Library Hi Tech

Web robot detection in scholarly Open Access institutional repositories

Joseph W. Greene

Web robot detection in scholarly Open Access institutional repositories

Joseph W. Greene
James Joyce Library, University College Dublin, Dublin, Ireland

Abstract

Purpose – The purpose of this paper is to investigate the impact and techniques for mitigating the effects of web robots on usage statistics collected by Open Access (OA) institutional repositories (IRs).
Design/methodology/approach – A close review of the literature provides a comprehensive list of web robot detection techniques. Reviews of system documentation and open source code are carried out along with personal interviews to provide a comparison of the robot detection techniques used in the major IR platforms. An empirical test based on a simple random sample of downloads with 96.20 per cent certainty is undertaken to measure the accuracy of an IR's web robot detection at a large Irish University.
Findings – While web robot detection is not ignored in IRs, there are areas where the two main systems could be improved. The technique tested here is found to have successfully detected 94.18 per cent of web robots visiting the site over a two-year period (recall), with a precision of 98.92 per cent. Due to the high level of robot activity in repositories, correctly labelling more robots has an exponential effect on the accuracy of usage statistics.
Research limitations/implications – This study is performed on one repository using a single system. Future studies across multiple sites and platforms are needed to determine the accuracy of web robot detection in OA repositories generally.
Originality/value – This is the only study to date to have investigated web robot detection in IRs. It puts forward the first empirical benchmarking of accuracy in IR usage statistics.

Keywords Usage statistics, Institutional repositories, Open access, Detection, Downloads, Web robots

Paper type Research paper

1. Introduction

Usage metrics are commonly used in library and information service environments to assist with decision making such as journal purchasing, collection building, and item deselection, and to demonstrate the overall value of the services themselves. Scholarly Open Access (OA) repositories, freely accessible full text repositories of scientific and scholarly publications, are one such service within the higher education and research information sector. Beginning around 1991 with arXiv.org, the electronic pre-print archive of papers in physics and similar subjects (Cornell University Library, n.d.), the number of OA repositories has grown to more than 4,000 worldwide in 2015 (University of Southampton and EPrints.org, n.d.). Many of these repositories are hosted locally by universities for self-archiving by the academic and research staff of those institutions and are known within the community as institutional repositories (IRs).

As with other information services, OA repositories often collect usage statistics for the items they host, typically as full text download counts. Opinions on download

The author would like to thank Paul Needham (University of Cranfield and IRUS-UK) and Stefan Amshey, Ann Connolly, and Jean-Gabriel Bankier (BePress Digital Commons) for invaluable discussions and suggestions on the draft of this paper.

Library Hi Tech
Vol. 34 No. 3, 2016
pp. 500-520
© Emerald Group Publishing Limited
0737-8831
DOI 10.1108/LHT-04-2016-0048

statistics are somewhat divided, with some arguing that they are problematic and unhelpful (Cornell University Library, n.d.), while others make free use of download statistics, ranking papers, and even authors, distributing them monthly to participants, and advertising them broadly to the public (Gordon and Jensen, n.d.; Zimmerman and Baum, n.d.). Download statistics have even been shown under certain conditions to be predictors of future citations (Brody *et al.*, 2006), arguably the most important metric for scholarly and scientific research publications.

Regardless of which stance one takes, any data used as a metric or simply publicised for promotional purposes must be accurate in order to be useful and credible. A great challenge to this in any web environment is the use of web robots, operated by search engines and comment spammers alike, and accounting for between 8.51 and 32.6 per cent of web traffic (Doran and Gokhale, 2011). Robot traffic can vary widely depending on the type of website, with a study on the internet archive finding as much as 93 per cent of requests attributable to robots (AlNoamany *et al.*, 2013).

Given the importance of accurate usage statistics, the sizable and widely variable impact of web robots, and the complexity of detecting them, we endeavour to answer the following questions: What techniques are commonly used for web robot detection? How do the main IR software packages implement web robot detection out-of-the-box? We then describe and test a web robot detection technique used in practice by an OA IR at a large Irish University and discuss an effective and practical approach to web robot detection for repositories that takes advantage of the theoretical models.

2. Web robot detection techniques

A close review of the existing literature on web robot detection yielded ten individual studies (Tan and Kumar, 2002; Geens *et al.*, 2006; Huntington *et al.*, 2008; Duskin and Feitelson, 2009; Stassopoulou and Dikaiakos, 2009; Doran and Gokhale, 2012; AlNoamany *et al.*, 2013; Song *et al.*, 2013; Lamothe, 2014; Zabihi *et al.*, 2014) and one overview/review article (Doran and Gokhale, 2011) that describe and test the main techniques and data used in web robot detection. Table I lists 23 distinct variables used in these studies, categorised here according to a simplified version of the schema proposed by Doran and Gokhale (2011). While the majority come from the field of computer science, three studies were found that focus on scholarly information systems (Bollen and Sompel, 2006; Huntington *et al.*, 2008; Lamothe, 2014).

None of these studies benchmark detection techniques used in an OA repository, though Huntington *et al.*'s (2008) research on an OA journal is very closely related in terms of the content. The technique of investigating outliers in library e-resource usage data proposed by Lamothe (2014) is similar not only in content, but also in terms of the technique, which is nearly identical with one of the techniques used by the repository investigated in this study.

Each study presents a different method for analysing the data, from matching data in the server logs against known robots (Huntington *et al.*, 2008) to complex machine learning techniques (Stassopoulou and Dikaiakos, 2009; Tan and Kumar, 2002). What is immediately clear is that no method is capable of accurately detecting all robots visiting a given web server. The stated goal of robot detection becomes to detect the highest percentage of all robots (recall) with the lowest number of false positives (precision), that is, capturing as many robots as possible while labelling the fewest number of human sessions as robots (Geens *et al.*, 2006). Table II summarises the recall, precision, and *F*-score (harmonic mean of recall and precision) achieved in a number of studies. Recall ranges between 0.85368 and 0.9751, precision between 0.82 and 0.95, and the *F*-score between 0.84466 and 0.94.

Simplified Doran and Gokhale (2011) classification	Data used in robot detection	Number of studies
Syntactic log analysis	User agent string	6[a,d,e,i,k]
	robots.txt access	5[a,b,e,h,i]
	List of known robot IP addresses	3[d,e,h]
	Time of request (night time)	3[d,g,i]
	Empty referrer field	2[d,i]
	Use of HEAD method	2[d,i]
	Reverse DNS name lookup	2[e,h]
	Trap file	1[j]
Traffic pattern analysis	Rate of requests	6[a,c,e,h,k]
	Web page components	5[h,k]
	Volume of requests	5[c,e,f,g,k]
	Duration of session	3[e,h,k]
	Interval between requests	3[c,d,i]
	Per cent image requests	3[d,h,i]
	Resource type requests	3[b,i,k]
	Image:html ratio	2[a,h]
	Multiple IP addresses used in a single session	2[g,i]
	User agents per IP address	2[a,g]
	Width of traversal	2[h,i]
	Absence of back-and-forth navigation	1[j]
	Depth of traversal in the URL space	1[i]
	Per cent 304 response codes	1[j]
	Per cent 4xx response codes	1[h]
	Per cent GET request	1[i]
	Per cent PDF request	1[h]
Turing tests	CAPTCHA	1[k]
	Key clicks	1[k]
	Mouse movements	1[k]

