available at www.sciencedirect.com

ScienceDirect

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

Digital Investigation

If *error rate* is such a simple concept, why don't I have one for my forensic tool yet?☆

*James R. Lyle**

National Institute of Standards and Technology (NIST), 100 Bureau Drive, Stop 8970, Gaithersburg, MD 20899-8970, USA

ABSTRACT

The Daubert decision motivates attempts to establish error rates for digital forensic tools. Many scientific procedures have been devised that can answer simple questions. For example, does a soil sample contain component X? A procedure can be followed that gives an answer with known rates of error. Usually the error rate of a process that tries to detect something is associated with a random component of some measurement. Typically there are two types of error, type I, also called a *false positive* (detecting it when it is not really there), and type II, also called a *false negative* (missing it when it really is there). At first thought, an error rate for a forensic acquisition tool or a write blocking tool is a simple concept. An obvious possibility for the error rate of an acquisition is k/n, where n is the total number of bits acquired and k is the number of incorrectly acquired bits. However, the kinds of errors in the soil test and in digital acquisition are fundamentally different. The errors in the soil test can be modeled with a random distribution that can be treated statistically, but the errors that occur in a digital acquisition are systematic and triggered by specific conditions. The purpose of this paper is not to define any error rates for forensic tools, but identification of some of the basic issues to stimulate discussion and further work on the topic.

1. Introduction

The motivation for the establishment of an error rate for forensic tools comes from a legal decision set in 1993 by the Supreme Court of the United States known as Daubert (Daubert v. Merrell Dow Pharmaceuticals, 509 U.S. 579 (1993)). The decision indicates four criteria that a trial judge may use to assess the admissibility of expert witnesses' scientific testimony during federal legal proceedings.

1. Has the theory or technique been tested?
2. Has the theory or technique been subjected to peer review and publication?
3. Is there a known or potential rate of error and do standards exist controlling the technique's operation? and
4. Does the technique have general acceptance within the relevant scientific community?

Many scientific procedures have been devised that can answer simple questions. For example, if we have a soil sample can we determine if some substance, call it X, is present? A procedure can be followed that gives a test result

☆ Certain trade names and company products are mentioned in the text or identified. In no case does such identification imply recommendation or endorsement by the National Institute of Standards and Technology, nor does it imply that the products are necessarily the best available for the purpose.

* Tel.: +1 (301) 975 3270.
E-mail address: jlyle@nist.gov

1742-2876/$ – see front matter © 2010 Digital Forensic Research Workshop. Published by Elsevier Ltd. All rights reserved.
doi:10.1016/j.diin.2010.05.017

with known rates of error. There are two possible test results: test positive and test negative. There are also two possibilities for the actual sample: yes, the sample contains X, and no, the sample does not. These can be combined into four possible outcomes:

1. the sample really contains X and the test result indicates X is present,
2. the sample does not contain X and the test result indicates X is not present,
3. the sample really contains X and the test result indicates X is not present, and
4. the sample does not contain X and the test result indicates X is present.

There are two types of error that can occur: if the test indicates that X is present when it is not this is called a type I error or a false positive. If the test indicates that X is not present (negative) when X really is there this is called a type II error or a false negative. For this sort of test a random distribution can be identified and statistical methods can be employed to determine an error rate for each type of error. One possible scenario that could cause the test procedure to produce an incorrect result is if there are two other items, Y and Z such that if Y is present then the test fails to react even if X is present and the test also reacts to Z as if X were present even if X is absent. The type I and type II error rates are related to the likelihood of having either Y or Z present in the sample.

Can we define similar error rates for forensic tools?

At first thought, an error rate for a forensic acquisition tool or a write blocking tool is a simple concept. An obvious possibility for acquisition is

$$\frac{k}{n} \tag{1}$$

where n is the total number of bits acquired and k is the number of incorrectly acquired bits. The same formula might be used for write blocking where n is the total number of write attempts and k is the total number of successful write attempts.

From the title of this paper the reader may correctly surmise that it is not going to be so simple. It turns out that many procedures followed by forensic practitioners tend to have errors that are systematic in nature rather than statistical. The purpose of this paper is not to define any error rates for forensic tools, but identification of some of the basic issues to stimulate discussion and further work on the topic.

It is also useful to consider possible sources of errors when following a procedure or process. There are three broad sources of error that can occur in execution of a procedure:

1. The algorithm intended for the process,
2. The software implementation of the algorithm, and
3. The performance of the process by a person.

As an example, consider a process to determine if two files have the same content without a copy of the original file. If we have some sort of checksum, message digest or hash of the original, we can compute the checksum of the test file for comparison to the checksum of the original. If we select a 16-bit CRC for our algorithm then the error rates for false negative is zero and the error rate for a false positive is 1 in 65,536. Of course, other algorithms such as 32-bit CRC, MD5 and SHA1 have much lower rates of a false positive.

