# Using association rule learning in a Semantic file system

Aseem Gogte, Sahil Gupta, Harshvardhan Pandit, Rohit Sharma

*Department of Computer Engineering, RSCOE, Tathawade*
*Pune University, Pune*
aseem2691@gmail.com
euphoric.sg@gmail.com
hpandit86@gmail.com
rohitc.dude@gmail.com

***Abstract*— The main concern in information-rich systems is to efficiently navigate and access desired information. Traversing a file system using long pathnames is cumbersome and requires the user to accurately remember where each file can be found. Semantic file systems help in finding a file under various contexts (called as tags) which act like directories. However, the user still has to traverse these paths to reach the files. It is left up to the user to create and manage an efficient system of tags. Since all data items and relations are stored in a database structure, association rule learning can be used to form useful relations between various tags and files. Using various algorithms, we can create associations (links) between data sets that can help the user traverse related data without the burden of long pathnames. Popular data mining algorithms can be readily adapted to a semantic file system's database. Thus utilizing associations, a semantic file system can offer a more efficient and contextual way to search, store and organize data.**

***Keywords*— *semantics, indexing, classification, database, tagging, virtual file system, information access, metadata***

## I. INTRODUCTION

The boom in information has created a situation where it becomes difficult to categorize and search relevant information. Compared to a file system, the web has highly active services and algorithms for data navigation. Tools such as Google, DuckDuckGo [1], and Apple's Siri etc. allow the user to search using keywords and show relevant information by utilizing data mining concepts. Developers creating web services think up of innovative ways to provide easy access to data. However, file system developers are still mostly focusing on stability and performance. While this becomes a necessity in a company server, the home user is more concerned with efficient navigation of data. Semantic file systems address this concern by providing access based on context. However, there is still scope for a richer and rewarding experience of navigating data in a file system.

This paper introduces the concept of using association rule learning in a semantic file system. Association rule learning is a popular and well researched method for discovering interesting relations between variables in databases. By utilizing this, it becomes easy to understand the relativity between different data sets. Since most semantic file systems utilize a database to store and manage meta-information, algorithms such as 'Apriori' can be easily adapted for such applications.

## II. RELATED WORK

Over the years, organizing and retrieving information accurately and efficiently has attracted lot of attention. While few have been successful, a number of innovative implementations [2] have emerged. The idea of using a file's semantics as the means to categorize it has been around for quite some time. This section discusses the various implementations made in the field of semantic file system. An efficient implementation of keyword based searching was brought to the desktop by Google's Desktop Search [3] and Apple's Spotlight [4]. Both allow efficient and quick file retrieval based on keywords. They support many file types and have a simple interface which attracts a large number of users. However, both of them are limited to returning search results without any way to organizing contents. In addition, they do not provide any provision to the user for classification of data. This limitation prevented the user from having a personalized way to retrieve data stored by them.

Semantic systems depend on data stored inside the files rather merely relying on a file's attributes. Most implementations use common methodologies like content recognition [5], tagging [6], extracting metadata, etc. to categorize files by using various algorithms.

Each of these has the drawback that although they make organizing data easier, the task of searching relevant data is not tackled. Although Semantic file systems allow the user to browse data by context, but there is simply no provision to recognize the relativity between various contexts.

Using data mining in the semantic-aware file-system, one can easily come up with relations that can help the user get to related data quickly and efficiently.

## III. Semantic File-System

A semantic file system [7] is a virtual file system that uses a database to store and manage file meta-information. It is capable of performing its own interpretation of common file system functionality.

The database is utilized to store tags which are virtual directory entries used to categorize files. Files can be tagged with any number of contexts as long no file system rule is violated. The system extracts metadata into tags and stores it in a relational database. These tags can be file attributes such as size, type, name etc. as well as extracted metadata such as author, content title, etc. Categorizing files by metadata allows linking a file in multiple ways while being able to search it using its context. This enables the users to find relevant information in as few searches as possible.

Virtual directories are used to display stored files in a semantic organization. Search results are displayed through dynamically created listings, which correspond to semantic segregation. The entire implementation is based on a virtual file system which manages only the data organization. The underlying file system takes care of storage. This allows it to be ported in future to any file system.

