Application of Duplicate Records detection Techniques to Duplicate Payments in a Real Business Environment

Hussein Issa

Rutgers Business School, Rutgers University

ABSTRACT

Databases are increasing in size at an exponential rate, especially with the business electronization and real-time economy. Processes that were once stored on paper are now stored in databases. Sometimes errors – such as data entry, incomplete information, and unstandardized formats from different data sources – can lead to the existence of more than one representation of the same object in a database. This paper presents the problem of duplicate records and their detection, and addresses the issue of one type of records in specific which is of great interest in the business world: that of duplicate payments. An application of record matching techniques to the database of a telecommunication company is used as an illustration.

1. INTRODUCTION

Databases sizes are increasing at a fast rate in today's electronic business world. Most business entities nowadays depend on data gathered by their systems, and that applies to multinational corporations as well as small companies. That said, it becomes of great importance to assure the accuracy of this data, which serves as the base and cornerstone of a company's business operations. With simple databases, and in an ideal situation, data should have a global or unique identifier for every record, which would allow for these records to be linked across tables. Unfortunately, this is not the case in real life's complex databases. Many organizations have more than one system that collects data (e.g. SAP, Oracle, legacy systems), and these may differ

not only in values or identifier, but also in format, structure, and schema of databases. To add to this, data quality is also affected by human errors, such as data entry errors, and lack of constraints (e.g. allowing an entry such as Age: 430) (Chatterjee & Segev, 1991).

A frequent problem arises when data is gathered from different sources, whether from different systems or different locations, and that is duplicate records. Weis et al. describe duplicate records as "all cases of multiple representations of same real-world objects, i.e., duplicates in a data source" (Weis, Naumann, Jehle, Lufter, & H, 2008). One aspect of this problem is very important in the business world: duplicate payments. Duplicate payments can indicate many things, from simple data entry mistakes, to intentional fraudulent activities. No matter what the intention behind them is, duplicate payments can cause great losses to organizations. In 1998, for example, the Department of Health & Human Services evaluated the duplicate payments made by Medicare to be $89 million.

The literature, although abundant with studies of duplicate records in the computer science field, is very limited when it comes to the accounting aspect of it, such as duplicate payments. Most papers are technical and directed towards the algorithm details. Other studies are conducted by audit software companies. In my paper I present some methods used in the detection of duplicate records in databases and then I apply one of them to duplicate payments. I use data from a telecommunications company to test the 3-way and 4-way matching. The remainder of the paper is organized as follows. Section 2 presents the problem of duplicate records. In section 3 I discuss duplicate payments and the different techniques used in their detection. In section 4 I describe the data and the validation process. Preliminary results are presented in section 5. Lastly, I conclude with section 6.

2. DUPLICATE RECORDS DETECTION

2.1. Problem History

The issue of duplicate records has been known for years, and so has designing methods to solve it. In the 1960s and through the 1980s, detecting duplicates was referred to as *record matching* or

records linkage (Newcombe, 1988; Tepping, 1968). Nowadays this problem is mostly referred to as *duplicate record detection*. The main objective of record matching is to detect multiple representations of the same real-world object in a database (Elmagarmid, Ipeirotis, & Verykios, 2007).

The first concern about duplicate records was for medical records and for the purpose of epidemiological research. Then other areas became interested in this problem, such as tax agencies who wanted to gather information about tax payers with missing or incorrect social security numbers. Duplicate record detection was also found to be a very useful tool in the detection of fraud and money laundering (Ted, Goldberg, Wooton, Cottini, & Khan, 1995).

Most studies focused on domain-specific algorithms. In other words, most of the proposed algorithms in the literature addressed a specific issue, such as the address or census record. Very few were more general, but they assumed that the domain-specific knowledge will be provided by human expertise (Hernandez & Stolfo, 1995; Wang, Madnick, & Horton, 1989). In this paper I follow the first stream of studies and address one issue, that of duplicate payments.

2.2. Duplicates Detection Methods

Now that we have presented the history of the problem, we need to discuss the methodology of solving this problem.

Weis and Naumann (2005) describe a generalized framework for duplicate records detection that can be divided into three steps or phases:

- *Phase 1:* candidate description or definition: to decide which objects are to be compared with each other. There is no point for example in comparing name and address.
- *Phase 2:* duplicate definition: the criteria based on which two duplicate candidates are in reality duplicates. This is the variables or attributes that are considered relevant in the identification of duplicate records. For example, when trying to detect duplicate payments, the amount paid is relative, whereas size of a company is not.
- *Phase 3:* actual duplicate detection, which is specifying how to detect duplicate candidates and how to identify real duplicates from candidate duplicates.

