International Journal of Advanced Computer Research (ISSN (print): 2249-7277 ISSN (online): 2277-7970)
Volume-2 Number-4 Issue-6 December-2012

Data Preparation for Web Mining – A survey

Amog Rajenderan
Department of Computer Science Rochester Institute of Technology Rochester, NY, USA

Abstract

An accepted trend is to categorize web mining into three main areas: web content mining, web structure mining and web usage mining. Web content mining involves extracting details/information from the contents of webpages and performing things like knowledge synthesis. Web structure mining involves the usage of graph theory to understand website structure/hierarchy. Web usage mining involves the mining of useful information from things like server logs, to understand what the user does while on the internet. This paper is intended to be a survey paper of recent papers that deal with cleaning and preparing the data that goes into the three types of mining mentioned earlier.

Keywords

Data Mining, Data Management, Data Cleaning, Preparation.

1. Introduction

The internet and the World Wide Web has become the most used method to create, consume and disseminate information in today's world. The web is also huge, constantly growing, dynamic and diverse. Dealing with this kind of unstructured data gives out huge scalability, complexity and temporal issues. One of the ways in which people would like to use the internet, is to fund relevant information. They could be web service providers, data analysts, or regular users. Once relevant information has been gathered, some people would like to create new information of that data. This kind of data could be used to further a business, or to make a standard user's internet experience better. Therefore this leads us to want to utilize this newly constructed data to allow personalization and perhaps added learning about users. All the above fall directly into the domain that is web mining.

Web mining allows e-commerce to bolster itself through personalized marketing, something that was not possible during the days of television, billboard or radio advertising. Web mining also has military applications such as anti-terrorism, threat identification and identification of illegal activities. Corporations can create better relationships with their customers by giving them exactly what they want, thus improving their businesses and retaining more customers. Targeted pricing and discounts will also boost their profitability, and all of these are just a few of the benefits afforded by web data mining.

While the benefits are great, most web mining will be completely useless if done on plain extracted data. Most web data is unstructured, messy, full of irrelevant logs and unstable. To prevent the happening of 'garbage in, garbage out', dirty data must be cleared out, and this is where web cleaning comes in. A majority of the work in the entire web mining process actually takes place in the data cleaning step. Once the data has been reasonably cleaned, the application of mining algorithms becomes relatively simple, giving out much better results.

This particular paper is a survey paper of various other papers that deal with web mining. This has a particular focus on how each of those papers went about cleaning their data before proceeding to mine it. The sections deal with cleaning done in the three traditional web mining areas: web log/usage mining, web content mining, and web structure mining.

2. Cleaning in web log mining

Munk et al [6] in their paper describe how the preparation of data is the most time consuming part of web log mining. The data used are extended transaction data that determines user behavior patterns via tracking IP addresses, browsing agents and pages visited. This kind of data is only useful when it has been thoroughly cleaned and prepared. Their steps include:

- Cleaning out redundant unnecessary data. Example: The logs caused by web crawlers, that access the data very differently from normal users do. (Web crawlers cache the web for the use of search engines).

- Identifying sessions/users, to identify different users behind the same computer or IP address. This can be done via say, a delimited time window.
- Another part of preparing this data is reconstruction of data when a user has used a back button. This kind of activity is not recorded by the browser, and web logs will show a user to have gone from page 1→2 without there having been any hyperlink from page 1 to 2. The sitemap is therefore used to reconstruct the possible path the user could have taken by hitting the back button to get to the aforementioned page.

This paper concludes by saying that the advantage of web log analysis vs. something like a survey is that the user does not know that he is the object of investigation. The disadvantage is that most of the effort goes into the preparation of the data, and in that, most of the work is done in reconstruction of behavior, since over 50 percent of accesses are done via backward access (hitting the back button).

Mobasher et all [4] in their paper describe preparation as "consists of converting the usage, content, and structure information contained in the various available data sources into various data abstractions". They say that the practical difficulties in preprocessing are great, and are constantly changing along with changes in web technology. Another difficulty is the incompleteness of the available data. His major tasks are cleaning, user identification, session identification (similar to the previous paper), pageview identification and path completion.

