Navigation of Cascading Data Errors

Kevin Marcus
Chief Technology Officer
Intelius
Bellevue, WA
kmarcus@intelius.com

Abstract— Growth in data mining, storage and retrieval systems combined with more affordable, powerful computer systems have created an insatiable appetite for high quality, robust data. Low cost, low impact systems such as mailing list software may be error tolerant, while life impacting systems such as employment screening may have substantial legal consequences. In 2007, the CDT reported identity theft, mistakes, simple typographical errors, and missing standards as key issues in the interpretation of data from these systems [1]. While the FCRA provides a legal framework around the use of data for certain screening purposes such as employment, the growing number of applications that lack transparency is a recipe for more litigation and legislation. The Internet, with its wealth of unstructured data of varying accuracy, further aggravates these challenges. As such, the need for improved tools to measure confidence and resolve conflicts within data is more important than ever. We examine several categories of data errors and resolution mechanics.

I. Introduction

Historically, there have been closed decision systems for gathering, analyzing and drawing inferences from data. Broad availability of data across the Internet has broken these closed systems into separate systems. These separate systems may no longer be aware of the data origin or how it was analyzed. For this paper, we examine data driven systems that are focused on gathering and analyzing data. These systems are subject to a multitude of possible data errors that can occur and materially influence the inferential.

Many data flow models take the output of one step as the input to the next. Even small errors in one step may be amplified as they move throughout the system. Reducing errors anywhere along this flow can have a profound impact on the lifecycle of data within a system. While many of the methods below may apply to a multitude of systems, here we focus on systems which are people centric and data which often is comprised of basic contact elements for people (name, address, phone), or attributes thereof (date of birth, hair color, etc.)

II. Error Origination

The first class of errors types we examine are related to data origination. These types of errors are at the beginning of most data flow models, so identifying (or correcting, if possible) the error is paramount to lessen its impact at later stages. There are two main solutions to these types of problems: data validation and multisampling. Simple data validation verifies a particular field is a legitimate entry, either by regular expression or table. For example, a standard US ZIP code is five digits. More complex implementations of data validation cross reference tuples of inputs such as comparing the ZIP code against the city/state pair to check for validity. Multisampling is the process of recording the same data from several vantage points. This could be asking the same user to input a field twice, visiting a web page several times over a period of time, or analyzing overlapping databases from different vendors who have different points of data origination (for example, a marketing file vs. a phone book).

A. Entry Errors: Human

The simplest of errors, data entry errors occur when a human interaction results in an incorrect recording of data. These can be as simple as typographical errors ("TEH" vs. "THE"), inputting the right information to the wrong field ("City: California"), or even transcription based (the form said 'B' but it really was an 8 ("Eight")). Data validation here can occur in several ways. One common approach is to ask the user to input the same value twice ("Please retype your email address"). This mechanism is prone to error primarily because the same user is more likely to make the same mistake. Furthermore if it is a system which allows copy/paste, this circumvents the purpose of asking for the same data twice. Depending on the system, another approach is to have a separate user also input the data. For example, in a transcription service, one might have the same page transcribed by two different people. Comparing the two pages for differences will highlight errors that can be then further improved. Input validation is another effective technique, prone to a separate set of problems. In this case, computer software compares inputs against expected possible inputs. Some data types may be difficult to analyze algorithmically this way; many "first names" could easily be "last names" as well. Clearly, combining these two methods is the most effective way to mitigate against these errors.

B. Digitization Errors: Software

While there have been major advances in Optical Character Recognition (OCR) software over the past decades, errors still do arise. These errors often appear similar to transcription errors, whereby a '5' may be recorded as an 'S' or similar. Multisampling is possible here via the use of separate OCR packages, the output of which still can run through systems that

attempt to validate the outputs. Disambiguating between the varied outputs can also be a resource challenge, as the ReCAPTCHA project demonstrates [2].