The first two steps can be done offline concurrently with system setup (Weis & Naumann, 2005). The third step on the other hand takes place when the actual detection is performed and the algorithm is run.

Generally speaking, there are two methods of matching duplicate records:

- *Exact matching:* where the two records would be exactly the same in the dataset, identical in all fields. It consists of standardizing the values by removing all spaces and changing all letters to the upper case before checking for an exact match.
- *Fuzzy or near-identical matching:* where the two candidate duplicates have "similar" values for certain attributes (or variables). This may happen due to key punch errors, different ways of entering values. An example is address format: it can be entered as *123 East Fourth Street* or *123 E. 4th St.*; it can be one cell with the whole address, or divided into several cells for street address, city, state, zip code, etc.

When using the exact matching method, the database is searched for identical records, and records are classified as duplicated when the values are exactly the same. On the other hand, for fuzzy matching, the decision whether two records are duplicates is made based on a certain similarity criteria and a threshold (Weis, Naumann, Jehle, Lufter, & H, 2008). Some of the similarity criteria used in fuzzy matching are:

- Special characters: like hyphens and slashes.
- Leading and trailing character positions: like the location of a comma in numbers (34,567 vs. 345.67)
- Levenshtein distance: minimum number of operations, such as character insertion, change, or deletion, needed to transform one value into another

The threshold and the similarity criteria are situation-specific, in other words, they depend on the organization's policy and needs. Some companies classify candidate duplicates as duplicates or not based on a profile. An example of a company using that is Schuffa, where they have *k* base classifiers. Records are checked using classifiers, and every time a classifier classifies two candidates as duplicates, a point is added to the profile. The opposite is also true, where a point is subtracted when classified as non-duplicate. At the end, the points are added; the two records are

classified as duplicates if the score is higher than the threshold, and non-duplicates otherwise (Weis, Naumann, Jehle, Lufter, & H, 2008). In my paper, I use a similar technique, where records that are suspected to be duplicates are classified based on classifying rules. The next section presents the issue of duplicate payments, and presents different proposed methods used in the detection of such duplicates.

3. DUPLICATE PAYMENTS

3.1. Problem Description and Implications

The improvements in information and telecommunication technologies encouraged businesses to shift their processes from being traditional paper-based processes to digital ones. The use of accounting information systems in organizations helped in this transition by generating, computing, analyzing, and storing transactional data. All that led to the generation of large amounts of data being stored in digital format (Rezaee, Sharbatoghlie, Elam, & McMickle, 2002). Many systems were developed to take advantage of these large databases. It became common for companies to implement systems ranging from simple automated audit tools to the more complex Enterprise Resource Planning (ERP) systems, Decision Support Systems (DSS) and Knowledge-based Expert Systems (KES) (Chou, Du, & Lai, 2007). These systems, which are often used for internal audit functions, can assist auditors in their audit processes and help them detect fraudulent activities.

However, the quality of these systems as well as their efficiency depends greatly on the quality of data that feed into the databases. Errors that occur due to the integration of different systems or due to human error affect the quality of audit when similar systems are used. An example of these errors is duplicate payments. This can be the result of simple human errors, (e.g. typing mistakes), object presentation (e.g. checks paid to *Rutgers* vs. *Rutgers University*), or more serious systematic errors, like different structures from different sources (e.g. date format). They can also indicate fraud.

Duplicate payments are not a rare event. Two duplicate payments, alone amounting to $1.08 million, were discovered by the Department of Veterans Affairs' financial audit (Inspector

General, 1997). Medicaid *identified* more than $9.7 million in duplicate payments in a two-year audit period, and estimated the actual amount to be around $31.1 million (Novello, 2004). The keyword here is *identified.* These duplicate payments often go undetected. It is therefore important to implement techniques that would help in their detection. In fact, there exist some commercial companies that are specialized in detecting duplicate payments, usually known as recovery agencies, who would charge from 15% to 50% of any recovered amounts (Warner, n.d.).

3.2. Duplicate Payments Detection

The methodology for duplicate payments detection is generally the same as general duplicate records. However, as it is a specific problem, the criteria used for duplicates identification and classification can be specified in a little more details.

Most studies in the academic literature as well as in practice follow the general framework describes in the previous section, and use a 3-way match, with invoice amount, invoice date, and vendor's name (or vendor's number). Other studies include a fourth variable to refine the system depending on the situation. One study that dealt with duplicate Medicare payments used a 4-way match; however it added two more identifiers to describe the services (McMullan, 2001).