Sources: [a]AlNoamany *et al.* (2013); [b]Doran and Gokhale (2012); [c]Duskin and Feitelson (2009); [d]Geens *et al.* (2006); [e]Huntington *et al.* (2008); [f]Lamothe (2014); [g]Song *et al.* (2013); [h]Stassopoulou and Dikaiakos (2009); [i]Tan and Kumar (2002); [j]Zabihi *et al.* (2014); [k]Other studies: Doran and Gokhale (2011)

Table I.
Data commonly used in robot detection

	Recall	Precision	*F*-score
Doran and Gokhale (2012)[a]	0.85368	0.83596	0.84466
Geens *et al.* (2006)[b]	0.9751	0.8935	0.932518
Kwon *et al.* (2012)	–	–	0.94
Stassopoulou and Dikaiakos (2009) (highest)[c]	0.86	0.95	0.903
Stassopoulou and Dikaiakos (2009) (mean)[d]	0.886	0.864	0.8706
Tan and Kumar (2002)[e]	0.95	0.82	0.880226

Notes: [a]Mean of recall, precision, and *F*-score of the five results listed in the study shown here; not reported directly by Doran and Gokhale. [b]Best combined method. *F*-score shown here is calculated from recall and precision as given in the publication using $F = 2(PR/(P+R))$ (Olson and Delen, 2008, p. 138); not reported directly by Geens *et al.* [c]Highest *F*-score reported by Stassopoulou and Dikaiakos (classifier 3). [d]Mean of recall, precision, and *F*-score of all five classifiers listed in the study shown here; not reported directly by Stassopoulou and Dikaiakos. [e]*F*-score calculated from recall and precision as given in the publication using $F = 2(PR/(P+R))$ (Olson and Delen, 2008, p. 138); not reported directly by Tan and Kumar

Table II.
Results of robot detection studies

In order to test a robot detection technique, researchers must first know "exactly which sessions in the log file were created by robots" (Geens *et al.*, 2006). This often involves checking and labelling the data manually (Huntington *et al.*, 2008; Geens *et al.*, 2006), though often, due to the size of the data set, robots in the test data are labelled automatically or semi-automatically (Stassopoulou and Dikaiakos, 2009; Tan and Kumar, 2002; Zabihi *et al.*, 2014) using reliable detection techniques different from the ones being tested. Doran and Gokhale (2012) discuss the strengths and weaknesses of manually versus automatically generated test sets. On the one hand, "there is a reasonable guarantee" that a manually generated test set is correctly labelled, however, manual labelling risks including only a narrow and non-representative selection of robots that would normally crawl the server (Doran and Gokhale, 2012, p. 9). On the other hand, they argue that there is no reason to assume an automatically generated test set is correctly labelled. To overcome the problem, the authors devise a method to test the test data (pp. 9-10).

This somewhat paradoxical situation is compounded by an interesting feature of some studies that Doran and Gokhale (2012, p. 9) point out and make efforts to avoid, that of the use of expert (human) opinion in robot detection and testing (Bollen and Sompel, 2006; Song *et al.*, 2013; Stassopoulou and Dikaiakos, 2009).

The fallibility and inconsistency of expert opinion, the paradoxical nature of technique testing, and the final results of various detection studies (as seen in Table II) suggest that absolute certainty in robot detection is very likely an unrealistic goal. Cast more positively, many studies clearly note that detection techniques should be multi-modal, making use of a variety of data and techniques to arrive at a best-possible result (Doran and Gokhale, 2011; Duskin and Feitelson, 2009; Geens *et al.*, 2006).

3. Robot detection techniques used in OA IRs

In our experience, IR content is most often discovered via the major search engines, a direct and positive result of robotic indexing. This supports the claim that "discovery happens elsewhere" (Dempsey, 2007) and has implications for web robot detection in OA repositories. On the one hand, repositories by nature wish to attract search engines in order to increase the visibility and discoverability of their content; however, they must present accurate usage statistics that have non-human usage filtered out. Since repositories cannot simply exclude robots altogether (and this would probably be impossible), they must develop practical web robot detection techniques that can stay apace of robot development.

To some degree the lack of discovery occuring within the actual repository site itself sets them apart from the detection studies listed above. Session data beyond the initial download request is limited or non-existent; a session in a repository often consists of a single direct bitstream (file) download with neither leading click-stream nor trailing lateral browsing. The only available trace in the logs may be limited to the date and time, the HTTP method and response code, and the IP address, user agent string, and referring website. It is not unusual for many of these fields to be left blank.

If this is generally true, then many of the robot detection techniques listed in Table I, such as requests for web page components, image:html ratios, and resource type requests are of limited or no use in repositories. The lack of interaction with the site rules out the use of real-time techniques such as mouse movements and key clicks. CAPTCHA could limit the discoverability of open resources since they are usually intended to deny access to robots. A list of robot detection techniques that remain relevant to repositories is put forward and the techniques used by the main OA repository systems are compared in Table III. Bearing in mind their mostly non-commercial nature, open source community-based development, and unique usage patterns, the web robot detection techniques used in these systems are described in detail below.

3.1 DSpace

DSpace, first released in 2002 (Smith *et al.*, 2003), is used by over 1,608 institutions and is the most used IR system worldwide (University of Southampton and EPrints.org, n.d.). DSpace collects usage statistics using Apache SOLR and has included web robot filtration since 2010 (Diggory and Luyten, 2015b).

DSpace has functionality to detect robots using three methods. It first checks the user agent string for each download and page view against a list of 235 known user agent patterns (regular expressions). Next it checks the IP address against lists of six large search engines' IP addresses, one list of 2,528 other known search engines' IP addresses, and one list of 48 robots not associated with search engines. Most of these IP address lists can be automatically updated via web queries. Finally, the detector does a reverse DNS name lookup for the fully qualified domain name and matches against a list of domains of known robots (Van de Velde and Diggory, 2015).

Unfortunately, robot detection in DSpace seems to have suffered a certain amount of neglect: at the time of this writing, the IP address lists have not been updated in nearly six years (since February 2010). The list of user agent strings was last updated in April 2015. The list of domain names includes only ten patterns so could be viewed more as a functionality or potential/experimental robot detection method rather than an actual method used in practice. There are notable omissions in both the IP address list and the user agents, for example Bingbot (Diggory and Luyten, 2015a). Though Bingbot would match against the regular expression "bot" in DSpace's agent list, this points to a key

Table III. Robot detection techniques used in institutional repository systems

	DSpace	EPrints	Digital Commons	Minho DSpace Statistics Add-on	IRUS-UK
Rate of requests			✓	✓[c]	✓
User agent string	✓	✓	✓	✓	✓
robots.txt access				✓	
Volume of requests		✓[a]	✓	✓[c]	✓
Interval between requests					
List of known robot IP addresses	✓[a]		✓	✓	✓
Empty referrer field					
HEAD method					
Reverse DNS name lookup	✓[b]				
Trap file				✓	
User agents per IP address			✓		✓
Width of traversal in the URL space				✓[c]	✓
Response codes other than 200 or 302			✓		
Value of referrer field			✓		
X-referred-by header			✓		
Country of origin			✓		
Top-level domain			✓		

Notes: ✓, technique used by the system in the corresponding column header. [a]See discussion in corresponding section for specific implementation of this technique; [b]only implemented nominally or experimentally; [c]data available as a configurable report for manual decision making problem with relying on lists of known robots: they must be kept up to date to be effective, furthermore, they are easily circumvented (Doran and Gokhale, 2011). In DSpace, robot detection is almost completely static and badly behaved robots will almost never be detected. The need for improvements to the usage statistics system has however been discussed as recently as 2014 (DSpace Community Advisory Team and Luyten, 2014).

