Digital Investigation 11 (2014) 314–322

Contents lists available at ScienceDirect

Digital Investigation

journal homepage: www.elsevier.com/locate/diin

Clustering digital forensic string search output

Nicole L. Beebe*, Lishu Liu

Information Systems and Cyber Security Department, The University of Texas at San Antonio, One UTSA Circle, San Antonio, TX 78249, USA

ARTICLE INFO

Article history:
Received 30 January 2014
Received in revised form 9 October 2014
Accepted 12 October 2014
Available online 12 November 2014

Keywords:
Digital forensics
Text string search
Clustering
k-means
SOM
LDA

ABSTRACT

This research comparatively evaluates four competing clustering algorithms for thematically clustering digital forensic text string search output. It does so in a more realistic context, respecting data size and heterogeneity, than has been researched in the past. In this study, we used *physical-level* text string search output, consisting of over *two million* search hits found in nearly 50,000 allocated files and unallocated blocks. Holding the data set constant, we comparatively evaluated k-Means, Kohonen SOM, Latent Dirichlet Allocation (LDA) followed by k-Means, and LDA followed by SOM. This enables true cross-algorithm evaluation, whereas past studies evaluated singular algorithms using unique, non-reproducible datasets. Our research shows an LDA + k-Means using a linear, centroid-based user navigation procedure produces optimal results. The winning approach increased information retrieval effectiveness, from the baseline random walk absolute precision rate of 0.04, to an average precision rate of 0.67. We also explored a variety of algorithms for user navigation of search hit results, finding that the performance of k-means clustering can be greatly improved with a non-linear, non-centroid-based cluster and document navigation procedure, which has potential implications for digital forensic tools and use thereof, particularly given the popularity and speed of k-means clustering.

Introduction

Text string searching is an extremely important digital forensic analysis technique, given the large amount of textual evidence that exists on digital media. Sources of important textual data include, but are not limited to email, web-based data, office productivity documents, address books, calendar data, journal data, activity logs, and application data. The strings of interest take on many forms, including human discourse, user created data, and alphanumeric strings such as named entities, phone numbers, IP addresses, credit card numbers, etc.

Although text string searching (literal or pattern matching) can improve investigator ability to answer key investigative questions, current digital forensic search techniques exhibit extremely poor information retrieval (IR) effectiveness, due to the onerous human analysis time caused by them (Beebe and Dietrich, 2007). Analysts are overloaded with string search output that exhibit extremely low signal to noise ratios. A reasonably constructed string search list processed physically over an average user's hard drive can easily net *millions* of search hits. Further, 95% or more of them are usually irrelevant to investigative objectives even though they are literal query-document matches. The analyst simply lacks the time and energy to wade through the voluminous amount of poorly organized search hits. As a result, important evidence can be overlooked (Alink et al., 2006).

As a result of the limitations of current digital forensic text string search hit techniques, investigators tend to employ one or more of the following coping mechanisms: 1) artificially constrain search lists by length and scope; 2)

* Corresponding author. Tel.: +1 210 458 8040; fax: +1 210 458 6305.
E-mail addresses: nicole.beebe@utsa.edu (N.L. Beebe), lishu.liu@utsa.edu (L. Liu).

http://dx.doi.org/10.1016/j.diin.2014.10.002
1742-2876/

analyze only a portion of the search hit output; 3) apply task forces where appropriate and possible; and 4) forego text string searching in lieu of other techniques. Analytical techniques commonly used in lieu of text string searching include: 1) allocated file review, 2) signature-based file carving, and 3) hash-based analysis. Unfortunately, such techniques are limited in their ability to uncover important textual evidence, particularly when it exists in fragmented or corrupted files in unallocated space.

Digital forensic practitioners need intelligent organization of text string search output to help them find *relevant* search hits more quickly. Digital forensic investigators need ranking and clustering algorithms, similar to what users of web search engines enjoy. However, given the stark contrast in data type heterogeneity between traditional information retrieval contexts (e.g. web searching and digital libraries) and digital forensic contexts, traditional ranking algorithms are not directly extensible to digital forensic applications. This is due to: 1) the inextensibility of critical ranking features (e.g. Google's PageRank™), and 2) the presumption that the corpus is *incrementally* indexed over time, which is not the case in individual, unique digital forensic cases.