In another paper, more related to my research, a four-way match was described where (vendor number, invoice number, invoice date, and invoice amount) were used as duplicates criteria. This matching technique used in that study was a fuzzy matching, and it used 9-core algorithms (Please see Appendix A). The similarity was defined as amounts within 3% of each other, half or doubles of each other, consist of the same first 4 digits, and invoice numbers without leading or trailing zeros (e.g. invoices # 12300 and 00123RE are considered similar). Recurring payments, such as monthly rent or installments, were excluded from the beginning, as they were considered legitimate and common (Warner, n.d.).

In the following two sections I describe the dataset used in my study and the validation process, and then I present the preliminary results of the test that I ran on this data. These are still

preliminary as the verification of the results is still in progress, and I am still waiting for the feedback from the company's representatives.

4. DATA SET

4.1. Data description

The data used in this study belongs to the telecommunications company, which is a multinational company based in the US with branches and offices abroad. The data comprises of two separate files that were extracted from two different data sources. The information in these two files does not overlap. The period covered in these files was from July 2008 to June 2010.

The first data set (henceforth Carrier data) included information on payments to telecommunication carriers, eight attributes or variables and 21606 records or transactions. (Please see Appendix B for a list of the attributes)

The second data set (henceforth Oracle Fin) included information on payments done by checks, and it contained 47683 records and 51 attributes (please see Appendix A for a list of the attributes).

4.2. Data validation

The data preparation method that was described by R. Kimball and J. Caserta was followed (Kimball, 2009). It involves three steps:

- Parsing: allows for easier comparison by identifying individual elements in the dataset
- Data transformation: makes data conform to the data types of their corresponding domains. E.g. renaming a field, or converting a data element.
- Data standardization: standardizing the data into one format. E.g. an address can be written as (*123 East Fourth Street*, or *123 E. 4th St.*) and these two may be seen by the system as 2 different addresses, which would eventually increase the number of false positives.

For the Carrier dataset, no data transformation was needed. It had no missing values, and was fairly clean with clear self-explained attributes, and was used almost immediately. Only the issue

of two confusing variables (*effective_date* and *entered_date*) was not clear, but was quickly resolved after consulting the company.

Oracle Fin dataset, on the other hand, included a lot of meaningless attributes. To address this issue, TELECOMMUNICATIONS COMPANY was contacted, and we were informed to apply some filters to get rid of these irrelevant values. In addition to that, we found many missing values; however this problem was resolved after we applied the necessary filters. Some attributes also needed standardization.

Both data files were imported to MS Excel prior to being exported to ACL. For the duplicate payment detection, several combinations of variables (discussed later on in this paper) were used based on a three-way (for the Carrier dataset) and four-way (for Oracle dataset) matching techniques in order to determine the attributes that would yield the optimal results. Scripts were written in ACL in order to match records based on the relevant attributes.
The effectiveness of a method is measured using three metrics:

- Recall: $\frac{True\ Positives}{True\ Positives+False\ Negatives}$

  In other words, the correctly identified duplicates over all true duplicates

- Precision: $\frac{True\ Positives}{True\ Positives+False\ Positives}$

  In other words, the correctly identified duplicates over all found duplicates

- f-measure: $\frac{2.Precision.Recall}{Precision+Recall}$

  In other words, the harmonic mean of recall and precision

The goal is to maximize the f-measure by maximizing precision and recall. As we have not received the company's reply about the preliminary results that we sent them, we could not calculate the different precision and recall values for the different combinations used. However,

we were still able to get an idea on what combinations we might expect to produce the best results. The preliminary results are presented in the next section for both datasets.

5. PRELIMINARY FINDINGS

5.1. Carrier Dataset Findings

For the Carrier dataset, I only used a three-way match, as when I included the fourth one (which is *Payment_ID* referring to the invoice number) it yielded zero duplicate payments. Four different sets of variables were used:

A. *Carrier Name, Effective Date, Amount*
B. *Account Id, Effective Date, Amount*
C. *Carrier Name, Entered Date, Amount*
D. *Account Id, Entered Date, Amount*

The first two yielded the same number of duplicates, indicating a consistency between Carrier name and Account ID, and thus eliminating the possibility of errors in matching carriers to account IDs. This was also confirmed by the results from the third and fourth combinations.

Using Entered date (third and fourth combinations) resulted in 168 possible duplicates, while using Effective date yielded 82 duplicate candidates. The results from A & B turned out to be part of C & D's results, as all the 82 records were included in the 168 records from C & D. After consulting the company, most of these results turned out to be false positives; however we are still waiting on the remaining results to decide on the best way and to refine the system.

