinks as edges connecting related pages. Web structure mining is the process of discovering structure information from the web. Both hyperlinks and document structure can be mine

1.3 Web Usage Mining

The Web usage mining is the application of data mining techniques to discover interesting usage patterns from web usage data, in order to understand and better serve the needs of web-based applications. Usage data captures the identity or origin of web users along with their browsing behavior at a web site. Web usage mining itself can be classified further depending on the kind of usage data.

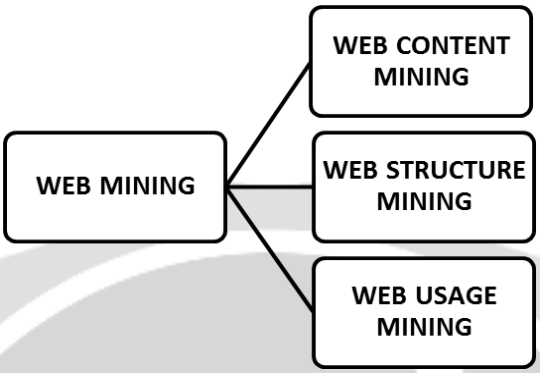

Fig 1 - Taxonomy of Web Mining

2. LITERATURE REVIEW

2.1 Apriori Algorithm

Apriori is a typical algorithm for frequent item set mining and association rule learning over transactional databases. It is proceed by recognize the frequent individual items in the database and extend them to big and big item sets as long as those item sets appear sufficiently often in the database. The frequent item sets find out by Apriori can be used to find out association rules which highlight general trends in the database: this has applications in domains such as market basket analysis. . It is is a typical algorithm for frequent item set mining and association rule learning over transactional databases. It is proceed by recognize the frequent individual items in the database and extend them to big and big item sets as long as those item sets appear sufficiently often in the databas e. The frequent item sets find out by Apriori can be used to find out association rules which highlight general trends in the database: this has applications in domains such as market basket analysis. It uses important property called Apriori property is used to reduce the search. All the non-empty subsets of frequent item sets must also be frequent this property belongs to a special category of properties called Anti-monotone. If a set can't pass a certain test than all of its supersets will fail the same test this property is called anti-monotone.

Apriori follows two steps approach:

- In the first step it joins two item sets which contain k-1 common items in kth pass. The first pass starts from the single item the resulting set is called the candidate set Ck.
- In the second step, the algorithm counts the occurrence of each candidate set and prunes all infrequent item sets. The algorithm ends when no further extension found

2.2 FP-Growth Algorithm

FP-growth is a well-known algorithm that uses the FP-tree data structure to achieve a representation of the database transactions and employs a divide-and-conquer approach to decompose the mining problem into a set of smaller problems. In essence, it mines all the frequent item sets by recursively finding all frequent item sets in the conditional pattern base which is efficiently constructed with the help of a node link structure.

The algorithm consists of two steps:

- Compress a large database into a compact, Frequent Pattern tree (FP-tree) structure – highly condensed, but complete for frequent pattern mining and avoid costly database scans

- Develop an efficient, FP-tree-based frequent pattern mining method (FP-growth) – A divide-and-conquer methodology: decompose mining tasks into smaller ones and avoid candidate generation: sub-database test only

FP growth is a noble approach that allows frequent patterns to be identified without generating candidate. But for large database and frequently changing or real time database, creating this tree can be a time consuming process.

2.3 Other Methods

In [1] an algorithm to predict user's behavior is proposed. The algorithm is named as single scan pattern recognition algorithm as it scans the database only once. The connectivity of the web data is taken into consideration. In usual approach to find frequent pattern a pattern tree is created and then analysis is done but in the approach proposed by this paper there is no need for tree creation and analysis is done on website architecture. In this algorithm first transaction is taken and then path is taken and followed, each node has 0 as account initially the value of count is increased by one whenever a path is traversed. When the structure of website structure is very complex the graph creation can become very complex and can result in poor outcome. Also in today's world the structure of website keeps on changing and every time the structure changes anew scan will be required. Hence a system for extracting user's navigational behavior is presented. An undirected graph based on connectivity between each pair of the Web pages was considered and also proposed a new formula for allocating weights to each edge of the graph.