3.2 EPrints

EPrints is the longest running repository platform, started prior to 2002 (EPrints.org and University of Southampton, n.d.). It is the second most prevalent repository platform, in use by at least 578 institutions (University of Southampton and EPrints. org, n.d.). EPrints includes a usage statistics module called IRStats 2 that includes web robot detection (Field, 2015).

IRStats filters downloads based on two principles. The first is a list of 960 user agent strings (regular expressions) of known robots or crawler software. If the user agent string recorded in the EPrints access database matches any of these, the download is not counted. The second filter checks how often a single IP address downloads distinct items; by default if it downloads an item more than once in a 24-hour period, only one download is counted towards that item for that period. The stated goal of this filter is "to detect so-called double-click downloads" (François, 2015).

The repeat download filter is an interesting application of the volume of requests technique. It is not robot detection in the strictest sense (and is not advertised as such) since it intentionally allows some downloads to be reported as legitimate despite having been (very likely correctly) detected as robot downloads. At least one robot download per item per 24 hours is allowed by the filter, and if the same robot downloads every item in a repository once in a single day, all of these downloads will be counted as legitimate. Still, the filter no doubt greatly limits the effect of robots that could only be detected this way, and does so without manual intervention.

Logically this method will result in a number of false negatives (unfiltered robot downloads). False positives (human downloads discounted as robots) are also a potential side-effect due to factors such as network address translation (NAT). This results in a forced trade-off between recall and precision: the shorter the timeout period, the more false negatives (reducing recall); the longer the timeout, the higher the false positives (reducing precision).

The NAT problem, where many users on one network appear to be using a single IP address, has been queried by at least one user of the EPrints system (Joint *et al.*, 2011), but only an empirical test can determine the impact of the repeat download filter's algorithm on the accuracy of download statistics in EPrints.

The user agent filter in EPrints is static and will not detect badly behaved robots. Since it is not intended to detect robots (though it probably does, and is a legitimate detection technique), the repeat download filter is a missed opportunity for basic machine learning, since it never records the IP addresses or user agent strings of the agents that it detects. This means that a robot that EPrints correctly identifies today (albeit accidentally) could be completely ignored tomorrow.

3.3 Digital Commons

Digital Commons is a hosted IR platform, with 400 participating institutions (Digital Commons, n.d.-c). All OA articles hosted on a Digital Commons IR are discoverable through a single system known as the Digital Commons Network, which consists of more than 1.5 million OA works (Digital Commons, n.d.-a).

Since Digital Commons is a centrally managed network of repositories, robot detection is carried out across a much larger data set than any single repository. The size of the data set affords a view of user behaviour that would not be possible at a local IR, and any rule applied to one repository's data is applied to all. This results in comparable COUNTER compliant download statistics across all sites (Digital Commons, n.d.-b).

The robot detection technique used by Digital Commons consists of a number of filters. Downloads from known robots declared in the user agent string are all discounted, as are download requests that result in a HTTP response code other than 200 or 302. The referrer field is checked for automatically generated URLs, for example a referring URL that is identical to the URL of the requested resource. COUNTER compliance rules are applied to reduce all downloads of a single item by a single IP address and user to one download whenever they occur within 30 seconds of the previous download of that item by the same IP address (S. Amshey, A. Connolly, and J.-G. Bankier, personal communication, December 2015-January 2016; COUNTER, 2015, p. 25).

Finally, a weighted algorithm designed in-house is applied in real-time based on five criteria, including overall activity from an IP address in the last 24 hours across all articles and repositories, requests coming from proxy servers (indicated by use of an x-referred-by header), the location of the download request, and whether or not the download is coming from a .edu domain. The fifth weighted criterion cross-compares the number of user agents used by an IP address with the number of item requests made by each of these IP/user agent pairs. Different agent strings in use by a single IP address that download similar numbers of items are an indication of algorithmic behaviour, which is weighted in favour of robot activity (S. Amshey, A. Connolly, and J.-G. Bankier, personal communication, December 2015-January 2016).

3.4 University of Minho Statistics Add-on for DSpace

The Minho Statistics Add-on for DSpace, first built in 2006, is an open source statistics system that integrates with and runs parallel to DSpace. The system was originally designed to promote the University of Minho's IR and show the worldwide usage of archived documents (Carvalho, 2010). The system also provides comprehensive workflow and administrative statistics.

The Minho Stats Add-on stores every bitstream download (PDF or other file format) in the DSpace database. The system takes a multi-faceted approach to robot detection including matching against a pre-populated list of 793 known agents, detecting accesses to a decoy web page, and accesses to the site's robots.txt file. The database contains related tables of IP addresses and user agent strings (in addition to the pre-populated agents list) that have been previously identified as robots through log analysis. Downloads found to be robots are labelled and discounted from the download figures presented to end users.

To label downloads, a robot detection script reads the server log file and checks each request to see if the IP address and/or the agents are in the database. If the exact agent string is found in the database, any new IP addresses using that user agent string are flagged as potential robots. If neither the IP address nor the exact user agent string is found, the script checks whether the agent matches the pre-populated agent list or if the request was for the decoy web page or robots.txt. If any of these conditions are true, the new IP/agent pair is recorded and all downloads from the flagged IP addresses from a given date forward are discounted from the download totals, pending a manual decision (Dantas and Miranda, 2012). Figure 1 describes this decision tree.

In addition to the robot detection script, the Minho Add-on provides an administrative interface that identifies IP addresses with an access frequency of less than one minute, with more than ten hits, and with a high number of sessions (Dantas and Miranda, 2008). Lists of the most frequent IP addresses can also be viewed for any period of time. These tools can be used in combination to make manual

