American University In Cairo

Seminar Report

Web Crawlers Detection

Author: Yomna ElRashidy
Supervisor: Ahmed Rafea

A report submitted in fulfilment of the requirements of Seminar 1 course for the degree of Masters of Science

May 2015
List of Figures

1.1 Web Crawler Methodology ............................................. 2
1.2 Robots.txt .................................................................... 3
1.3 Members list show table example .................................... 4
1.4 Detection Using Statistical Analysis ................................. 6

2.1 Decision Tree ............................................................... 10
Chapter 1

Literature Review

1.1 Introduction

A web crawler is a program that traverse the web autonomously with the purpose of discovering and retrieving content and knowledge from the Web on behalf of various Web-based systems and services. Such acts range from being beneficial such as indexing for search engines like Google or Yandex to malicious ones that attempts searching for vulnerabilities, spamming bulletin boards or undergoing DDOS attacks. It is very important for systems to distinguish different kinds of users effectively and accurately in order to be able to reduce the burdens on sites that is caused due to such activities. There are many reasons why it is important to identify visits by web robots and being able to distinguish them from other users, some of those reasons shall be discussed in section 1.2.

1.2 Why do we need web crawlers detection?

There are plenty of reasons, why is it so important for websites to be able to distinguish between different types of users effectively and accurately. Due to those reasons web sites managers find it necessary to detect web crawlers from all visitors in order to take proper measures to redirect the web crawlers or to stop responding to their requests.

- The amount of traffic caused by crawlers especially poorly designed ones may consume lots of network and server resources, resulting in performance degradation of websites since servers would be busy serving web crawlers requests rather than normal users.
• Content delivery web sites such as Wikipedia or ebay may not wish to serve incoming HTTP requests from unauthorized web crawlers.

• In human-user profiling using data mining of log files, requests originating from crawlers may provide misleading results regarding the navigational patterns of real users.

• Pay-per-click advertising can be seriously harmed by click fraud which involves among other things the unwilling or malicious repetitive ”clicking” on advertising links by Web robots.

• Detecting cyber attacks such as DDOS attacks that targets websites through messages flooding by robots.

1.3 Web Crawling Methodology

A Web crawler is one type of robot, or software agent. It starts with a list of URLs to visit and as the crawler visits these URLs, it identifies all the hyperlinks for the page’s image files, script files, CSS, etc belonging to the requested URL and all linked pages. It adds those hyperlinks to the list of URLs to visit which is known as the crawl frontier. Those URL’s are recursively visited according to a set of policies and their content is downloaded. The policies used in fetching takes into account reasonable measures for quality and freshness of downloads. Figure 1.1 shows graphically how do web crawlers work.

![Figure 1.1: Web crawling methodology.](image)
1.4 State Of The Art

There exists several methodologies used in detecting and distinguishing web crawlers behaviors from users requests. Each of those methodologies follows different techniques, which makes them vary in their speed of detection, whether or not they can be applied in real time as well as the amount of resources and calculations required to reach a classification. In the upcoming sections, an outline would be introduced about each of those existing methodologies.

1.4.1 Detection Using Robots Trap Strategy

This methodology relies on a very basic fact, that most authorized web crawlers follow which is accessing the Robots.txt file before crawling. The Robots.txt is a convention used to prevent web crawlers from accessing all parts of the websites which is otherwise publicly viewable. It specifies which areas are not to be accessible by the crawler as shown in figure 1.2.

A robot trap is then used to distinguish a normal user from a crawler, by trap linking one of the pages. Trap linking by adding an invisible link within one of the pages which points to page whose access is restricted within the robots.txt file. And hence in case it gets accessed, this would identify that the requesting entity is indeed a crawler rather than a normal user. However, this methodology is one of the oldest techniques and most web crawlers now, including non authorized ones obey the robots.txt file when requesting pages from the queue.

```plaintext
User-agent: *
Disallow: /site=
Disallow: /cite
Disallow: /54801ac.
Disallow: /gp/
Disallow: /audio.html/
Disallow: /houseads/
Disallow: /askhome/
Disallow: /t2opt/
Allow: /
Sitemap: http://www.thesaurus.com/thesaurus-sitemap/Sitemap.xml
```

Figure 1.2: Robots.txt File.
1.4.2 Detection Using Web Page Members List

This methodology relies on key differences point between a normal user and crawler’s requesting behaviour. A normal human user, on HTML document request, the browser analyses and requests all embedded and linked objects to the requested document, such as CSS, image/audio files, script files, etc in order to view the whole page to the user. This activity of the browser requesting the embedded objects occurs at once within 0-5 seconds of the initial request, where the total requests intervals never exceeds 30 seconds. However, the behaviour of a web crawler is different, on HTML document request, the crawler analyzes all embedded and linked objects to the requested document such as CSS, image/audio files, script files, etc. The crawler doesn’t request linked objects at once and some crawlers add them to waiting lists. The time interval for the requests that follow the main page request is greater than 30 seconds.