An interesting and even surprising finding was the presence of three *Commission payments*, which should not have been there since the company does not allow for such payments. After consulting the company, it turned out to be a "fat-finger mistake." *Commission payment* was underneath *Check payment* in the dropdown list under ***Transaction Type***. We also found that they were not correctly reversed. The company was advised of the issue, and we were informed later that it was removed from the Commission payment was removed from the dropdown list, and the reversal was adjusted.

5.2. Oracle Fin Dataset Findings

For this dataset I first tried the three-way match, using (*DATE, AMOUNT,* and *VENDOR_NAME)* but I got more than twenty four thousands of positively identified duplicates, so I needed to refine it. I added INVOICE ID (equivalent to invoice number) as a fourth variable, and the number of duplicated records detected by the system decreased tremendously. However, and due to the way the data was constructed, there were several columns related to *DATES* and *AMOUNTS*, so I had to try several combinations of variables using different dates and amounts columns (e.g. *CHECK_AMOUNT, APPLIED_INVOICE_AMOUNT, PAID_AMOUNT*) to see which ones would give the best results. That was a time consuming task, and the results were sent to the company for evaluation and verification.

Some of the combinations of variables that were used are:

A. *CHECK_AMOUNT, INVOICE_DATE, INVOICE_ID, VENDOR_NAME*
B. *INVOICE_AMOUNT, INVOICE_DATE, INVOICE_ID, VENDOR_NAME*
C. *APPLIED_INVOICE_AMOUNT , INVOICE_DATE, INVOICE_ID, VENDOR_NAME*
D. *CHECK_AMOUNT, CHECK_DATE, INVOICE_ID, VENDOR_NAME*
E. *PAID_AMOUNT, PAYMENT_GL_DATE, INVOCE_ID, VENDOR_NAME*

Other combinations were also used, but their results were not reasonable, so I did not list them here.

Here too we came across an interesting finding. There were more than twelve thousand refunds records in the data set. I am still running tests on them, and the company still needs to reply back to me. I tried using a threshold ($7000 based on the company's recommendation and $500 based on my intuition). This large number of refunds records may indicate the existence of some fraudulent activities, and if that was not the case, it may be an indicator of some inefficiency in the company's processes.

6. CONCLUSION

Globalization and electronization of business led to a great increase of databases. Systems were developed to handle and use effectively these databases. The use of automated audit and fraud detection systems became popular in the business world. However, the output quality of these systems remained largely dependent on the quality of the databases that fed into those systems.

Notwithstanding the obvious advantages of these large databases and their importance, databases streaming from different sources did not come without a cost. Different database systems have different formats, structures, and identifiers. They may also change from a country or region to another.

Duplicate payments, which can be defined as multiple representations of the same real-world object or entity, has a serious effect on the quality of audit and fraud detection systems. They can signify the presence of fraud, systematic errors arising from different database systems incompatibilities, or simply human errors. There is a plethora of cases in the literature showing the effect of duplicate payments on organizations and the amount of money lost because of it.

The problem of duplicate records, and more specifically duplicate payments, was discussed in this paper, in addition to the framework followed and methodology used in duplicates detection. I used the data from a telecommunications company as an illustration, and I explained the methodology that was followed and presented the preliminary results, until I get the final evaluation and assessment of the findings from the company.

7. REFERENCES

(2010). *Auditor's Guide to Duplicate Payments* (pp. 1-23).

Chatterjee, A., & Segev, A. (1991). Data manipulation in heterogeneous databases. *ACM SIGMOD Record*, *20*(4), 64-68. Retrieved from http://portal.acm.org/citation.cfm?id=141356.141385.

Chou, C., Du, T., & Lai, V. (2007). Continuous auditing with a multi-agent system. *Decision Support Systems*, *42*(4), 2274-2292. doi: 10.1016/j.dss.2006.08.002.

Elmagarmid, A., Ipeirotis, P., & Verykios, V. (2007). Duplicate Record Detection: A Survey. *IEEE Transactions on Knowledge and Data Engineering*, *19*(1), 1-16. doi: 10.1109/TKDE.2007.250581.

Hernandez, M., & Stolfo, S. (1995). The merge/purge problem for large databases. *Proceedings of the 1995 ACM SIGMOD*. Retrieved from http://portal.acm.org/citation.cfm?id=223807&dl=GUIDE,.

Inspector General. (1997). Audit report duplicate payments. Retrieved from http://www.va.gov/oig/52/reports/1997/7AF-G01-035--duppay.pdf.