As for the error rate of the software implementation of the algorithm, this is a little unclear. An error in the implementation can modify both the false positive and the false negative rates observed. However, it is unlikely that the change would follow a statistical distribution. A typical example of an error resulting from a faulty implementation would be something like the checksum for binary files is correctly computed but the checksum for text files is incorrectly computed. (The author has made this implementation error before.)

The performance of the procedure by a person introduces the chance that the person makes some type of mistake that introduces an error into the process. For example, if we want to compare the checksum of the file word.exe to the target file but the user substitutes the checksum for winword.exe we may not get the result we should.

The next sections of the paper consider if using the results of testing several classes of digital forensic tools to provide any insight into determining error rates for tools.

2. Disk imaging

Let's try to use Eq (1) k/n to establish an error rate for an imaging tool based on empirical data from a single test case reported in an NIJ Test Report for the tool SafeBack (Test Results for Disk Imaging Tools: SafeBack 2.18, June 2003). We are using SafeBack as an example even though the tool is outdated because the test report provides a rich set of interesting behaviors. For test case DI-006 of this report, the result of comparing a copy of a source disk containing 3,335,472 sectors to the original source is that 1,008 sectors differ. This gives an error rate of 0.0003025. However, this error rate is limited in application since it is based on a single test case from a total of 112 test runs. Selecting a different test run gives a different error rate. For example, most of the test cases acquired the source completely and accurately with no differences, i.e. an error rate of zero. This leads to the possibility of aggregating several test runs to obtain an overall error rate. Consider three of the other behaviors observed during testing SafeBack 2.18 and let's see if the other behaviors can be combined into a single error rate.

1. SafeBack allows acquisition of a drive through either of two access interfaces: BIOS access or ATA command access. For some hard drives, a different number of sectors are acquired for each interface.

For the test hardware, the computer BIOS underreports the drive size by 5,040 sectors. This happens because a BIOS may adjust the disk geometry parameters that it reports to force values within a required range (less than 1024). In this case, the BIOS adjusts the number of drive cylinders to be less than 1,024 by dividing the number of physical cylinders by 4, and multiplying the number of heads by the same constant. The BIOS gets the math wrong and reports one cylinder too few.

But, there is an additional problem when the cylinder count is adjusted. If there is a remainder from the division, then the modified parameters do not reflect any remaining cylinders. In this case, one cylinder of 1008 sectors is not accounted for. The actual drive has 3039 cylinders and 16 heads (1008 sectors per cylinder) for a total of 3,335,472 sectors. The BIOS reports 826 cylinders and 64 heads (4032 sectors per cylinder) for a total of 3,330,432 sectors. The BIOS should report 827 cylinders to get 3,334,464 sectors.

If SafeBack is used to acquire one of the drives by access through the BIOS, the underreporting is partially compensated for but the last 1008 sectors are omitted from the acquisition. Although SafeBack is designed to compensate for BIOS underreporting of drive size the tool only acquires 4032 of the possible 5040 sectors available beyond the reported drive size because it does not compensate for the partial (BIOS) cylinder.

However, if SafeBack is used to acquire one of the drives by ATA command access then all sectors are acquired.

2. If the drive contains a faulty sector, the tool attempts alternate read commands to obtain the sector contents. If the sector error correction code (ECC) indicates a data error then the corrupted data is acquired. For other errors, the sector is replaced by a block of zeros.
3. If a FAT32 partition is acquired to an image file the entire partition is acquired correctly. However, if the partition is copied or restored to a destination drive a few bytes in two sectors of partition metadata are changed.

It is unclear how to aggregate the test results into a single measurement of an error rate. The BIOS error occurs under specific conditions of access method, BIOS implementation and hard drive to acquire. If all the conditions are present then 1008 sectors are not acquired, if any of the conditions are not met, different BIOS, different access method or different hard drive, then all the sectors may be acquired or a different number of sectors may be omitted. One further observation is relevant. The omitted sectors are not used in normal operation with Microsoft operating systems and therefore do not usually contain used data. Of course, a sophisticated user may exploit this fact to hide a small amount of data.