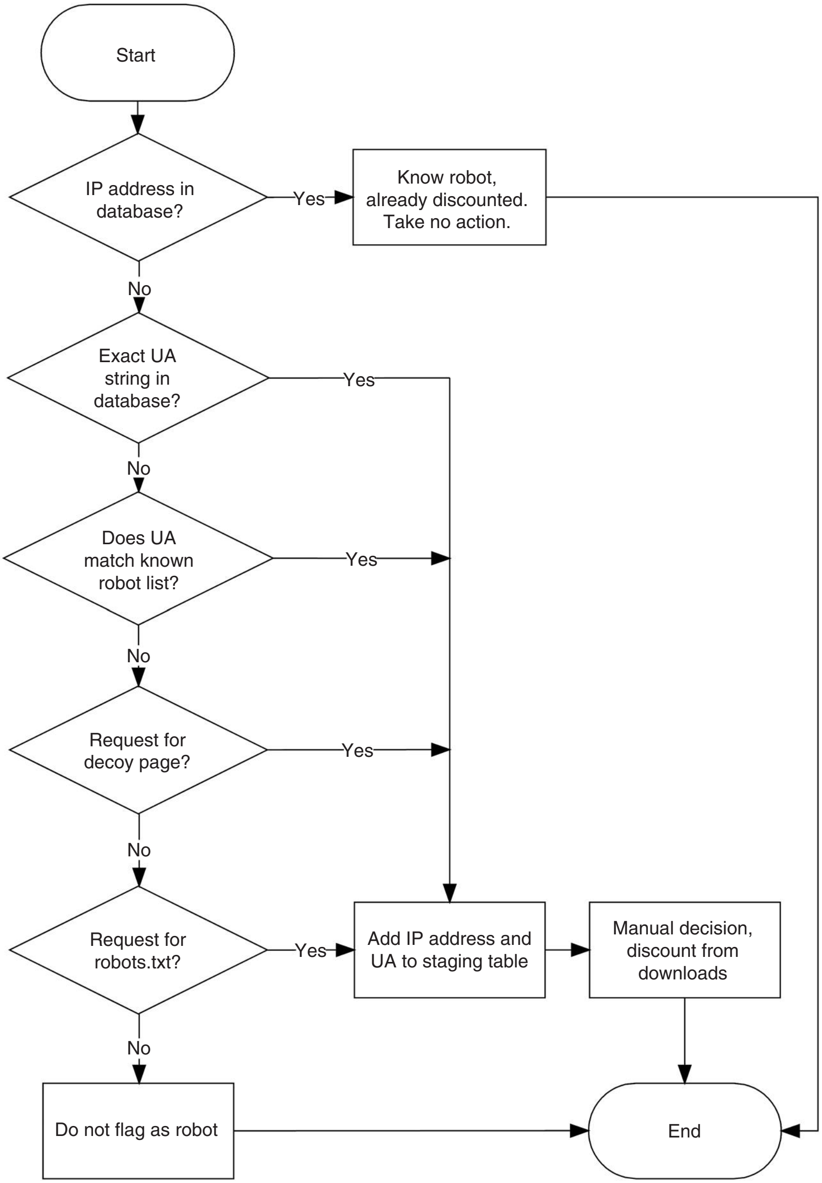

Start
IP address in database?
Yes
Know robot, already discounted. Take no action.
No
Exact UA string in database?
Yes
No
Does UA match known robot list?
Yes
No
Request for decoy page?
Yes
No
Request for robots.txt?
Yes
Add IP address and UA to staging table
Manual decision, discount from downloads
No
Do not flag as robot
End

Figure 1. Decision tree used by the robot detection script of the University of Minho DSpace Statistics Add-on

decisions on an individual IP address basis, which can then be added to the database using a Bash script.

There are two important points to note about the Minho robot detection script. Once an IP address is labelled as originating from a robot, it is added to a near permanent blacklist. If the IP address is reused later by a human user, it will still be discounted from the download totals. There is only limited functionality to remove an IP address from the blacklist. This is problematic at least in theory due to the constant recycling of IP addresses by the dynamic host control protocol (DHCP). This issue will be addressed further below.

Second, a user agent string that has been associated with a robot IP address will cause any new IP addresses using that user agent to be flagged as a robot. This includes legitimate human user agent strings forged by robots or an agent string belonging to a human user accessing the robots.txt file. The problem is mitigated somewhat by the fact that the match must be exact to the letter, so in the example UA string, "Mozilla/5.0 (Windows; U; Windows NT 6.1; en-US) AppleWebKit/534.16 (KHTML, like Gecko) Chrome/10.0.648.204 Safari/534.16", any change in the version numbering for the operating system, browser, rendering engine, etc., will prevent a match and the IP address will not be flagged as a robot. Since these numbers change constantly, the exact UA string for human users will most likely be phased out over time.

Despite the wide array of statistics and the attention paid to web robot detection, the Minho Add-on suffers from a severe lack of API and code-level documentation (Dantas and Miranda, 2008, 2012). Usage statistics are recorded in the database, as opposed to the much more efficient Solr index used by the DSpace native usage statistics system (Diggory and Luyten, 2015b); removing robots and re-aggregating the statistics can take many hours and is very costly in terms of system resources (CPU, RAM, and database connections). In our experience, there are countless coding errors that require many hours of debugging, especially in the SQL that produce usage statistics. Fortunately, the vast majority of SQL is encoded in a single XML file and can be manipulated without searching and recompiling the code.

3.5 Institutional Repository Usage Statistics UK (IRUS-UK)

IRUS-UK is a national service that aggregates, processes, and disseminates usage statistics from (to date) 91 IRs across the UK (IRUS-UK, 2015). IRUS-UK bases its processes on the COUNTER-PIRUS Code of Practice in order to provide comparable COUNTER compliant usage statistics to all participating repositories (Needham and Stone, 2012).

Like the Digital Commons Network, IRUS-UK is a large-scale service that applies web robot detection techniques centrally and thus consistently across a number of IRs. Both Digital Commons and IRUS-UK collaborate on the Usage Data Interest Group of the Confederation of Open Access Repositories (COAR, n.d.), and IRUS-UK are instrumental in forming the COUNTER Working Group on Robots (MacIntyre, 2014).

To detect robots, IRUS-UK uses a combination of the COUNTER robots list, consisting of 241 user agent patterns (regular expressions) of known robots (COUNTER, 2015), and a set of thresholds to limit the number of "overactive" IP addresses. These thresholds were initially set to filter out all downloads from IP addresses that made more than 200 downloads in a single day across all participating repositories (excluding known proxies), and most downloads from IP addresses that made more than 100 downloads in a day (again excluding known proxies) (IRUS-UK, 2013). The thresholds have since been reset to a maximum of 40 downloads per day from any IP address (P. Needham, personal communication, December 2015). IRUS-UK has also commissioned a study to investigate strengthening their web robot detection techniques (Information Power Ltd, 2013).

4. Benchmarking a robot detection technique used in an OA IR at a large Irish University

The IR at University College Dublin (UCD) collects usage statistics using the University of Minho Statistics Add-on for DSpace (version 4 for DSpace 1.8.2). Download statistics are visible in the item record of each paper and at each level of the collection hierarchy from a dedicated subsection of the website. Individual statistical reports are sent automatically each month to every author that has uploaded a paper to the repository. Reports are occasionally provided to schools and research centres in the university and are often used by them in formal quality reviews.

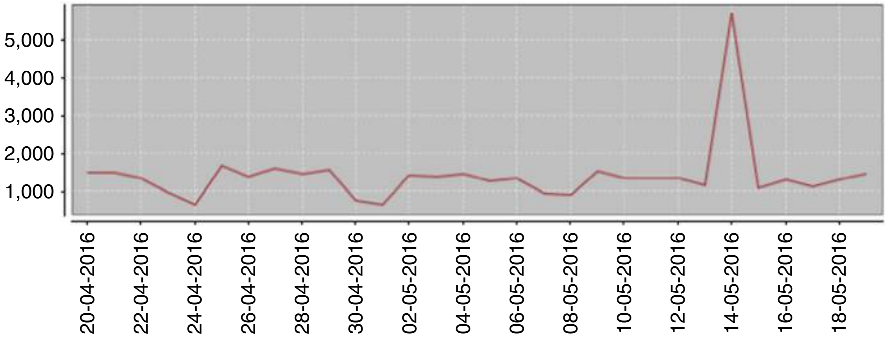
The importance of the usage statistics, both as indicator of the effectiveness and value of the service, and as a service itself, raises the following questions: How successful is the robot detection technique used at this repository? How accurate are the alleged human download statistics given to end users? The study may be able to shed light on the effect an IP address-based permanent blacklist (the DHCP problem) has in terms of human downloads classified as robots (false positives).

4.1 Description of the detection technique in use

At UCD, the Minho robot detection script runs nightly. New robots are labelled and aggregated in the download database weekly. Anything flagged by the robot detection script is assumed to be a robot. There are currently 49,556 IP addresses and 1,086 user agent strings flagged as robots in the database.

The decoy web page (trap file) feature is not currently used, so the detection procedure is essentially based on accesses to robots.txt, self-declared robots (in the user agent string) and all IP addresses and user agent strings previously determined to be robots. The date limit also is not used, so downloads from all flagged IP addresses are removed from the totals from the earliest date.