Here cleaning involves site specific activities likes log merging, removing accesses to pictures or other multimedia etc. Identifying sessions depends a lot on the server technology used. For those sites that use cookies, session identification becomes much easier. Without cookies, other heuristics must be employed. Pageview identifying is deciding which access contributes to a page view. This is trivial when it comes to a single frame site, but for sites with multiple frames, the site structure plays a huge role in inferring pageviews. Path completion is very similar to the behavioral reconstruction described previously. The other content that was previously ignored is now taken into account during content preprocessing, and the images and scripts are converted into forms that can be actually used during the mining process. This mainly will consist of classification, and this is often an area of research in itself.

In [1] Cooley et al go into these facts in considerable detail. They describe cleaning as an initial way to eliminate irrelevant data. When a user accesses a website, several log entries are made since several components of the webpage are downloaded simultaneously on loading. It is obvious that a user will never (or) very rarely put in an access request for a single graphics image alone, since these are automatically downloaded in the html code. And since web usage mining deals exclusively with human behaviors on the internet, all log entries with the end suffix like gif/jpeg/png etc. can be deleted automatically, since these are almost certainly made by bots and web crawlers for caching purposes. This must be used with caution though, since certain websites such as those that have graphical archives may not fit this trend.

These authors assert that user identification is particularly hard because of things like local caching, firewalls and usage of proxies. A reasonable naive assumption is that even if the IP remains the same, changes in the browsing agent will represent a change in user. However I personally think this is incorrect, due to the fact that many people nowadays use multiple browsers during a single session. Another heuristic is to use the access log in tandem with a referrer log, and use the site topology to generate browsing paths, helping to identify different users.

Session identification allows for users to visit a website more than once, and its goal is to separate the accesses into various individual sessions. The authors say the simplest way of carrying this out is via a page timeout, say a 30 minute window. Other researchers have established 25.5 minutes as a good timeout window based on data analysis.

Path completion, according to [1] essentially tells us what has been already covered on back button usage and reconstruction. In addition, they talk about an algorithm that utilizes timestamps and the assumption that any visit to a page already seen is a visit to an auxiliary page. They state "the average reference length for auxiliary pages for the site can be used to estimate the access time for the missing pages."

The last part in their preprocessing pipeline is formatting, where once all the previous steps have

been applied onto the server logs, a final preparation module is used to properly format the data to suit the type of mining that will take place. An example that the authors in [1] gives is: temporal information not being needed when mining for association rules, and so a final association rule module will strip out all the time data for all the references.

In 'Advanced data preprocessing for intersites Web usage mining' [7], D. Tanasa et al define their preprocessing as a four step process, consisting of:

- Data fusion
- Data cleaning
- Data structuration
- Data summarization

Data fusion involves the merging of several web logs that are retrieved from various servers, as well as the website structure or maps. These log files are joined, and then anonymized for privacy reasons. The authors present an algorithm to do this, and can be read in detail in [7]. Once the logs are joined, the data is anonymized by encrypting all the available host names and IP addresses. The original host name is replaced by a custom identifier that holds information about the domain extension. These parameters can be used/replaced later in the process, and this allows the log files to be shared without fear of privacy violations or leaking of sensitive data.

Data cleaning in [7] is quite similar to previous methods, where useless log requests are removed, along with requests concerning non analyzed resources (images, multimedia etc.). Data cleaning also identifies web robots/crawlers and removes their log data. This process becomes very useful when dealing with popular websites. Often, these sites gather logs in the size range of hundreds of gigabytes in an hour, and manipulating this kind of data is extremely difficult. By filtering out the data, the sizes are reduced drastically, sometimes by 40 or 50 percent.

Keeping or removing log requests to non-analyzed resources depends on the purpose of the site. In cases where the purpose is to support caching or prefetching, then these logs should not be removed. When dealing with web robots, a noticeable behavioral pattern can be used to identify their presence. A web crawler will begin to follow every single link present inside a page. Google's crawler does this periodically, every four weeks. Therefore the minimum number of requests a crawler puts in is equal to the number of hyperlinks present in that page.