In [2] a graph based approach is suggested to mine frequent sequential access patterns for users. It uses frequent sequential pattern algorithm. It compromises of three basic steps which are - Construct a graph, Prune a graph, Mine a frequent sequential pattern from web usage graph. It emphasizes on to showing how frequent pattern discovery tasks can be accomplished by capturing complex user's browsing behavior in to a graph data structure in order to obtain hidden useful user's access patterns.

In [7] the same above work is extended to generate useful recommendations Recommender System is one kind of filtering system which is used for ranking or priority of the objects. The recommendations are retrieved for a given user's web access sequence. Length of the user web access sequence must satisfy the thresholds. If its length is greater than max length then we have to remove first element. If it contains next item then the recommendation rule order by the support is returned

In [3] a method to predict the user's navigation patterns is proposed using clustering and classification from Web log data. First phase of this method focuses on separating users in Web log data, and in the second phase clustering process is used to group the users with similar preferences. Finally in the third phase the results of classification and clustering are used to predict the users' next requests.

Theint Theint, Aye in [4] stressed the importance of pre-processing step in the mining of web logs. Data pre-processing is an important task of Web usage mining. Therefore, data must be processed before applying data mining techniques to discover user access patterns from web log. The data preparation process is often the most time consuming. This paper presents two algorithms for field extraction and data cleaning. Not every access to the content should be taken into consideration. So this system removes accesses to irrelevant items and failed requests in data cleaning. After that necessary items remain for purpose of analysis. Speed up extraction time when user's interested information is retrieved and users' accessed pages is discovered from log data. The information in these records is sufficient to obtain session information.

In [6], a model for association rules to mine the generated frequent k-itemset is proposed. This process is taken as extraction of rules which expressed most useful information. Therefore, transactional knowledge of using websites is considered to solve the purpose. In this paper interestingness measure that plays an important role in removing invalid rules thereby reducing the size of rule data sets is used. The performance analysis attempted with Apriori, most frequent rule mining algorithm and interestingness measure to compare the efficiency of websites. The proposed work reduces large number of immaterial rules and produces new set of rules with interesting measure. The current algorithm is not capable of handle very large number of log entries and thus suffers from scalability point of view.

3. PROPOSED SYSTEM

We propose a method which constructs a compact prefix-tree structure with one database scan and provides the same mining performance as the FP-growth technique by efficient tree restructuring process

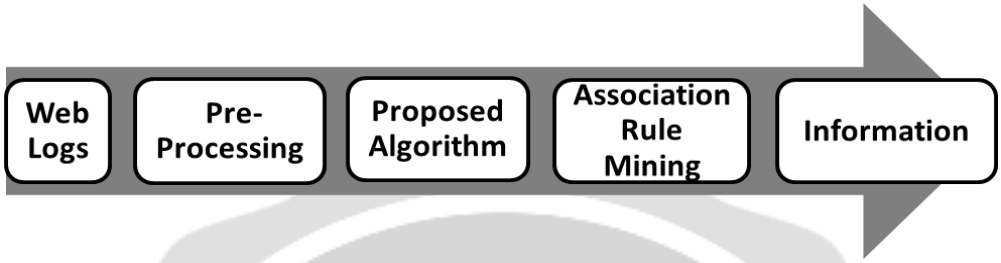

Fig 2 – Proposed System Model

The Algorithm compromises of following three steps

- Insertion phase
- Restructure phase
- Mega Node Phase

3.1 Algorithm – Single Scan

Input: Transaction Database and Min_Sup.

Output: Frequent Item-sets.