Outside the robot detection script a number of indicators are browsed monthly including the most downloaded items, the top 20 most frequent IP addresses (not already flagged as robots), and the daily download rate for the previous month. Any



Source: Taken from the University of Minho DSpace Statistics Add-on at University College Dublin

Figure 2. Outlier in daily download statistics for all items

notable spikes (outliers) are investigated by checking the logs and performing a reverse DNS name lookup on the IP address. Figure 2 shows an example of an outlier that would warrant investigation. Quite often, a number of "badly behaved" robots are found this way each month. This technique is very similar to that described by Lamothe (2014).

Through a combination of the automatic robot detection script and manual intervention including reverse DNS name lookup, rate, volume, and interval between downloads, the robot detection technique used at UCD takes into account up to nine of the 17 elements listed in Table III.

4.2 Study method

The majority of studies reviewed were based on session data for the identification of robot sessions (AlNoamany *et al.*, 2013; Duskin and Feitelson, 2009; Geens *et al.*, 2006; Stassopoulou and Dikaiakos, 2009; Tan and Kumar, 2002; Zabihi *et al.*, 2014). As mentioned, in our experience IRs typically have little or no real session data, instead usage is mostly limited to once-off file downloads directly from search engines. For this reason, rather than group downloads into sessions, we manually label a simple random sample of individual downloads to determine the recall and precision of the robot detection technique in use at UCD.

Two previous studies were quite similar to this study in that they were not based on session data. Huntington *et al.* (2008) tested a technique for an OA journal, a system and environment that is very similar to OA IRs. However, the purpose of the study was to measure the impact of robot usage on the journal rather than to propose or benchmark new detection techniques.

Song *et al.* (2013) investigated click-fraud on advertisements appearing on multiple websites to whose logs the authors had no access. Thus the data they analysed was limited almost exclusively to IP address, user agent, and click behaviour and did not include session data as would be created when browsing a website. Behavioural indicators included the total number of clicked ads (malicious users often have dense clicks on one advertiser), average clicks per advertiser (trusted users show high diversity), and total clicks per IP address among others. We adapted these indicators to our study in addition to a number of standard indicators as found in the literature. Table IV describes these data elements.

At the time the data were taken for the current study, the main download table in the UCD IR's database contained close to four million downloads. The Minho statistics package calculates a "relative value" for downloads of multi-file items by dividing each download by the total number of bitstreams attached to the item. For example, a journal article could be uploaded as two separate PDFs in one item: full text in one PDF and figures in a second PDF. A download of either PDF is counted as one download/two bitstreams, for a relative value of 0.5 downloads. There are currently 39 items (less than 1 per cent) with more than one bitstream in the UCD repository. For the purposes of this study, we ignore the relative value and consider a download of a single bitstream to be one full download.

A period of 24 months (9 November 2013 to 8 November 2015) was chosen to focus the study, bringing the total downloads to $N = 3{,}344{,}219$. Using an error of estimation bound $B = 0.05$ (for 95 per cent certainty) and an estimated ratio of robots to total downloads $p = 0.692896$, we determined that $n = 341$ downloads constitutes a representative sample for simple random sampling (Sheaffer *et al.*, 2006). The p value was estimated by dividing the total number of robots flagged by the robot detection procedures by the total number of downloads since recording began in 2009. This produced a more conservative p value, requiring 110 more random downloads than if we had estimated only on figures from the sample period. The 341 downloads were selected using the following SQL query:

```
select download_id, ip from stats.download where date between '2013-11-09' and '2015-11-08'
order by random() limit 341;
```

The data elements in Table IV were captured for each download using a combination of SQL queries, Bash scripts, and regular expressions (see Appendix 1).

To label each download, the flagged field was concealed from view and four passes were taken through the data. Self-declared robots were marked in the first pass. Reverse DNS lookup was performed on the remaining IP addresses in the second pass, where unambiguously robotic behaviour originating from hosting companies, cloud servers, and search engine companies were labelled robots. In the third pass, the sessions created by IP addresses exhibiting ambiguous behaviour were examined in the log files and the IP addresses checked against the Project Honeypot database (Unspam Technologies Inc., 2015). At this point, all downloads had been labelled and were compared against the flagged field to determine true and false positives and true and false negatives. All false positives and true negatives were then checked against the Project Honeypot database and examined in the log files, completing the fourth and final pass.

A number of assumptions were made while labelling the downloads. First, that IP addresses originating from search engine companies are always robots. This could potentially rule out genuine downloads made by employees of the company.

Table IV. Data used to manually label downloads and measure the detection technique's accuracy

Field	Description
IP	The IP address registered for this individual download
origin	Reverse DNS name lookup. Only entered where required
agent	Agent string(s) from logs – from all sessions for this IP during the period
agent_notes	Notes about this agent/download
robots_txt_access	Did this IP access robots.txt during the sample period?
HEAD method used	Did this IP use the HEAD method during the sample period?
dl_peak_this_item	Total downloads of this item by this IP during the period
dl_peak_any_item	Highest total downloads of any single item by this IP during the period
dl_site	Total downloads by this IP during the period
dl_per_day_peak	Peak downloads by this IP on a single day during the period
total_items_downloaded	Number of items downloaded by this IP during the period
first_seen	Date of first session during the period
last_seen	Date of last session during the period
indicator[a]	Indicator or sum of multiple indicators showing indicator(s) used to determine robot (single integer)
other_indicators	Description of any other indicators about robot/human behaviour
robot	Final decision based on manual checking whether this download is/is not a robot
flagged	Was this download flagged by the robot detection procedures?

Notes: [a]Indicators – 1: Agent name. 2: Reverse-lookup. 4: Downloads/frequency. 8: robots.txt access. 16: HEAD method. 32: Other indicators

IP addresses originating from cloud, rack space, IT infrastructure, and/or hosting companies were generally assumed to be from robotic agents. This raises a number of problems such as proxy servers, VPN users, cloud-hosted Ethernet users, and outsourced IT infrastructures. In general the download behaviour of these agents was sufficiently indicative, but in a many cases detailed log analysis, web search on agent/IP address pairs, and checking against Project Honeypot was required to make a final decision.

Lastly, it was assumed that downloads originating from end-user-oriented ISPs are typically from human users. This was occasionally overridden by abnormal download behaviour, for example a download from an end-user IP address provided by a major Irish ISP, with a user agent string Mozilla/5.0 (Linux; U; Android 4.0.4; en-ie; SonyST21i Build/11.0.A.4.22) AppleWebKit/534.30 (KHTML, like Gecko) Version/4.0 Mobile Safari/534.30, which downloaded the same item 4,016 times in a single day.

Despite the assumptions, we endeavoured to examine the data for each download holistically and in many cases our assumptions were overridden.

4.3 Results

Of the 341 random downloads between 9 November 2013 and 8 November 2015, 292 (85.63 per cent) were determined to be robots based on detailed manual checking. This finding is consistent with a report commissioned by IRUS-UK which found a minimum of 85 per cent of downloads to be from robots (Information Power Ltd., 2013). IRUS-UK had approximately 20 participating IRs at the time the data for their study was taken (IRUS-UK, 2015). Amending the robots/total downloads proportion using this ratio for the p value gives an error of estimation bound $B = 0.0380$, or 96.20 per cent certainty (Sheaffer *et al.*, 2006).

The first manual pass through the data produced 242 robots (82.88 per cent of the total robots found, 70.97 per cent of all downloads) that self-identified in their user agent string. Totally, 40 robots (13.70 per cent of all robots, 11.73 per cent of all downloads) were determined through a combination of origin, user agent string and behaviour in the second pass.