In cases of pages that are not often visited by people, the crawler requests will often outnumber the actual human requests.

To identify robots, the author provides three heuristics:

- Look for all hosts that have requested “robots.txt”.
    - This is because all robots are required to access this text file, and under certain conditions specified within this text file, they will be forced to not crawl that site.
- Use an already available list of all known robots, said to be available at http://www.robotstxt.org/
- Simply guess if the user is a robot.
    - A useful way to do this is to calculate the browsing speed.
    - Say, Number of viewed pages/Session time. If this exceeds a certain upper limit/threshold, then the user is almost certainly not a human.

  Once robots have been identified, removing the logs is straightforward.

Data structuration is similar to previously discussed concepts of separating a log file into requests by a user, the user's session. Page views, visits, etc.

User Identification is done in most cases by the IP address and the user agent. In cases where a website requires a login, the computer will already store cookies dealing with the user's login, and this makes user identification much easier.

- In Page View Identification, the authors group page views via the following two step algorithm:
- If a request for a page p is in the log file, remove the log entries corresponding to all the embedded resources in that page p, and keep only that page.
- If a request for that page p is absent (perhaps due to a proxy), and only a few entries to its corresponding resources are there, then these are replaced with a request to the page p itself. The time of this request is set to the

minimum time recorded among all the resource access requests.

If the site map is unavailable, the views are instead calculated based on the time of requests. Among those requests made at the exact same time, only the first request is kept and the ones following are discarded. In Visit Identification, we try and split the user session data that has been obtained into separate visits. The provided heuristic details:

- H_{Page} uses a threshold D_t for measuring time gaps. Anytime a gap greater than D_t occurs, the next page is considered a new page visit.
- H_{Visit} uses a similar threshold for the whole visit.
- H_{Ref} utilizes the visit history and referrer. If no such referrer exists, a new visit has begun.

Episode Identification, a newer concept, are calculated by the "semantic distance between two consecutive pages or a page and the group of previously visited pages". Once this distance exceeds a certain value, a new episode begins.

Data summarization is the final step in [7]'s data preparation process. The files are first transferred to a relational database, and data generalization is applied onto the requested URLs, and "aggregated data computation" is applied in the visits and user session data.

The transferring into a relational database is done by attaching new tables to the existing ones, since they were designed differently during the initial preprocessing. An alternative to attaching tables is for the analyst to only select pertinent information when mining for data. To fill these tables we apply the aforementioned generalization and aggregated data computation.

Data generalization changes the URLs to reduce their numbers. An example is www.cs.rit.edu/~axr1545/project/index.html becomes www.cs.rit.edu/~axr1545/project/ which in turn can be reduced to www.cs.rit.edu/~axr1545/. This kind of generalization has been found [7] to greatly reduce data dimensionality, although a few drawbacks do exist to this. Pages that are very different may end up being grouped under the same URL by this method. An alternative to this syntactic generalization is semantic generalization.

Aggregated data computation calculates new parameters for visits, which can be utilized later. These parameters deal with statistical values. An example provided by the author, if the object under analysis is a user session, then the values that can be computed include:

- Total number of visits in that session
- Session time length
- Number of visits in a period of time (month, year, day)
- Percentage of requests made to a web server
- Another example, if the object under analysis is a visit, then we can compute
- Visit length in time, and visit length in number of page views
- Percentage of successful requests, percentage of bad requests
- Average time spent per page

3. Cleaning for web content mining

A lot of data, such as natural language data can be easily mined from the web. The training data collected from the web however, is naturally very messy, polluted by not only the existing linguistic errors, but also by things like html data, navigation bars, page headers, lists disclaimers and various advertisements. Evert in his paper [2] says that most search engines are not bothered by this kind of data, especially since they apply ranking algorithms, but this plays a major role in extracting web content as training data. In his paper, he presents a new tool, NCLEANER, which he submitted to the CLEANEVAL competition, and which he claims to clean web pages with 'state of the art' accuracy.