This methodology relied on this key difference between the behaviors. In this technique member lists are constructed for every page with all its linked objects such as css or java script files. Then an algorithm analyzes web logs data for every visitor and constructs ShowTable as shown in figure 1.3. Where this ShowTable identifies whether or not the requesting entity requested the objects that are linked to the main page within a time interval of 30 seconds or not. And according to that, an entity that is found to have all linked objects per pages are requested in higher intervals would be identified as a web crawler rather than a human visitor.

<table>
<thead>
<tr>
<th>url</th>
<th>NumberOfMember</th>
<th>ShowNumber</th>
</tr>
</thead>
<tbody>
<tr>
<td>index.htm</td>
<td>6</td>
<td>0</td>
</tr>
<tr>
<td>introduction.htm</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>myideas.htm</td>
<td>4</td>
<td>0</td>
</tr>
<tr>
<td>bigdog.jpg</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

Figure 1.3: Members list show table example.
1.4.3 Detection Using Web Log Analysis

This methodology relies on processing the data within the access logs in order to identify through the patterns in the log whether or not the requesting entity is a web crawler or not. This methodology follows the following steps in the identification,

- “Preprocessing log files” which pertains information about web access attempts such as client IP address, date and time, status code, etc.

- “Session identification”, done by grouping HTTP requests in the log files into sessions. Grouping based on IP address. Where sessions within a certain timeout period are grouped together, specifically 30 minutes, any request following the timeout is to be added as a new session.

- “Crawler identification” from identified sessions based on certain features found within a session:
  - Access of robot.txt, Web crawlers has to access the robot.txt before downloading any content from website.
  - Access of hidden links, Introducing hidden links that are not visible in the browser as a honeypot for web crawlers. sessions that access the hidden links are web crawlers.
  - Blank referrer with hit count, Crawlers initiate HTTP requests with an unassigned referrer, which is used for identification in conjunction with hit count exceeding certain threshold. **Since many browsers exclude referrer which might confuse normal user with a crawler.
  - Hit count, Hit count per a particular period, which is the number of HTTP requests during each session. If hit count exceeds a certain threshold, a web crawler is detected.

Using the values and knowledge found about the requesting entity through it’s sessions, it can then be identified to be a web crawler rather than a human user.

1.4.4 Detection Using Statistical Analysis

Detecting web crawlers in real-time using machine learning techniques. It utilizes an off-line, probabilistic web crawler detection system, in order to characterize crawlers and extract their most discriminating features based on statistical analysis to be used in detection. The algorithm works as follows, which includes characterization of crawler
sessions using off-line detection by access-logs analysis and session identification followed by extraction of session features to be used in the Bayesian network and learning the Bayesian network parameters which is then used in classification of sessions into crawlers or humans. Finally new Features gets extracted from the classified sessions. Statistical Analysis of those features in order to select the most discriminant ones to be used by the final real-time detection system. The figure 1.4 shows the starting features and the final most discriminant features used in classification in the last phase, and all the steps carried out within the algorithm in a graphical form.

1.5 Limitations Of Existing Methodologies

There are several limitations to the existing methodologies, including the following:

- Methodologies that rely on the fact that web pages usually has other linked resources (page’s member list) such as images, style sheets or scripts, may regard a human as a crawler if the web pages of a web site contain only plain text. If user uses simple browser or set the browser to not display anything but text.

- Methodologies that rely on session’s behaviour such as percentage of requests or time between requests, requires logging a certain number of requests per session to identify the requesting entity as a human or crawler.

- Methodologies that rely on the request logs for identifying the requesting entity, perform their analysis on log records in offline environments, does not detect crawlers on real time.

- Methodologies that rely on using statistical techniques for real time identification provides average precision of 86
Chapter 2

Using Decision trees for web crawlers detection

2.1 Approach Objective

As has been discussed in previous chapter, there are several limitations to the existing approaches in distinguishing web crawlers. In order to be able to limit those drawbacks of current approaches. Among those objectives, the following:

- Real time detection of web crawlers using machine learning techniques, with highest accuracy while minimizing the number of requests needed from a specific IP before detection.
- Detection results can be used for classifying web crawlers to malicious or ethical web crawlers on future work.

2.2 Decision Trees For Web Crawlers Detection

This approach is to involve several steps, which eventually helps us reach to the knowledge of whether or not a requesting entity is a web crawler or not, the following are the steps of the proposed approach,

- Preprocessing server logs
  - Session identification
  - Feature extraction
– Dataset labelling

• Building decision tree

### 2.2.1 Preprocessing server logs steps

In this section, the steps of the server logs preprocessing shall be described in more detail. This is the initial preparation for the proposed approach in order to prepare the data required for building the decision tree that would be used in the classification of requests.