Step 1: Create root for tree ({})
root → null
Step 2: For Each Transaction from DB
Do
Select the Item from Transaction and sort Alphanumeric order → Insert into treeand also add one count to I-list
End
End
Step 3: Discard items from I-list
Support (I) < Min_Sup

Step 4: Arrange remaining items of I-list into descending order and Restructure the tree

Step 5: After creation of tree structure, Transform nodes with similar occurrence into mega nodes

Step 6: Mine the frequent item set from restructured tree doing Projection of conditional base tree

3.2 Theoretical Analysis

In Insertion phase it scans transactions, inserts them into the tree according to item appearance order. It also maintains I-list and updates frequency count of respective items in I-list when inserts transaction into the tree. In Restructure phase it rearranges the I-list according to frequency descending order of items and after creation of tree structure nodes with similar occurrence will be transformed into mega nodes

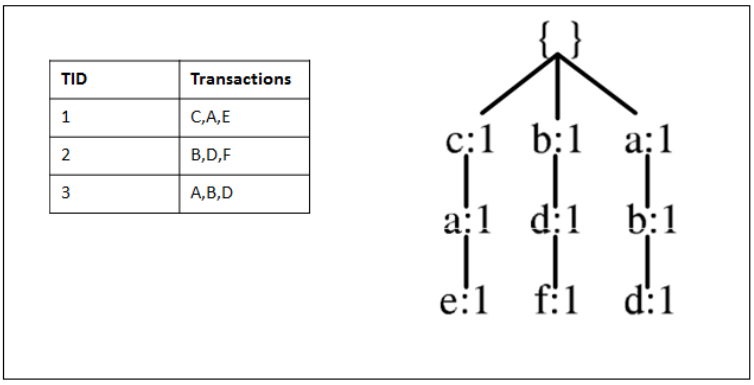

Fig 3 – Insertion Phase

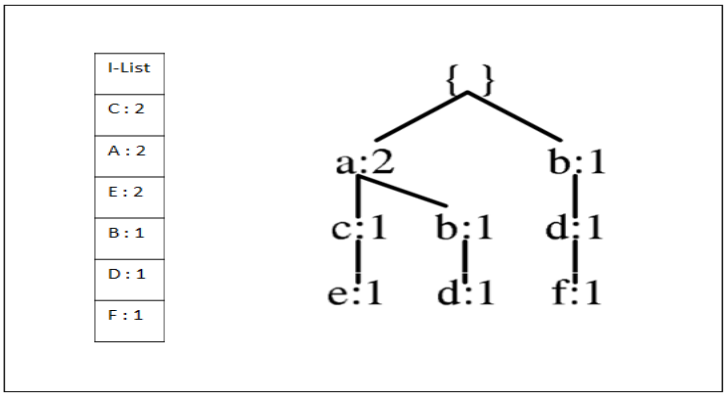

Fig 4 –Restructure Phase

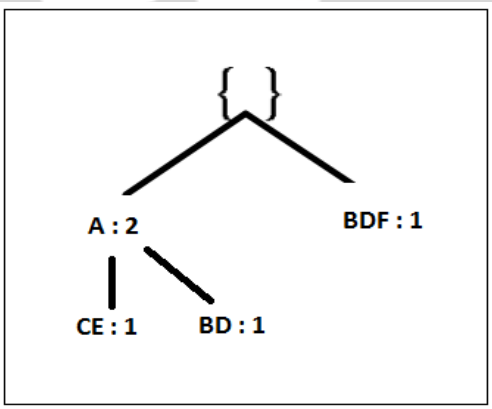

Fig 5 – Node Transformation Phase

4. EXPERIMENTAL RESULTS

We have used Eclipse IDE for compiling and execution purpose. We have also used the SPMF tool, It contains a base workspace and an extensible plug-in system for customizing the environment which mostly written in Java Experiment was carried out on live web logs of a book selling website. The system used by us was a windows 10 system with 4 gb of ram and i3 processor.