B. Mobasher et al. in [5] talks about effective content personalization, and how web mining can play a better role than traditional methods like collaborative filtering and content based filtering. His paper presents for both content and web usage mining, and hence the following will deal with both. Like the previously mentioned web usage mining, a lot of the steps involved in the preprocessing overlap. The author details user identification, then pageview identification and finally an additional transaction identification. This is explained as a final step done before pattern discovery. Support filtering is shown as a way to eliminate very low support or very high support pageview references, viz. those views that appear in either nearly nothing, or nearly everything. This is a noise elimination method.

Usage Preprocessing: [5] states that usage preprocessing results in a set of n records of pageviews appearing in the file where P = { p1, p2,..., pn}, each pageview being unique because of its associated URL, and also a set of m transactions by users, T= {t1, t2,.... , tm}, where each element is a subset of P. The authors say that to facilitate clustering, we can view these as an n dimensional vector, i.e. t = < w(p1,t), w(p2,t)..., w(pn,t)> where w are weights. These weights are determined through various methods, an example being the duration of the pageview, allowing us to capture user interest in a particular page.

The section following this is how [5] deals with actual content preprocessing. Their definition is "extraction of relevant features from text and meta-data." And meta data extraction is said to be of particular importance especially when dealing with product oriented pageview, and supposedly those pages that do not involve textual content. In the author's current implementation of their framework, they extract embedded data that is in the form of XML and HTML meta tags, as well as from the regular textual content of the pages. Appropriate weights are assigned to all of these, in order to be of use during association mining and similarity computations. [5] states "For features extracted from text we use a standard function of the term frequency and inverse document frequency (tf.idf) for feature weights as commonly used in information retrieval".

To go into detail, each pageview p is represented in a feature vector that is k dimensional. K here is the number of extracted features from the site. Each dimension will represent the weight of that feature in a pageview. The feature vector is defined as p = < fw(p, f1), fw(p, f2)...,fw(p, fk)> [2] , here fw() is the weight of the kth feature in the pageview set p. For all the extracted features from textual content, the weight is calculated via the normalized value of the aforementioned tf.idf value. In the end, both feature weights from meta data and feature weights from the textual data must be combined, and this is done via normalization, and organizing into an inverted file structure holding the dictionary of all the extracted features, and posted files for each feature describing pageviews of the page that the feature occurs as well as its weight. This is said to be conceptually equivalent to a feature-pageview matrix, where each column is a feature vector that corresponds to a pageview.

4. Cleaning web structure

Lan Yi in [8] describes a novel method to eliminate/reduce web page noise (irrelevant details like ads, navigation bars, copyright notices), and build a compressed structure tree, based on the webpage structure. This method is able to drastically improve results when it comes to web mining and web page clustering.

The basic concept is that most web pages tend to follow a common structure and fixed layout. The parts of a page that appear repeated in every page are most likely the noise that we are trying to avoid. The parts that are sufficiently different are most likely the actual content of the page. The compressed tree structure's purpose is to capture this common structure. Once the tree has been built, an importance measure is assigned to each word feature, and these weights are used directly during the mining.

Other work that is similar to [8] include Lin and Ho in 2002 [3], and Bar-Yossef and Rajagopalan in 2002. Lin and Ho use a concept similar to structure trees that are called informative blocks, however they work on two main assumptions.

- The system knows beforehand how a webpage can be partitioned into coherent blocks
- The system knows beforehand which blocks are the same in different web pages.

[8]'s system does not require the system to know all these details before hand, and can perform block classification and partitioning automatically. Other assumptions such as viewing a page as a flat collections of blocks, will work best only in certain domains like a news website, but will not work generally enough. Their assumptions are too strong.
Coming back to [8], one could argue that the Document Object Model (DOM) tree is sufficient to use without having to create a compressed structure tree, however the authors not that the DOM tree is insufficient for the task because it cannot represent the common structure of a set of webpages. They then go into the tree structure in detail, which this paper will give an overview of:

- A tag node is a node in a DOM tree, presented as (Tag, Attr) with tag being the tag name and Attr is the set of display attributes. This presentation style is denoted by S_T