## IV. Apriori Algorithm

In data mining, association rule learning is a popular and well researched method for discovering interesting relations between variables in large databases. It is intended to identify strong rules discovered in databases using different measures of interestingness [8]. Based on the concept of strong rules, Rakesh Agrawal [9] introduced association rules for discovering regularities between products in large-scale transaction data recorded by point-of-sale (POS) systems in supermarkets. For example, the rule found in the sales data of a supermarket would indicate that if a customer buys onions and potatoes together, he or she is likely to also buy hamburger meat. Such information can be used as the basis for decisions about marketing activities such as, e.g., promotional pricing or product placements. In addition to the above example from market basket analysis association rules are employed today in many application areas including Web usage mining, intrusion detection and bioinformatics [10]. As opposed to sequence mining, association rule learning typically does not consider the order of items either within a transaction or across transactions.

Apriori [11] is a classic algorithm for frequent item set mining and association rule learning over transactional databases. It proceeds by identifying the frequent individual items in the database and extending them to larger and larger item sets as long as those item sets appear sufficiently often in the database. The frequent item sets determined by apriori can be used to determine association rules which highlight general trends in the database. This has applications in domains such as market basket analysis.

Apriori is designed to operate on databases containing transactions (for example, collections of items bought by customers, or details of a website frequentation), where the algorithm attempts to find subsets which are common to at least a minimum number of the item sets.

Adapting the same for a semantic database, we can utilize apriori to show us files which are common to various tags (or directories). Having a cut-off percentage, the apriori results can determine whether two tags are related by looking at files common to them. This way, the user can be shown related information through links to tags obtained from the apriori algorithm.

This example demonstrates a music player like capability where the user is suggested similar music artists based on his organization of files. This is similar to a music player application or on-line service which uses data collected from hundreds of users to show similar tags. The in-built metadata in a music file is also utilized to show similar artists by genre.

The user has various audio files tagged according to their metadata which results in the following setup.

| Tag Artists | Tag Rock | Tag Party |
|---|---|---|
| Linkin Park | Linkin Park | Enrique |
| Coldplay | Coldplay | Pitbull |
| Metallica | Metallica | Rihanna |
| Enrique | Green Day | |
| Pitbull | | |
| Rihanna | | |
| Green Day | | |
| Taylor Swift | | |

Tags based on Artist and Genre

The user then creates new tags to form playlists based on these artists. The user manually (or through some process) adds artists under the new tags created to form the following playlists.

| Tag MyBest | Tag Evening | Tag Travel |
|---|---|---|
| Linkin Park | Linkin Park | Coldplay |
| Coldplay | Coldplay | Metallica |
| Enrique | Metallica | Green Day |
| Green Day | Enrique | |
| | Pitbull | |
| | Rihanna | |
| | Green Day | |
| | Taylor Swift | |

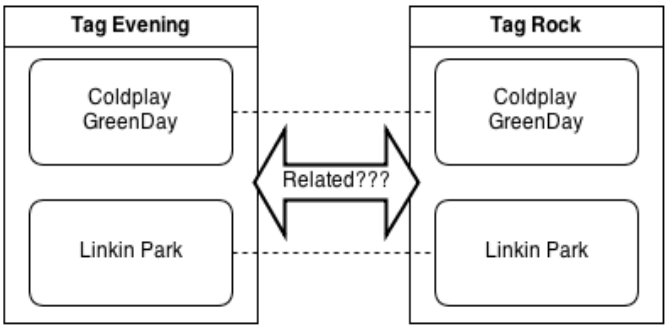
Playlists(Tags) based on user's preferences

Running an instance of the apriori algorithm on these item sets, we can form the following association rules -

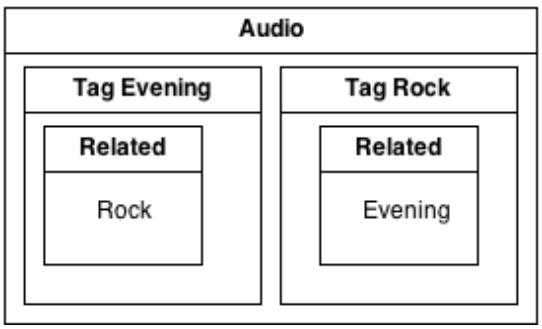
**1. Common occurrence**


Whenever the tag of files tagged as ‘Coldplay’ occurs, the tag 'Green Day' is also in the same item set. Therefore, Coldplay and Green Day can be safely assume to be related. Thus, whenever the user wants to create a new play list, and selects one of them, the system can suggest the other to be included as well.