4.1 Pre-Processing

We Pre-processed the web logs, simplified the web logs and also the new file which we obtained is smaller in size. This Preprocessed file will be given as an input to our proposed algorithm. Each IP address in web logs was assigned a new line and each book was assigned an id, the id's in a same line represent the books accessed by a single user/IP address.

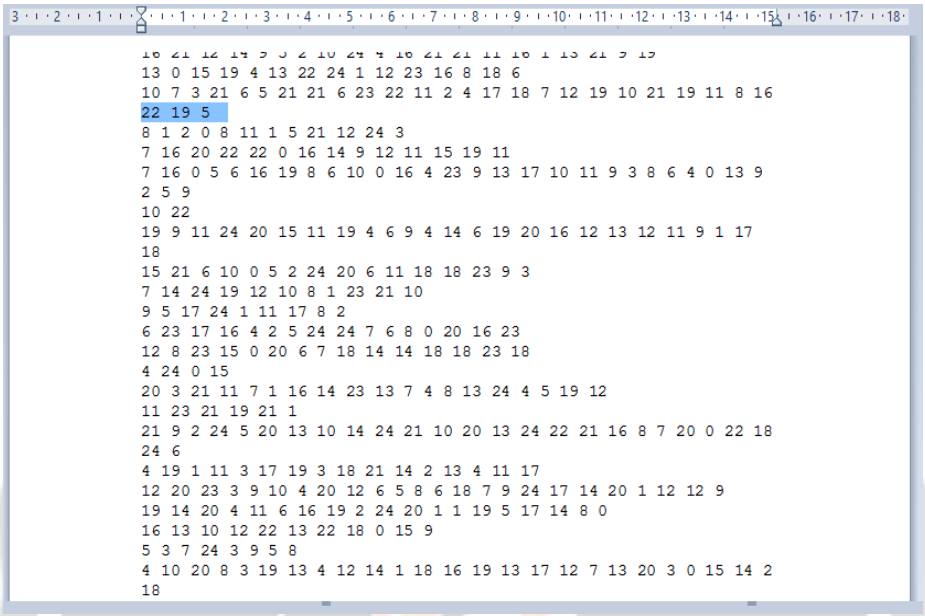

Fig 6 – Pre-processed File

4.2 Comparison based on time and memory

The below table shows comparison between the old and new approach based on time and memory, the readings were taken for different min_sup as shown below

Table 1 : Comparison of approach

Approach		***Different Values of Min_Sup***		
		0.72	**0.73**	**0.75**
Old	***Time(ms)***	66146	22187	7543
	Memory(mb)	348.65644	279.15219	61.9908
Proposed	***Time(ms)***	27710	7769	2687
	Memory(mb)	222.4559	188.5733	40.0657