#### 2.2.1.1 Session Identification

First of all, the session identification phase, which is the process of dividing a server access log into sessions, by grouping HTTP requests from same IP address and same user agent string together. Grouping requests together into sessions using timeout approach, with timeout interval of 30 minutes or when session continuity condition is violated.

#### 2.2.1.2 Features Extraction

The session identification is followed by feature extraction phase from the sessions identified in the previous phase, where the features extracted should include all of the following together in order to have better chances of building a decision tree with the most effective attributes forming the nodes closer to the root for faster and more accurate classification.

* Click rate, number of HTTP requests per session. Higher rates signals the presence of web crawlers.
* HTML-to-image ratio, the number of HTML page requests over number of image requests. Web crawlers usually ignore images.
* Percentage of 4xx response codes, usually higher for crawlers as there is higher chances of hitting an outdated or deleted page.
* Percentage of HTTP HEAD requests, indicates a web crawler.
* Percentage of unassigned REFERRER header of request.
* ‘/robots.txt’ file request. Indicates a web crawler attempt to access web pages.
• Standard deviation of request pages depth, SD of depth of all requested pages within a session. Deeper requests usually indicates a human user.

• Percentage of consecutive sequential HTTP requests, higher values indicate a human user due to requesting connected

• Percentage of night requests.

• Average time between requests.

• Standard deviation of time between requests.

• Members list files of page requests within session and average time between page request and linked files requests.

• Percentage of cookies enabled requests, crawlers don’t have cookies identified. content...

2.2.1.3 Dataset Labelling

Then, the final phase in the preprocessing of logs is the dataset labelling. Where each of the feature vectors extracted from the server log sessions gets labelled into either a human visitor or a web crawler.

The labeling is done using a log analyzer which uses known features for identifying web crawlers such as access to the robots.txt to label as well as according to the assigned user agents or HEAD requests.

All sessions labels will undergo a manual inspection by a human to ensure the correctness of the classification before training the decision tree.

2.2.2 Building the Decision Tree

Building the decision tree, which best fits the labelled feature vectors extracted, with the features representing the tree’s attributes.

Decision trees classify incoming traffic sessions by sorting them down the tree from root to leaf nodes. Where each node would question one of the features within the feature vector of the session. In the leaf nodes, session gets classified into human or crawler.

The expected shape of the decision tree would be as in figure
2.3 Hypothetical Output

The proposed approach is expected to address the limitations that exist within the current approaches, this approach shall be able to minimize the number of requests served before a session is to be identified as originating from a normal user or a web crawler, which makes it able to serve on real time environments as well as the number of features that are to be used in building the tree will prevent it from growing in large sizes and hence reducing the amount of time needed in processing the requests and identification.
Chapter 3

Bibliography

• Identification and characterization of crawlers through analysis of web logs Al-
giriyage, N. ; Univ. of Moratuwa, Moratuwa, Sri Lanka ; Jayasena, S. ; Dias,
G. ; Perera, A. 8th IEEE International Conference on Industrial and Information
Systems (ICIIS), 2013

• A Web Crawler Detection Algorithm Based on Web Page Member List Weigang
Guo ; Sch. of Electron. Inf. Eng., Foshan Univ. Foshan, Foshan, China ; Yong
Zhong ; Jianqin Xie 4th International Conference on Intelligent Human-Machine
Systems and Cybernetics (IHMSC), 2012

• PUBCRAWL: protecting users and businesses from CRAWLers regoire Jacob, En-
gin Kirda, Christopher Kruege, Giovanni Vigna Security’12 Proceedings of the 21st
USENIX conference on Security symposium,2012

• Real-time web crawler detection Balla, A. Dept. of Comput. Sci., Univ. of Cyprus,
Nicosia, Cyprus Stassopoulou, A. ; Dikaiakos, M.D. 18th International Conference
on Telecommunications (ICT), 2011

• A research on a defending policy against the Webcrawler’s attack Wei Tong ;
Sch. of Comput. Sci. Technol., Guizhou Univ., Guiyang, China ; Xiaoyao Xie
Anti-counterfeiting. 3rd International Conference on security, and Identification
in Communication, 2009. ASID 2009

• Securing web service by Automatic Robot Detection Park, Pai Princeton Univer-
sity, Kang Lee, Calo BMW Watson research centre 2006

• Unsupervised clustering of web sessions to detect malicious and non malicious
website users Stevanovic, Vlajic York University, Canada 2011
• Discovery of Web Robot Sessions based on their navigational patterns Tan, Kumar
  University of Minnesota, 2002 content...