C. Incorrect Information

In some cases, the raw data may be incorrect. This can occur inadvertently or intentionally. For example, someone may input their business phone number when prompted for their home phone number. Inadvertent errors which pass validation rules require multisampling to effectively mitigate these types of errors. Intentional errors pose more challenge. For example, someone may lie about their age (for example, on a dating website). Since the user is actively trying to deceive the system, they will continue to try new inputs to pass form validation mechanics. In this case, multisampling may require additional factors: obtain similar information from different environments (compare the dating site age to the social network age), or introduce source confidence levels (voting records will have more accurate age information than a user submitted age). Particularly insidious, intentional errors propagate like a virus, spreading through an entire data set to corrupt and potentially invalidate other data. To help vaccinate against this, additional algorithms beyond simple validation and multisampling are required.

III. Conflict Handling

Handling data conflicts identified by comparing multiple samples is another challenge. Attempts at resolving these conflicts can introduce a new set of problems: Information may be improperly formatted, ambiguous, incomplete, or contradictory. As such, the tools required also change and most Extract, Transform and Load (ETL) systems lack the necessary sophistication to perform these enhancements. The end objective of these systems is to identify as many distinct attributes as possible over a period of time, identify persons who are the same, combine the best attributes together, and produce a single view of that individual.

A. Formatting Ambiguities

Internet sourced data will nearly always vary widely, often without any metadata to describe it. To begin deciphering this chaos, field definition and value must be carefully examined. Simple metrics such as dates can have widely varying layouts. European countries use DD/MM/YY, compared with the American MM/DD/YY (does "01/02/50" mean "Jan 2nd 1950" or "Feb 1st 1950"?) In the event all the data comes from the same source (or site), it may become possible to determine the date scheme from multiple profiles. For example, one could guess at a representation and then check to see if any of the records violate the format.

B. Precision

Sites yielding full presentation control to the user will make classification much more difficult. Many social networks will reveal users age but not their full date of birth. Yet, with the addition of a "zodiac sign", you can obtain a birth date by multisampling over a period of a month. In this case, one identifies the days of the year that apply to the sign, and then resample the data source to see if the age incremented. For example, a Taurus is born between April 19 and May 20. If a user is identified as a 20 years old Taurus, one could resample the site between April 19 and May 20 to determine the exact date of birth. This is substantially easier than resampling every day to check for Age increments.

Still other issues exist with name synonyms ("Rob" vs. "Robert" vs. "Bob"), nicknames, and abbreviations. Auxiliary translation tables can be used to map these to the same name for comparison purposes. Extreme care must be taken with translation tables to avoid loss of original user inputs. For example, mobile number portability and cell phones have disjointed the relationship between an area code and the phone's physical location. Furthermore, area codes change over time, further distancing physical location from a phone number.

Lastly, hierarchical information can be related, but not necessarily always disambiguated. For example, a person who likes all sports must like football, but a person who likes football does not necessarily like all sports. Care must be taken at each step to avoid losing the original intent of the distinct sample. All samplings must be viewed in the context which they were obtained.

C. Standardization

Standardization comes in many flavors, from simple translational mechanisms such as "California" morphs into "CA", to more complex issues, such as address standardization. While the USPS provides guidelines and many software packages are available to perform address standardization, there are still several unsolved problems [3]. First, there is no single global address standardization package. Second, even for locales where there is such a package, some addresses will still not standardize. In such cases, there are many possible causes: completely erroneous addresses, bad user inputs (substantially wrong city/state/ZIP for a particular street), addresses which are too new, or even addresses which no longer exist! For example, an individual may have address history which dates back thirty years or more. ZIP codes change over time, streets are renamed and apartments are bulldozed.

D. Incomplete or Contradictory Information

Over time, it is common for multisampling of a single individual to produce varied data points. This occurs when the data recording mechanism changes. A user may change their privacy settings on a social networking site, or warranty information cards for a different product may ask for interests in different ways. Furthermore, characteristics of people change over time, possibly yielding contradictory information. Clearly everyone should have a single birth day, but marital status can change (potentially along with the individuals last name). Each of these different samples should be recorded and used as inputs to a record merging algorithm.