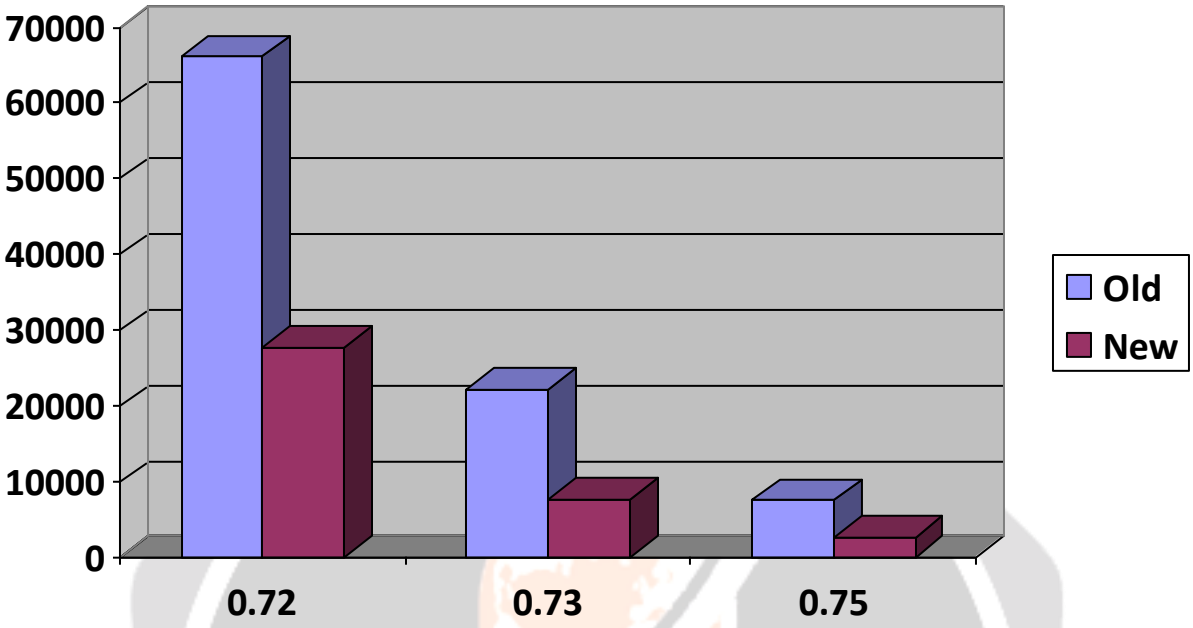

Chart 1: Execution Time comparison of OLD and Proposed Algorithm with different min_sup
Here x axis represents different values of min_sup &y axis represents execution time in ms

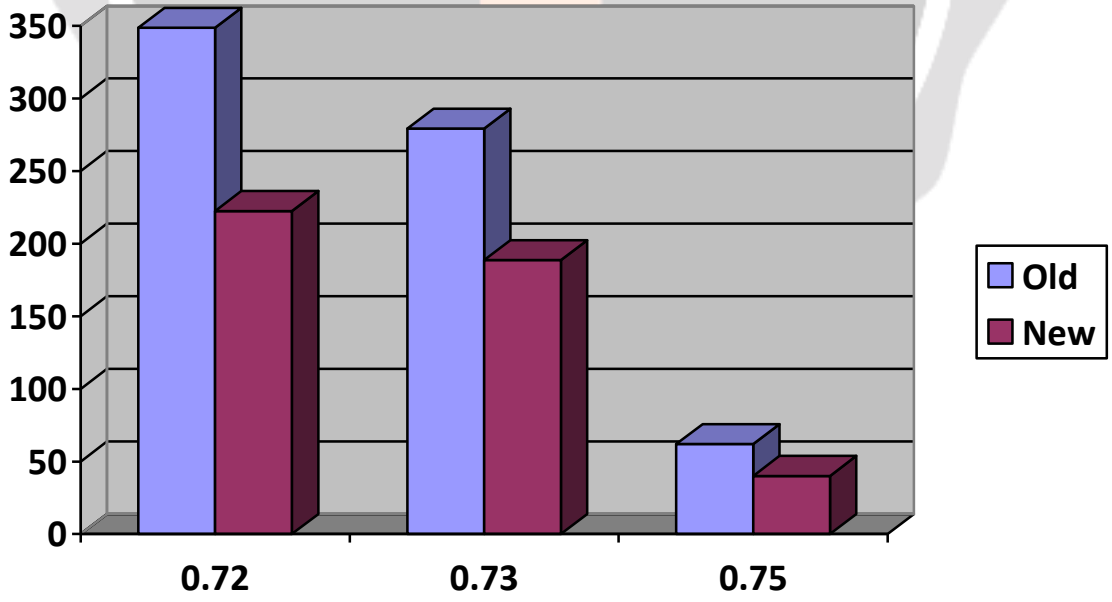

Chart 2: Memory comparison of OLD and Proposed Algorithm with different min_sup
Here x axis represents different values of min_sup &y axis represents memory in mb

5. CONCLUSIONS

The frequent pattern mining is important task of data mining. According to observation all other frequent pattern mining algorithms are not more memory efficient compared to proposed approach. Frequent rules generating algorithm scan database more than one time and are complex hence they will take more time and memory. The proposed algorithm scans the database only once and is a lot simple compared to other algorithms and hence takes less memory as well as time compared to the previous approach. The results shown in the comparison table prove this. Future work may be done on generating recommendations using this approach.