The status of 11 downloads was ambiguous after the first two passes. Seven came from hosting companies. One download was removed from the data set due to lack of data and replaced with a new random download from the period (without replacement of the previous 341 downloads). Of the remaining ten ambiguous downloads, eight were found to be robots through detailed log analysis, evidence from Project Honeypot, and web searching on the IP addresses, agents, and DNS name.

Checking false positives and true negatives (downloads where manual labelling could have missed a robot) against Project Honeypot resulted in five new robot downloads being identified. However, six IP addresses flagged by Project Honeypot as potential comment spammers were ruled out and were not labelled as robots.

True positives numbered 275 (80.65 per cent), false positives 3 (0.88 per cent), true negatives 46 (13.49 per cent), and false negatives 17 (4.99 per cent). This gives the robot detection technique in use at UCD a recall of 0.9418 with 0.9892 precision, an F-score of 0.9649, and overall accuracy of 0.9413 as defined by Olson and Delen (2008, p. 138).

4.4 Discussion

The unexpectedly high recall and precision measured here may be in part due to the fact that the ratio of robots to humans is very high. Doran and Gokhale (2011) put forward a theory that may go some way in explaining this high ratio when they describe Huntington *et al.*'s (2008) study on a scholarly OA journal. They suggest that "[t]he percentage of robots at this site may be high because only a small number of human users are interested in visiting an online scientific journal, whereas archival or scholarly article search services will commonly employ robots to visit the journal frequently to index or archive new articles" (Doran and Gokhale, 2011, p. 191). The percentage of downloads coming from robots in Huntington *et al.* is 32.6 per cent (approximately 1:2 robots to humans), whereas the percentage at UCD is 85.63 per cent (approximately 6:1 robots to humans).

Whatever the case may be, the downloads in this data set are dominated by one robot in particular: Googlebot accounts for 165 of the 341 downloads, a massive 48 per cent. To determine whether this is typical of repositories in particular or of web traffic in general would require further study. Treating Googlebot as an outlier and removing its downloads from the data, the recall, precision, and F-score are still encouraging at 0.8661, 0.9735, and 0.9167, respectively, with an error of estimation bound $B = 0.0528$ for 94.71 per cent certainty ($n = 176$). This is still within the range of the detection studies reviewed here.

Neither recall, precision, nor by extension the F-score, take into account the number of true negatives captured by a system (Powers, 2011). Since the studies in our review focused exclusively on robot detection and did not report true negatives, there is no way to compare the effectiveness of the system in our study with previous studies in terms of the accuracy of human usage. With this caveat in mind, the inverse recall, precision, and F-score in the present study are 0.9388, 0.7302, and 0.8214, respectively.

The ratio of robots to humans visiting a site determines the relationship between robot detection recall and precision and the precision of reported human downloads (inverse precision; see Appendix 2). Since the robot to human ratio is very high in our data, an increase of 0.01 in robot recall at the current precision would improve the precision of reported human downloads (inverse precision) by 0.03. Increasing robot recall by 0.05 would improve human download precision by 0.22. If the robot:human ratio of 85 per cent is generalisable, as the findings here and by IRUS-UK would suggest (Information Power Ltd., 2013), small improvements in any OA IR's robot detection could have significant effects on the precision and veracity of their usage statistics. Conversely, robot detection techniques that do not evolve with the advancement of web robots will result in usage statistics whose accuracy diminishes exponentially over time.

This study offers some findings towards the question of the DHCP problem. At 0.88 per cent, even the total number of false positives is very low, suggesting that permanently blacklisting an IP address (even if it could be reassigned to a human user at a later date) is not a major problem. The trade-off in precision is likely insignificant in comparison to the increase in recall.

Finally, this benchmarking study shows that a low cost and practical robot detection technique can produce remarkably high robot recall and precision. The technique consists of an extendable list of known IP addresses and user agent strings garnered from robots.txt accesses via an automated process. This is coupled with a simple way to visually locate unusual behaviour (outliers), allowing for manual robot detection.

Manual outlier checking, performed monthly at UCD, increases robot recall by 0.05; this in turn improves the reported human download precision by 0.14. Currently, neither DSpace nor EPrints support the ability to manually check for outliers. Adding this capability and/or the robots.txt/trap file feature, common to many studies and the Minho system, could significantly improve the accuracy of these systems' usage statistics.

While there is some evidence to suggest that the finding of 85 per cent robot downloads is generalisable for OA repositories, this study alone cannot make broad conclusions as to the accuracy of web robot detection in IRs. The study is performed on only one repository that uses a single, somewhat idiosyncratic web robot detection technique. Search engine optimisation and crawl behaviour influences (e.g. differing use of robots.txt, use or non-use of sitemaps.xml files) will likely change the effects of robots on repositories' usage statistics. Future studies adding to the breadth of empirical data, or larger studies across multiple sites and platforms can improve on these limitations.

5. Conclusion

Web robot detection is most successful when a variety of data and techniques are combined to achieve a best-possible result; no technique or combination of techniques will produce usage statistics that are completely free of robot downloads. This study has shown that very accurate robot detection at low cost in terms of computing resources and staff time is possible in community developed, free open source OA IR systems.

This is the first web robot detection benchmarking study performed on a scholarly OA IR to be reported. It differs from previous benchmarking studies in that the majority are experimental methods and do not test an operational robot detection technique, with the only exception being (Lamothe, 2014).

Given the high proportion of robot downloads being made in OA IRs, small improvements in robot detection increase the precision and veracity of reported usage statistics exponentially. It is well worth the effort in order to demonstrate the value of these services in a more transparent and trustworthy manner.

References

AlNoamany, Y., Weigle, M.C. and Nelson, M.L. (2013), "Access patterns for robots and humans in web archives", *Proceedings of the ACM/IEEE Joint Conference on Digital Libraries*, pp. 339-348.

Bollen, J. and Sompel, H.V.D. (2006), "An architecture for the aggregation and analysis of scholarly usage data", *Proceedings of the 6th ACM/IEEE-CS Joint Conference on Digital Libraries*, pp. 298-307.

Brody, T., Harnad, S. and Carr, L. (2006), "Earlier Web usage statistics as predictors of later citation impact", *Journal of the American Society for Information Science and Technology*, Vol. 57 No. 8, pp. 1060-1072.

Carvalho, J. (2010), "StatisticsAddOn – DSpace – DuraSpace Wiki", available at: https://wiki.duraspace.org/display/DSPACE/StatisticsAddOn (accessed 20 November 2015).

COAR (n.d.), "Interest group: usage data and beyond", available at: www.coar-repositories.org/activities/repository-interoperability/usage-data-and-beyond/ (accessed 12 December 2015).

Cornell University Library (n.d.), "arXiv.org e-Print archive", available at: http://arxiv.org (accessed 27 November 2015).

COUNTER (2015), "Release 4 of the COUNTER Code of Practice for e-Resources", available at: www.projectcounter.org/code_practice.html (accessed 3 December 2015).

Dantas, A. and Miranda, A. (2008), Stats-addon-2.1.1 – Version 2.1.1 for DSpace 1.5.1 (Computer software), University of Minho and KEEP SOLUTIONS, Braga.

Dantas, A. and Miranda, A. (2012), Stats Addon – Version 4 for DSpace 1.8.2 (Computer software), University of Minho and KEEP SOLUTIONS, Braga.

Dempsey, L. (2007), "Discovery happens elsewhere", available at: http://orweblog.oclc.org/discovery-happens-elsewhere/ (accessed 10 June 2016).

Diggory, M. and Luyten, B. (2015a), "DSpace configuration files (Computer software)", available at: https://github.com/DSpace/DSpace/tree/master/dspace/config/spiders (accessed 6 December 2015).