- The basic information unit is called an element node, denoted by 5 components: tag, attribute, set of tag nodes in the original DOM tree, set of presentation styles and set of pointers to child nodes.
- Tag nodes of different Dom trees can only be merged after ensuring that the merged tag nodes are the same logical blocks from different web pages.
- The building of a CST is done by merging the DOM trees from top to bottom via:
- Root tag nodes are merged to form an element node.
- Styles of element node created are calculated. (common presentation styles are combined)
- Styles of element node created are calculated. (common presentation styles are combined)
    - For a pair of child element nodes, if the respective Tag and Attrs are the same, the textual contents are compared to see if merging is possible. If they are merged, a new element node is formed and inserted into the set of initial child element nodes.
    - This step ends when no pairs of child nodes can be merged
- If no child was created in the preceding step, stop. Else for each child node created, go to step 2.
- The weights are then assigned via a weighting policy. If an element node appears to have many types of presentation styles, then it is more likely important and will be assigned a high weight. Otherwise it is more likely to be noise and will be assigned a low weight. There is an even more in depth weighting policy described, which this paper will not go into.

The weights from this algorithm are directly used during web mining, and the experimental results that the authors describe show an effective improvement in eliminating noise.

5. Conclusion

Although web cleaning is very important, it appears as though very little work has actually been done in this field. Something similar was said by L. Yi in [8]. This holds even more true when it comes to data cleaning for web mining, as the progress through this field does not seem particularly ground breaking. This is evidenced by the fact that most of the concepts discussed in this paper seem to follow the same central theme, or the same general algorithm. There also seem to be little or no survey papers created for the field of web cleaning, and only a handful of papers in the field of web mining.

Another point of concern is that a majority of the work in this field was pioneered in the early part of this decade, and not much work has taken place since, except for incremental improvements on the existing cleaning standards.

Overall the idea this paper tries to present is that though data preparation is only the initial step in the vast process that is web cleaning, it clearly is the most difficult and most involved process among the lot.

Future work: The author hopes to see a lot more headway made in this field in the future, and though the past few years have not shown it, he is confident that a lot of work will be done in the near future, especially because of the dynamic nature of the internet.

References

[1] D. Cai, S. Yu, J. rong Wen, W. ying Ma, D. Cai, S. Yu, J. rong Wen, and W. ying Ma. “1 vips: a vision-based page segmentation algorithm”, 2003.

[2] S. Evert. “A lightweight and efficient tool for cleaning web pages”, Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC'08), Marrakech, Morocco, may 2008.

[3] S.-H. Lin and J.-M. Ho, “Discovering informative content blocks from web documents”, In Proceedings of ACM SIGKDD'02, pages 588-593, 2002.

[4] B. Mobasher, R. Cooley, and J. Srivastava, “Automatic personalization based on web usage mining”, Commun. ACM, 43(8):142-151, Aug. 2000.

[5] B. Mobasher, H. Dai, T. Luo, Y. Sun, and J. Zhu. , “Integrating web usage and content mining for more effective personalization”, Electronic Commerce and Web Technologies, volume 1875 of Lecture Notes in Computer Science, pages 165-176. Springer Berlin/Heidelberg, 2000.

[6] M. Munk, J. Kapusta, and P. Svec, "Data reprocessing valuation for web log mining: reconstruction of activities of web visitor", Procedia CS, 1(1):2273-2280, 2010.
[7] D. Tanasa and B. Trousse, "Advanced data preprocessing for intersites web usage mining", Intelligent Systems, IEEE, 19(2):59- 65, mar-apr 2004.
[8] L. Yi., "Web page cleaning for web mining through feature weighting", In Intl. Joint Conf. on Artificial Intelligence (IJCAI), pages 43-50, 2003.Shao Feng jing, Yu Zhong ," Principle and Algorithm of Data Mining ",Water Power Press, Beijing: China 2, 126-170, 2003.

Amog Rajenderan is a graduate student at the Rochester Institute of Technology. He graduated with a bachelors in engineering from SVCE, Anna University in Chennai, India. His interests include data visualization, computer graphics and high performance computing. He has also previously been involved with the Intel Corporation.