**2. Deducing Related tags**




The tags ‘Evening’ and ‘Rock’ have some common artists. So are they related to each other? It may be possible, since Rock and Evening both can play some common files. Therefore while browsing; a link to this related information can be shown. For example use the tag ‘Related’ to show related tags.




When the user browses the tags Evening and Rock under Audio, related tags (Evening – Rock) can be shown as linked. This allows the user to play all related files from one location.

## V. Extending model for all file-types

This example demonstrated how association rule learning can be useful and helpful to the user wishing to play related files. Similarly it can be easily demonstrated that the model can be extended to work with almost any file-type as long as it has been tagged properly. For the apriori algorithm, the file-type is of no special significance. As long as the semantic database keeps meta-information about the file, the apriori can form association rules by scanning the database.

The association rules thus obtained can be utilized to show related files and directories to the user while navigating the semantic file system. Extending the example to include all files, the Apriori algorithm can be utilized to include the following:

1. Give suggestions when the user is tagging a file. For example a user tagging a certain document as ‘Project’ can be suggested to also tag it under ‘Confidential’ as most files under ‘Project’ are tagged as ‘Confidential’.

2. Show related files and tags while browsing. Consider a user is browsing through all pictures tagged under ‘Monuments’. Suggested pictures can include cities where the pictures were taken or documents tagged describing those Monuments.

## VI. Conclusion And Future Work

Combined with the semantic file system's ability to store data by context, associations rules help improve the user experience by providing a richer experience in browsing data. This is achieved by providing links to directories which are determined to be related using algorithms such as the apriori. Users can navigate to related directories using these links. This allows the users to browse through related data without the burden of long or incomprehensible pathnames.

## References

[1] Jon. B, "DuckDuckGo: A New Search Engine Built from Open Source"

[2] Mangold. C, *A survey and classification of semantic search approaches*, Int. J. Metadata, Semantics and Ontology, Vol. 2, No. 1, 2007, Page(s): 23-34.

[3] Google Desktop Search, *http://googledesktop.blogspot.in*

[4] Apple Spotlight, *http://developer.apple.com/macosx/spotlight.html*

[5] Gopal. S, Yang. Y, Salomatin. K, Carbonell. J, *Statistical Learning for File-Type Identification*, 2011 10th International Conference on Machine Learning and Applications, Page(s): 68-73.

[6] Bloehdorn. S, Grlitz. O, Schenk. S, Vlkel. M, *TagFS - Tag Semantics for Hierarchical File Systems*, In Proceedings of the 6th International Conference on Knowledge Management (I-KNOW 06), Graz, Austria, September 6-8, 2006.

[7] Mohan.P, Venkateswaran.S, Raghuraman, Dr.Siromoney.A, *Semantic File Retrieval in File Systems using Virtual Directories*. Proc. Linux Expo Conference, Raleigh, NC, Page(s): 141-151, May 2007.

[8] Piatetsky-Shapiro, Gregory; and Frawley, William. J, *Discovery, analysis, and presentation of strong rules*, Knowledge Discovery in Databases, AAAI/MIT Press, Cambridge, MA (1991).

[9] Rakesh. A, Tomasz. I, Arun. S, *Mining Association Rules between Sets of Items in Large Databases*, SIGMOD Conference 1993, Page(s): 207-216.

[10] Chang. K, Perdana. I, Jain. M, Kartasasmita. I, Ramadhana. B, Sethuraman. K, Le. T, Chachra. N, Tikale. S, *Knowledge File System - A principled approach to personal information management*, 2010 IEEE International Conference on Data Mining Workshops, Page(s): 1037-1044.

[11] Rakesh. A, Ramakrishnan. S, *Fast algorithms for mining association rules in large databases*, Proceedings of the 20th International Conference on Very Large Data Bases, VLDB, Page(s): 487-499, Santiago, Chile, September 1994.