IV. Merging and Deduping

Once data has been recorded in a consistent, well defined way, there must be a mechanism for identifying duplicate or

complimentary records. Duplicate records can be wholly removed, while complimentary records can be combined. One should note the distinction between how a typical internet search engine would perhaps identify pages with similar topics, but does not merge them together into a single page; rather they are left separate and distinct. For example, consider the volumes of pages mentioning "Michael Jackson" vs. the number of distinct individuals named "Michael Jackson".

The process for identifying which records are complimentary is studded with many unique challenges. For example, celebrities may have an enormous amount of information available, potentially dwarfing other people sharing the same name. Records with common names may not have enough information to distinctly identify them as a unique individual. As aforementioned, marriages can change last names. One approach to these challenges is to use a statistical model to identify probabilistic matches. However, any statistical model is prone to some error rate, and these can introduce still additional errors.

V. Application Logic

The last step in these data flow models is application use. Often times the application developer may not understand the extreme precision used to prepare data and may begin to make assumptions. For example, it is easy to assume one social security number has only one name associated with it. Yet, as mentioned earlier, names can change. Furthermore, with identity theft on the rise, it is becoming more commonplace to have multiple names associated with a single SSN. Data input errors could have associated incorrect information with a particular SSN. In part, this is the reason why credit reports are keyed by both SSN and last name: there can be multiple reports available -- potentially on totally different people -- with the same SSN.

Many applications use their own mechanics to determine whether or not there should be a match. Exact string matches may work well for phone numbers, but when querying for a name, a whole host of additional problems creep up. Should the application search for only "Rob" or "Robert" or "Bob" and "Bobby" also? People location services may want a broad matching mechanic while searching for Sex Offenders or Criminals may require more stringent matching requirements.

Occasionally, the raw data may not support enough precision for an application. For example, the Office of Foreign Asset Control and the US Treasury have lists of individuals and entities which are prohibited from doing business with the U.S. Yet a quick evaluation may find many individuals with varied date of births only specifying a year [4].

Applications may also violate "data symmetry," the concept that two distinct queries on separate fields that identify the same set of individuals should in fact, identify the same set of individuals. Consider, for example, a record where a unique name, address, and phone number exist. Querying by the name should produce this record, just as querying by phone number. However, for reasons such as access cost, interfaces, or even permissible use, applications may query alternate sources in alternate orders. For example, perhaps the source which contains the phone number is too expensive to query in response to a name based query. While querying by name or phone, both may return the address. However, the application may not show the phone number when there is a query by name. At the same time, a query by phone may show the name. In effect, these "short circuits" can provide alternate views of the same individual based on differentiated inputs.

VI. Transparency

End users often only see the application interface and receive a binary response. There is little transparency into an application response. It can be difficult to determine which source of information within the data flow model influenced a particular decision. Yet even with this transparency, end users do not control these data flows, further complicating the problem. In a recent article, Sylvia Short depicts a scenario where "it took me six weeks, four memos, and countless telephone calls to prove my existence" with a large financial institution [5]. Unfortunately, these systems treat all negative responses as "guilty until proven innocent."

As time progresses, more applications will make decisions on these data models. However, failure to provide transparency and empower users to verify and validate information about them ripens the legislative fruit. Growth in the use of these systems, combined with a loss of confidence is sure to bring more scrutiny to the industry.

Acknowledgment

The author would like to thank C. Andrew Neff for review and Jim Adler for encouraging the production of this article.

References

[1] Center for Democracy and Technology, "Amended Complaint and Request for Investigation, Injunction and Other Relief", pp. 5, http://www.cdt.org/privacy/20070725fcra-amendedpetition.pdf

[2] ReCAPTCHA, http://recaptcha.net/

[3] United States Postal Service, "Address Quality", http://www.usps.com/ncsc/

[4] Office of Foreign Assets Control, " Specially Designated Nationals List", http://www.treas.gov/offices/enforcement/ofac/sdn/sdnlist.txt

[5] S. Short, "A Lesson in Personal Finance from Citibank", http://dshort.com/articles/guest/Citibank-AA-mastercard.html