Diggory, M. and Luyten, B. (2015b), "SOLR statistics", available at: https://wiki.duraspace.org/display/DSDOC5x/SOLR+Statistics (accessed 6 December 2015).

Digital Commons (n.d.-a), "Digital Commons Network", available at: http://network.bepress.com/ (accessed 22 December 2015).

Digital Commons (n.d.-b), "Download statistics matter", available at: www.bepress.com/download_counts.html (accessed 28 November 2015).

Digital Commons (n.d.-c), "Institutional repositories published with Digital Commons", available at: http://digitalcommons.bepress.com/subscriber_gallery/ (accessed 22 December 2015).

Doran, D. and Gokhale, S.S. (2011), "Web robot detection techniques: overview and limitations", *Data Mining and Knowledge Discovery*, Vol. 22 Nos 1-2, pp. 183-210.

Doran, D. and Gokhale, S.S. (2012), "Detecting web robots using resource request patterns", *Proceedings – 2012 11th International Conference on Machine Learning and Applications, Vol. 1*, pp. 7-12.

DSpace Community Advisory Team and Luyten, B. (2014), "DCAT Meeting October 2014", available at: https://wiki.duraspace.org/display/cmtygp/DCAT+Meeting+October+2014 (accessed 12 December 2015).

Duskin, O. and Feitelson, D.G. (2009), "Distinguishing humans from robots in web search logs: preliminary results using query rates and intervals", *Proceedings of Workshop on Web Search Click Data, WSCD'09*, pp. 15-19.

EPrints.org and University of Southampton (n.d), "EPrints – Digital Repository Software", available at: www.eprints.org/software/ (accessed 28 November 2015).

Field, A. (2015), "IRStats 2 technical documentation – eprints documentation", available at: http://wiki.eprints.org/w/IRStats_2_Technical_Documentation (accessed 28 November 2015).

François, S. (2015), "IRStats2 – The EPrints Bazaar (Computer software)", available at: http://bazaar.eprints.org/365/ (accessed 28 November 2015).

Geens, N., Huysmans, J. and Vanthienen, J. (2006), "Evaluation of web robot discovery techniques: a benchmarking study", *Proceedings of the 6th Industrial Conference on Data Mining conference on Advances in Data Mining: Applications in Medicine, Web Mining, Marketing, Image and Signal Mining*, pp. 121-130.

Gordon, G. and Jensen, M. C. (n.d.), "Social Science Research Network", available at: www.ssrn.com (accessed 27 November 2015).

Huntington, P., Nicholas, D. and Jamali, H.R. (2008), "Web robot detection in the scholarly information environment", *Journal of Information Science*, Vol. 34 No. 5, pp. 726-741.

Information Power Ltd (2013), "IRUS download data: identifying unusual usage", available at: www.irus.mimas.ac.uk/news/IRUS_download_data_Final_report.pdf (accessed 11 December 2015).

IRUS-UK (2013), "IRUS-UK position statement on the treatment of robots and unusual usage", available at: www.irus.mimas.ac.uk/news/IRUS-UK_position_statement_robots_and_unusual_usage_v1_0_Nov_2013.pdf (accessed 3 December 2015).

IRUS-UK (2015), "IRUS-UK", available at: www.irus.mimas.ac.uk/ (accessed 12 December 2015).

Joint, N., Field, A. and Gregson, M. (2011), "Please change the way IRstats works", available at: www.eprints.org/tech.php/15695.html (accessed 28 November 2015).

Kwon, S., Kim, Y.-G. and Cha, S. (2012), "Web robot detection based on pattern matching technique", *Journal of Information Science*, Vol. 38 No. 2, pp. 118-126.

Lamothe, A.R. (2014), "The importance of identifying and accommodating e-resource usage data for the presence of outliers", *Information Technology and Libraries*, Vol. 33 No. 2, pp. 31-44.

MacIntyre, R. (2014), "IRUS-UK: making scholarly statistics count in UK repositories", *1:AM London Altmetrics Conference*, London, September 2014, available at: www.irus.mimas.ac.uk/news/IRUS-UKatAltMetricConf2014.pdf (accessed 12 December 2015).

Needham, P. and Stone, G. (2012), "IRUS-UK: making scholarly statistics count in UK repositories", *Insights*, Vol. 25 No. 3, pp. 262-266.

Olson, D.L. and Delen, D. (2008), *Advanced Data Mining Techniques*, Springer, Berlin Heidelberg.

Powers, D.M.W. (2011), "Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation", *Journal of Machine Learning Technologies*, Vol. 2 No. 1, pp. 37-63.

Sheaffer, R.L., Mendenhall, W. and Ott, R.L. (2006), *Elementary Survey Sampling*, Thomson, London.

Smith, M., Barton, M., Bass, M., Branschofsky, M., McClellan, G., Stuve, D., Tansley, R. and Walker, J.H. (2003), "DSpace: an open source dynamic digital repository", *D-Lib Magazine*, Vol. 9, p. 1.

Song, L., Gong, X., He, X., Zhang, R. and Zhou, A. (2013), "Multi-stage malicious click detection on large scale web advertising data", 1st International Workshop on Big Dynamic Distributed Data, Riva del Garda, Vol. 1018, 30 August, pp. 67-72.

Stassopoulou, A. and Dikaiakos, M.D. (2009), "Web robot detection: a probabilistic reasoning approach", *Computer Networks*, Vol. 53 No. 3, pp. 265-278.

Tan, P.N. and Kumar, V. (2002), "Discovery of web robot sessions based on their navigational patterns", *Data Mining and Knowledge Discovery*, Vol. 6 No. 1, pp. 9-35.

University of Southampton and EPrints.org (n.d.), "Registry of open access repositories", available at: http://roar.eprints.org (accessed 27 November 2015).

Unspam Technologies Inc. (2015), "Project Honeypot", available at: www.projecthoneypot.org (accessed 25 November 2015).

Van de Velde, K. and Diggory, M. (2015), "SpiderDetector.java (Computer software)", available at: https://github.com/DSpace/DSpace/blob/50b8cfd77e2640c3ae07a4e8d3e2482cbaa8df6b/dspace-api/src/main/java/org/dspace/statistics/util/SpiderDetector.java (accessed 6 December 2015).