The treatment of a faulty sector raised a different issue. The presence of a faulty sector is not an error made by the tool, but it might or might not be considered an error as far as the imaging process is concerned. If the sector is completely unreadable then it should not be considered an error to fail to read the sector. For example, if a hard drive, a very small drive, has 100 sectors and only 5 sectors are readable and the other sectors are unreadable, it seems that an error rate of 0.95 is misleading. The data from the 5 sectors can be accurately acquired with confidence that the acquired data is correct. If the bad sectors are still readable then the issues are more interesting. When a sector of data is read from a hard drive, a reference ECC is read in addition to the sector data. The sector data is used to compute another ECC. If the computed ECC does not match the reference ECC then there may be some corruption of the sector data and under some conditions the incorrect data bits can be identified and corrected. This happens within the hard drive and the host computer is not notified. If the corruption is too severe for correction then the read operation may be retried several times. If the data cannot be read and corrected then the host is notified that the sector is faulty. An imaging tool may be able to get the corrupted data through alternate read commands. The amount of corruption can range from none (the reference ECC is corrupted) to the entire sector. If the corruption follows a well-behaved random distribution then an error rate might be constructed for such sectors.

The FAT32 behavior occurs when a copy of a FAT32 partition is created. If either a copy is created directly from the source or a copy is created from an image file the resulting copy differs by two or three bytes in two or three sectors. If the image file is examined the differing bytes match the source, not the resulting copy. It should be noted that the forensic tool is writing the correct information to the partition, but that the operating system (Windows) is adjusting the partition metadata when the system is shut down. This can be verified by removing power to the drive containing the partition before doing a normal shutdown. In this case, the copy exactly matches the source. This behavior highlights a critical observation. The error behavior of a forensic tool may be influenced by the execution environment (operating system).

There does not seem to be a reasonable way to combine or aggregate all the errors into a single error rate. The errors can be quantified one by one, but no error rate presents itself that can be treated. The reason for this seems to be the nature of errors (Allchin, 2001). The kinds of errors that occur in the soil test are fundamentally different from the errors that occur in disk imaging processes. The errors in the soil test are statistical in nature and can be modeled with a random distribution, but the errors that occur in a digital acquisition are systematic in nature and triggered by specific conditions.

3. Write blocking

The following are some typical types of behavior that can be observed for write blocking software tools and hardware devices.

- One BIOS based software write blocker tool blocked both write commands (WRITE and WRITE LONG) that were in use at the time the tool was written. However, technology changed and a new write command was added to BIOS implementations and the tool failed to block the new command.
- Some hardware write block devices when paired with specific file systems and operating systems would allow acquisition of a hard drive but the operating system would lock-up or freeze if an attempt was made to browse the files on the protected drive.
- If a hardware write block device is updated with the wrong firmware, the blocker can be converted into a bridge.

These are, like disk imaging errors, also systematic errors. None of the errors have a statistically random component; the write blocker either works or not. So a write blocker would either have an error rate of zero (it works) or 1 (it always fails).

4. String search

String search tool behavior often varies in subtle ways depending of the settings of search parameters. There are many limitations and possibilities for design errors. Sometimes two different string search tools report slight differences in the number of matches for a given search string. For example, by doing a physical search of an image file without regard for the logical file structure of the source, any strings that span the boundaries of non-contiguous file fragments would not be matched. However, the missed strings would be found if the same tool is applied to each file in turn. Some of the other issues that can lead to differences in search results include the following:

- Strings in postscript (and many other document formats) may have formatting metadata embedded within the string. A search tool that cannot recognize the metadata will not match strings that should be found.
- The character representation, i.e., ASCII or some Unicode variant, is relevant to matching strings.
- A tool needs to be aware of diacritical marks such as tilde, accent, umlaut, etc. to be able to match text in many languages.

There is little room for differences in tool behavior to be attributed to random error, but there is quite a bit of latitude for variation in tool behavior due to limitations imposed by design decisions rather than any actual errors. It should also be noted that any matches found by a search can be verified for relevance by looking at the location indicated by the match. Since a false positive is easy to detect and eliminate, having an error rate may be of little value for string searching. It may be more useful to have a clear characterization of the search behavior and limitations.

5. File recovery and carving

There are two similar forensic activities that try to reconstruct previously deleted files. They are usually distinguished by the information used as the starting point for the reconstruction. Deleted file recovery begins with file system metadata for the deleted file. File carving on the other hand ignores file system metadata and scans unallocated data blocks for patterns that indicate the beginning and end of certain interesting file types. The success of both techniques is influenced by file system allocation and reuse policies within an operating system. There is opportunity for development of statistical models of file system block allocation, file fragmentation and block reuse policies. This requires additional work such as by Garfinkel (2007) to characterize real file system behavior.

6. Observations

One issue that has a major influence on any error rate is the degree of granularity to apply to any defined error rate. For example, it would be desirable to define an aggregate error rate, *r*, for disk imaging that states something about the reliability of the acquired data as a whole regardless of the source of an error. But what should *r* mean? Some possible meanings are that *r* is:

- the chance that an acquisition is corrupt,
- the fraction of the acquisition that is corrupt,
- the chance that any particular byte in the acquisition is corrupt.