6. REFERENCES

[1] Murli Manohar Sharma, Anju Bala, "An approach for frequent access pattern identification in web usage mining" IEEE Advances in Computing, Communications and Informatics (ICACCI, 2014 International Conference in September 2014, Pages 730 – 735, ISBN 978-1-4799-3078-4.

[2] Dheeraj Kumar Singh, Varsha Sharma, Sanjeev Sharma "Graph based Approach for Mining Frequent Sequential Access Patterns of Web pages", International Journal of Computer Applications 2012 Applications (0975 – 8887) Volume 40– No.10, February 2012

[3] V. Sujatha, Punithavalli "Improved user Navigation pattern prediction technique from web log data", Science Direct, Procedia engineering, Volume 30, 2012, Pages 92 – 99.

[4] Theint Theint, Aye "Web Log Cleaning for Mining of Web Usage Patterns" IEEE Computer Research and Development (ICCRD), 2011 3rd International Conference (Volume:2), March 2011, Pages 490 – 494, ISBN 978-1-61284-839-6.

[5] Hussain S, Asgar S, Masood N" Web usage mining: A survey on pre-processing of web log file" IEEE Information and Emerging Technologies (ICIET), 2010 International Conference, June 2010, Page 1 – 5, ISBN 978-1-4244-8001-2.

[6] Avadh Kishor Singh, Ajeet Kumar, Ashish K. Maurya, "Association Rule Mining for Web Usage Data to Improve Website", IEEE Advances in Engineering and Technology Research (ICAETR), 2014 International Conference, Aug 2014, Page 1 – 6, ISSN 2347-9337.

[7] Valera M, Chauhan U, "An efficient web recommender systembased on approach of mining frequent sequential pattern from customized web log pre -processing", IEEE fourth conference on computing, communications & networking technologies (ICCCNT), July 2013, Page 1 – 6, ISBN 978-1-4799-3925-1.

[8] R. Sharma and K. Kaur, "Review of Web Structure Mining Techniques using Clustering and Ranking Algorithms", International Journal of Research in Computer and Communication, IJRCCT, Vol. 3, No. 6, pp. 663-668, 2014 ISSN 2278 - 5841.

[9] H. Fu Li, S. Lee and M. Kwan, "Online mining (recently) maximal frequent item sets over data streams", Proceedings of the 15th International Workshop on Research Issues in Data Engineering, pp no1118, 2005

[10] C. Tsai, C. Lai and M. Chiang, "Data mining for internet of things: A survey," IEEE Communication Surveys & Tutorials, Vol. 16, No. 1, 2014 ISSN 1553-877X.

[11] M. S. Chen, J. S. Park and P. S. Yu, "Efiicient Data Mining for Path Traversal Patterns", IEEE Transaction on Knowledge and Data Engineering, Vol. 10, Issue 2, pp no. 209-220,ISSN 1041-4317.

[12] F. S. Gharehchopogh and Z. A. Khalifelu, "Analysis and evaluation of unstructured data: text mining versus natural language processing" Application of Information and Communication Technologies (AICT), 2011 5th International Conference on. IEEE, pp. 1-4, 2011 ISBN: 978-1-61284-831-0.

[13] C. Kaur, and R. R. Aggarwal, "Reference Scan Algorithm for Path Traversal Patterns.", International Journal of Computer Applications, Vol. 48. Pp no. 20-25, 2012 ISSN 7360-013