The purpose of this study is to empirically examine four competing clustering algorithms, in the context of: 1) a more realistically sized set of string search hits, and 2) a more realistically heterogeneous data set, than has been examined in the past. Further, we examine multiple search hit output navigation algorithms to determine if a linear, centroid-based cluster navigation procedure is optimal. We theorized and tested two distinct algorithms simulating how users navigate through clustered search hit results to evaluate IR effectiveness. We studied potential factors that may further improve the performance, e.g. sequence of accessing documents and different cluster exit conditions. Such empirical insight is informative for tool developers as they design tools that cluster search hit output, since navigating clustered output is highly variable and less intuitive than navigating ranked list output.

The remainder of the paper is organized as follows. First, we position our contribution by reviewing related work and discussing research gaps that remain. We then outline our methodology, provide our empirical results, and discuss the implications of those results. We conclude with a discussion of the limitations and contributions of this research.

Related work

String search based research in the digital forensics domain exists, but is limited. Beebe and Clark (2007, Beebe et al., 2011) applied a scalable self-organizing map (SSOM) algorithm (Roussinov and Chen, 1999) to a student created synthetic case and a real-world case. Other clustering algorithms have also shown promise, specifically: (1) Latent Dirichlet Allocation (LDA) topical modeling (De Waal et al., 2008), (2) partitional clustering algorithms, such as k-means, bi-secting k-means, k-medoids, and expectation maximization (Decherchi et al., 2009; Nassif and Hruschka, 2013), and (3) hierarchical clustering algorithms, such as average-link and complete-link (Nassif and Hruschka, 2013).

A significant shortcoming in the related works is the fact that the majority of the referenced studies tested the algorithms on small and/or homogeneous data sets. In most cases, they did not test the algorithms on *physical level* string search output. Many studies have demonstrated the ability to cluster logically allocated (e.g. saved) files of same or similar file type. However, physical level string searches produce hits in saved files and unallocated blocks from extremely diverse data types and content. Past research has largely ignored the true heterogeneity that results from physical level string searches. De Waal et al.'s (2008) data set included five allocated text files types (.doc, .txt, .pdf, .html, .rtf). Nassif and Hruschka (2013) limited their data set to allocated .doc, .docx, and .odt files. Other studies focused solely on email messages, such as from the Enron email dataset (Decherchi et al., 2009; Hadjidj et al., 2009; Iqbal et al., 2010). Agreeably, such data is heterogeneous in content, but not in data type. Past research fails to show how prevailing clustering algorithms perform on the physical level string search hit output—characterized by extreme heterogeneity and large size.

Past research also lacks a comparative evaluation of promising algorithms. We cannot draw accurate conclusions regarding relative algorithm performance from related work, since past studies varied datasets and vector space model dimensionality. It is important to evaluate promising clustering algorithms "head-to-head," holding dataset, feature space, and algorithm implementation constant. Hence, this research empirically assesses the IR effectiveness of competing clustering algorithms on physical level string search hit output, while holding key variables constant. Based on algorithms selected in past research, we comparatively evaluated k-means, traditional Kohonen SOMs, LDA topic modeling followed by k-means clustering, and LDA topic modeling followed by Kohonen SOM clustering. We omitted hierarchical clustering algorithms, due to inherent scalability issues. Hierarchical clustering requires pair-wise similarity computations, so its computational expense is a quadratic function of the number of inputs being clustered $\sim O(n^2)$. While they tend to produce accurate results, we do not believe they will scale to real-world digital forensics problems.

Materials and methods

Dataset

We conducted our comparative experiments using the M57 Patents dataset, which is a synthetic case from DigitalCorpora.org (Garfinkel et al., 2009). The dataset contains, among other things, daily images of four synthetic users' work hard drives. Specifically, we utilized the *police seizure* images, which are the images from the last day of the scenario and simulate when law enforcement would have seized the disks. We conducted a literal string search with a 36 term search query. The search query was formed by compiling search strings recommended by several skilled digital forensic investigators, after receiving basic information about the case and the investigative goals. We specifically instructed the investigators to identify search strings *without regard* to the potential for false positives, for the following reasons: 1) to mitigate the impact of poorly formed search queries, and 2) to enable investigators to