On closer examination, none of these seems very useful and furthermore none of these measures reflect the actual way that acquisition tools actually fail. An acquisition may fail by omitting data or by including data from a source other than the intended source. These failures are typically triggered by the presence of a specific condition. For example, acquiring a hard drive replaces in the image file readable sectors of data with zeros if faulty sectors are present (Lyle and Wozar, 2007). Under the following conditions a rate can be established for the number of omitted sectors:

- *n* is the total number of sectors imaged,
- *e* is the total number of bad sectors,
- there is a separation of at least 8 sectors between bad sectors,
- the operating system is Fedora Linux,
- the imaging tool is the **dd** command without the **−direct** option, and
- the drive is imaged directly over the ATA interface.

The number of readable sectors omitted is 7*e and the error rate for omitted sectors is (7*e)/*n*. However, this is not a general error rate and changing any of the conditions may change the error rate. For example, changing the interface from ATA to USB (say by a bridge) changes the number of readable sectors replaced with zeros from 7 to a variable depending on the location of the faulty sector. Another example is using the **−direct** option on the **dd** command. In this case, no readable sectors are replaced with zeros, but the faulty sector, rather than being replaces with zeros, is replaced with data from an undetermined source. Again, it should be noted that these are actually systematic error; not errors with a statistical distribution.

It is also apparent that a specific forensic tool function may fail in any number of ways. For example, consider the following forensic functions and possible failure modes:

- Disk imaging
  - Completeness, getting all the data
  - Accuracy, the obtained data accurately reflects the source data
  - Logged parameters correctly
- Deleted file recovery
  - Completeness, identifying all deleted file meta-data
  - Correct association of meta-data to deleted content
  - Correct recovery of file meta-data
  - Identification of unrecoverable file content
  - Correct recovery of file content
- File Carving
  - Correct identification of file signatures: false positive rate and miss rate

- Correct reconstruction of end of file
- Correct omission of unrelated data

Each of the subdivisions represents a candidate for an error rate targeted to a particular type of error that could occur.

7. Conclusions

We make the following conclusions.

- A general error rate may not be meaningful.
- It is important to consider the source of the error: the intended algorithm or the implementation.
- An error rate should be defined that targets specific type of error that can occur during execution of a specific tool function. There may be several error rates associated with a specific tool function.
- It is in the fundamental nature of errors that some errors can be treated as statistical in nature, but some other errors are systematic in nature.
- The errors that occur in some critical forensic activities, e.g., disk imaging, are systematic in nature and no statistical error rate exists.
- In some situations, e.g., write blocking, the process either works correctly or it fails.
- For some forensic tools, e.g., string search, an error rate may be of limited value. The list of matches should be examined directly in any case to determine the context of the match. A rate for false positives matches returned may give the user a feel for the expected number of red herrings to be encountered but it gives no measure of the quality of the located information after the match is placed in context by direct examination.
- For a few tools, e.g., file recovery and carving, statistical error may yet play a role. A statistical error rate based on empirical data from the layout of real file systems may be possible. There would be difficulties since the allocation and block reuse policies vary across operating systems and can be expected to change over time.
- While human factors are important and would be useful to incorporate in a characterization of tool error behavior, it seems to be too difficult to be practical.
- To satisfy the spirit of Daubert, tools and techniques without a statistical error rate should have the types of failures and triggering conditions characterized.

REFERENCES

Allchin D. Error types. Perspect Sci Spring 2001;9(No.1):38–58.

Garfinkel S. Carving contiguous and fragmented files with fast object validation. Proceedings of the 2007 digital forensics research workshop, DFRWS, August 2007.

Lyle J, Wozar M. Issues with imaging drives containing faulty sectors. Proceedings of the 2007 digital forensics research workshop, DFRWS, August 2007.

Dr. James R. Lyle, Computer Scientist, wrote his first FORTRAN program in 1968 and has been programming ever since. He received a B.S. in Mathematics (1972) and an M.S. in Mathematics (1975) from East Tennessee State University; from the University of Maryland at College Park, Dr. Lyle received an M.S. (1982) and PhD (1984) in Computer Science.

Before joining NIST full time in 1993, Dr. Lyle was a Faculty Associate at NIST and an Assistant Professor at the University of Maryland Baltimore County.

Dr. Lyle's interests include Software Engineering, Computer Graphics, Human Factors, Digital Forensics and Computer Science Education.

His interests within Digital Forensics include:

- Specification of requirements for digital forensic tools
- Testing digital forensic tools.