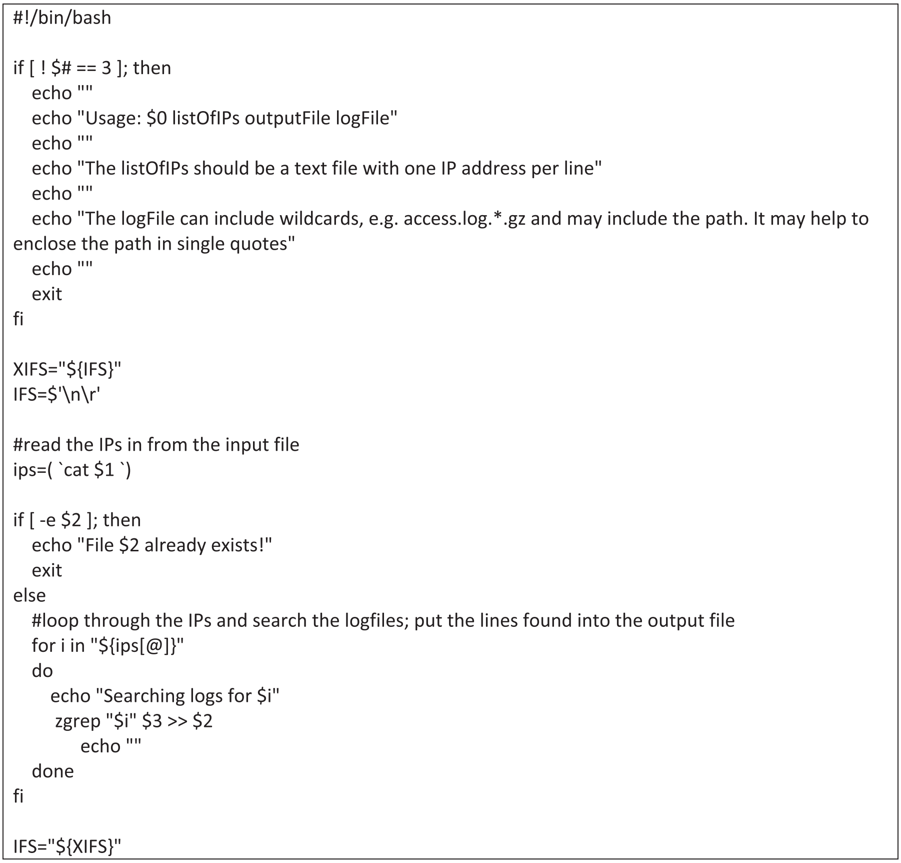
Zabihi, M., Jahan, M.V. and Hamidzadeh, J. (2014), "A density based clustering approach for web robot detection", *Proceedings of the 4th International Conference on Computer and Knowledge Engineering, ICCKE 2014*, pp. 23-28.

Zimmerman, C. and Baum, K. (n.d), RePEc: research papers in economics", available at: http://repec.org (accessed 27 November 2015).

Appendix 1. Selection of scripts, regular expressions, and queries used to extract data elements for download sample

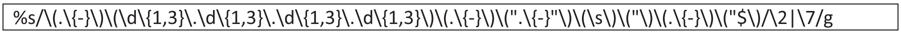
The following tools assume Apache combined format logs as input, and use of the VIM text editor (for regular expressions).

Script to extract all requests in a log file made by a set of IP addresses



```
#!/bin/bash

if [ ! $# == 3 ]; then
   echo ""
   echo "Usage: $0 listOfIPs outputFile logFile"
   echo ""
   echo "The listOfIPs should be a text file with one IP address per line"
   echo ""
   echo "The logFile can include wildcards, e.g. access.log.*.gz and may include the path. It may help to
enclose the path in single quotes"
   echo ""
   exit
fi

XIFS="${IFS}"
IFS=$'\n\r'

#read the IPs in from the input file
ips=( `cat $1 `)

if [ -e $2 ]; then
   echo "File $2 already exists!"
   exit
else
   #loop through the IPs and search the logfiles; put the lines found into the output file
   for i in "${ips[@]}"
   do
      echo "Searching logs for $i"
       zgrep "$i" $3 >> $2
           echo ""
   done
fi

IFS="${XIFS}"
```

Regular expression to extract IP address and user agent pairs from logs



```
%s/\(.\{-}\)\(\d\{1,3}\.\d\{1,3}\.\d\{1,3}\.\d\{1,3}\)\(.\{-}\)\(".\{-}"\)\(\s\)\("\)\(.\{-}\)\("$\)/\2|\7/g
```

Regular expression to extract date/time and IP addresses from logs for accesses to robots.txt

```
%s/^\(.\{-}\)\(\d\{1,3}\.\d\{1,3}\.\d\{1,3}\.\d\{1,3}\)\(.\{-}\[\)\(.\{-}\)\(\].\{-}"\)\(.\{-}robots\.txt.\{-
}\)\(".\{-}".\{-}".\{-}"\)\(.\{-}\)\("$\)/\2|\4|\6|\8/g
```

Regular expression to extract IP addresses and HTTP method from logs for accesses using the HEAD method

```
%s/^\(.\{-}\)\(\d\{1,3}\.\d\{1,3}\.\d\{1,3}\.\d\{1,3}\)\(.\{-}\)\(HEAD\)\(.*\)/\2|\4/g
```

SQL query for count of highest downloads of any item by a set of IP addresses (dl_peak_any_item)

```
SELECT tt.*
FROM
(select ip, item_id, count(*) from stats.download where date between '2013-11-09' and '2015-11-08'
and ip in
('65.55.abc.xyz',
'… etc.')
group by item_id, ip) tt
INNER JOIN
   (SELECT ip, MAX(count) AS maxcount
   FROM
(select ip, item_id, count(*) from stats.download where date between '2013-11-09' and '2015-11-08'
and ip in
('65.55.abc.xyz',
'… etc.')
 group by item_id, ip) b
   GROUP BY ip) groupedtt
ON tt.ip = groupedtt.ip
AND tt.count = groupedtt.maxcount;
```

This query is a base template for maximum count type queries, including dl_peak_this_item and dl_per_day_peak. Credit is given to stackoverflow user Michael La Voie for the logic of the query (http://stackoverflow.com/questions/612231/how-can-i-select-rows-with-maxcolumn-value-distinct-by-another-column-in-sql).

Appendix 2. Inverse precision as a function of recall, precision, and the ratio of robots to total downloads

Variables

Tp = true positive	R = recall (of robots)
Fp = false positive	P = precision (of robot detection)
Tn = true negative	Pinv = inverse precision (precision of human download statistics)
Fn = false negative	T = ratio of robots:total downloads

Formulae used in substitutions (Powers, 2011 except T)

$R = \frac{Tp}{Tp+Fn}$ $\qquad$ $Pinv = \frac{Tn}{Tn+Fn}$

$P = \frac{Tp}{Tp+Fp}$ $\qquad$ $T = \frac{Tp+Fn}{Tp+Fp+Tn+Fn}$

Substitutions

(1) First solve the formula T for Tp:

$$Tp = \frac{T(Fn+Tn+Fp)-Fn}{1-T}$$

(2) Substitute Tp in the formula R using Tp from 1:

$$R = \frac{T(Fn+Tn+FP)-Fn}{T(Tn+Fp)}$$

(3) Solve 2. for Tn:

$$Tn = \frac{T(RFp-Fn-Fp)+Fn}{T(1-R)}$$

(4) Substitute Tn in the formula Pinv using Tn from 3:

$$Pinv = \frac{T(RFp-Fn-Fp)+Fn}{T(RFp-Fp-RFn)+Fn}$$

(5) Solve the formula R for Tp:

$$Tp = \frac{RFn}{1-R}$$

(6) Substitute Tp in the formula P using Tp from 5:

$$P = \frac{RFn}{R(Fn-Fp)+Fp}$$

(7) Solve 6. for Fp:

$$Fp = \frac{RFn-PRFn}{P-PR}$$

(8) Substitute Fp in 4. using Fp from 7:

$$\mathrm{Pinv} = \frac{\mathrm{TRFn(R{-}PR{-}1) + 2TPRFn{-}PFn(T + R{-}1)}}{\mathrm{(R{-}1)(TRFn{-}PFn)}}$$

(9) Factor out Fn from 8:

$$\mathrm{Pinv} = \frac{\mathrm{TR(R{-}PR{-}1) + 2TPR{-}P(T + R{-}1)}}{\mathrm{R(TR{-}P{-}T) + P}}$$

This function shows primarily the effect that increasing or decreasing robot recall (at a given precision and ratio of robot downloads) has on the precision of human download counts. It could also be used to determine the inverse precision of a benchmarking study that reports R, P, and T.

Corresponding author
Joseph W. Greene can be contacted at: joseph.greene@ucd.ie