Kimball, R. (2009). The data warehouse toolkit. wiley. Retrieved from http://scholar.google.com/scholar?hl=en&btnG=Search&q=intitle:the+data+warehouse+etl+toolkit#0.

McMullan, M. (2001). *Duplicate Medicare Payments by Individual Carriers*. *Public Law*.

Newcombe, H. (1988). Handbook of record linkage: methods for health and statistical studies, administration, and business. Retrieved from http://portal.acm.org/citation.cfm?id=63465.

Novello, A. C. (2004). Duplicate Medicaid Transportation Payments, 1-4. Retrieved from http://www.osc.state.ny.us/audits/allaudits/093004/04f2.pdf.

Rezaee, Z., Sharbatoghlie, A., Elam, R., & McMickle, P. (2002). Continuous auditing: Building automated auditing capability. *Auditing*, *21*(1), 147-163. doi: 10.2308/aud.2002.21.1.147.

Ted, E., Goldberg, H., Wooton, J., Cottini, M., & Khan, A. (1995). Financial Crimes Enforcement Network AI System (FAIS) Identifying Potential Money Laundering from Reports of Large Cash Transactions. *AI Magazine*, *16*(4), 21-39. Retrieved from http://www.aaai.org/ojs/index.php/aimagazine/article/viewArticle/1169.

Tepping, B. (1968). A model for optimum linkage of records. *Journal of the American Statistical Association*, *63*(324), 1321- 1332. Retrieved from http://www.jstor.org/stable/2285887.

Wang, Y., Madnick, S., & Horton, D. (1989). Inter-database instance identification in composite information systems. *Conference on System*, (June), 677-684. IEEE Comput. Soc. Press. doi: 10.1109/HICSS.1989.49184.

Warner, C. (n.d.). Duplicate Payment Detection. Retrieved from www.autoaudit.com.

Weis, M., & Naumann, F. (2005). DogmatiX tracks down duplicates in XML. In *Proceedings of the 2005 ACM SIGMOD international conference on Management of data* (p. 442). ACM. Retrieved from http://portal.acm.org/citation.cfm?id=1066207.

Weis, M., Naumann, F., Jehle, U., Lufter, J., & H. (2008). Industry-scale duplicate detection. *Proceedings of the*, *1*(212), 1253-1264. Retrieved from http://portal.acm.org/citation.cfm?id=1454159.1454165.

Appendix A

Duplicate Payment Logic:

Vendor Number	Invoice Number	Invoice Date	Invoice Amount
Exact	Exact	Exact	Exact
Different	Exact	Exact	Exact
Exact	Similar	Exact	Exact
Exact	Exact	Similar	Exact
Exact	Exact	Exact	Similar
Exact	Similar	Exact	Similar
Exact	Similar	Similar	Exact
Exact	Exact	Similar	Similar
Different	Exact	Similar	Exact

Appendix B

List of variables in the Carrier dataset:

Account ID, Payment Id, Carrier Name, Transaction Type, Effective Date, Entered Date, Amount, Source Info, Payment Number

List of variables in the Oracle Fin dataset:

BATCH_NAME, BATCH_DATE, ORG_NAME, ORG_ID, CHECK_NUMBER, CHECK_AMOUNT, CHECK_DATE, VENDOR_NAME, VENDOR_TYPE, VENDOR_PAY_GROUP, APPLIED_INVOICE_AMOUNT, INVOICE_ID, INVOICE_NUM, INVOICE_DESCRIPTION, INVOICE_DATE, AMOUNT_PAID, INVOICE_AMOUNT, PAYMENT_GL_DATE, PAYMENT_GL_PERIOD, ADDRESS_LINE1, ADDRESS_LINES_ALT, ADDRESS_LINE2, ADDRESS_LINE3, DISTRIBUTION_LINE_NUMBER, APPLIED_DIST_AMOUNT, DIST_DESCRIPTION, QUANTITY_INVOICED, DIST_GL_DATE, DIST_GL_PERIOD, DIST_CREATION_DATE, DIST_CREATED_BY, DIST_CREATED_BY_NAME, DIST_UPDATE_DATE, DIST_UPDATE_BY, DIST_UPDATE_BY_NAME, ASSETT_CATEGORY, SEGMENT1, SEGMENT2, SEGMENT3, SEGMENT4, SEGMENT5, SEGMENT6, SEGMENT7, SEGMENT8, SEGMENT9, DISTRIBUTION_ACCT_DESCRIPTION, BANK_ACCOUNT_NAME, BANK_ACCOUNT_NUM, TERM_NAME, CHECK_STATUS, CHECK_DESCRIPTION