Patient Matching Deduplication

Xinning Chu, Dandan Feng, Kang Fu, Ashley Hall

The ONC Patient Matching Challenge

Purpose

- Create greater transparency and data on the performance of existing patient matching algorithms
- Spur adoption of performance metrics for patient data matching algorithm vendors
- Positively impact other aspects of patient matching such as deduplication and linking of clinical data

Data

Uses a large data set, provided by ONC, against which participants ran deduplication algorithms and provided results for evaluation and accuracy measures. A small set of true-match pairs exist within the large data set, as served as the “answer key”

How

Three ways to Approach Patient Matching, see next slide for these

The Data

Provided by ONC

- Cleaned data as described in previous presentation

Three Approaches

Deterministic Matching

Unique identifiers for each record are compared to determine if two records are duplicates. This method tends to have high precision, low recall, which makes it a strong starting point to become familiar with a data set

Probabilistic Algorithms

The likelihood of duplicate records is determined by calculating the frequency of a value ('John') and the difference between two records ('Jon' vs 'John'), for example

Machine Learning

A set of rules are created by first "training" the algorithm (various). The algorithm is then applied to the complete dataset to identify duplicate records

Approaches

Deterministic Matching

What is deterministic matching?

This kind of matching algorithm involves finding exact matches between variables in two or more records.

In other words, with deterministic algorithms, several data elements must match exactly—without any typos or variation.

Thus, it is especially useful when unique identifiers such as social security number (SSN) are available.

Deterministic Matching

What kinds of variables do we choose to be the deterministic variables?

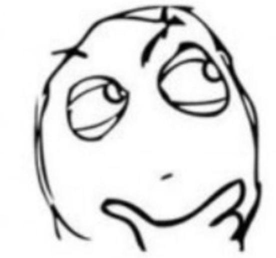
Variables we have in each record

Row ID	Enterprise ID	Last name	First name	Middle name
Date of birth	Gender	SSN	Address 1	Address 2
Zip code	City	State	Phone number 1	Phone number 2

Deterministic Matching

Need to find variables that cannot be changed so easily:

1. Row ID, Enterprise ID: Useless.
2. Names (First, Middle, Last) :May be changed. (A woman gets married)
3. Date of birth: Crucial (Apart from spelling errors and writing errors)
4. Gender?



Deterministic Matching

Need to find variables that cannot be changed so easily:

5. SSN : Crucial (Apart from spelling errors and writing errors)
6. Address, Zip code, City, State, Phone number: All can be changed in many situations.

So, the deterministic variables are:

1. **SSN.**
2. **Date of birth.**

Deterministic Matching

How to implement this method in R?

1. Package: RecordLinkage.
2. Function: Compare.depup

Problems:

1. Records with the same SSN and DOB may not represent the same person:

Example:

newLAST	newFIRST	newMIDDLE	DOB	numericalDOB	monthDOB	dayDOB	yearDOB	GENDER	SSN
LODATO	LYNN	NA	4/29/1974	27148	4	29	1974	F	879573579
DRAGOVICH	MATTHEW	NA	4/29/1974	27148	4	29	1974	M	879573579
KIRSCHBAUM	KATHLEEN	NA	5/15/1984	30817	5	15	1984	F	887274022
RITCHIE	DONALD	NA	5/15/1984	30817	5	15	1984	M	887274022
HAY	KAREN	NA	3/23/1965	23824	3	23	1965	F	873334989
BILBREY	CHARLES	NA	3/23/1965	23824	3	23	1965	M	873334989
ROE	KATHLEEN	NA	9/14/1991	33495	9	14	1991	F	893260266
JACOBSON	EDUARDO	BRADLEY	9/14/1991	33495	9	14	1991	M	893260266
BLACK	ROXANNE	NA	7/15/1999	36356	7	15	1999	F	869892089
DHANPAL	RICHARD	NA	7/15/1999	36356	7	15	1999	M	869892089

Embezzling other people's SSN?
Database error?

Maybe Gender should also be considered as a deterministic variables?

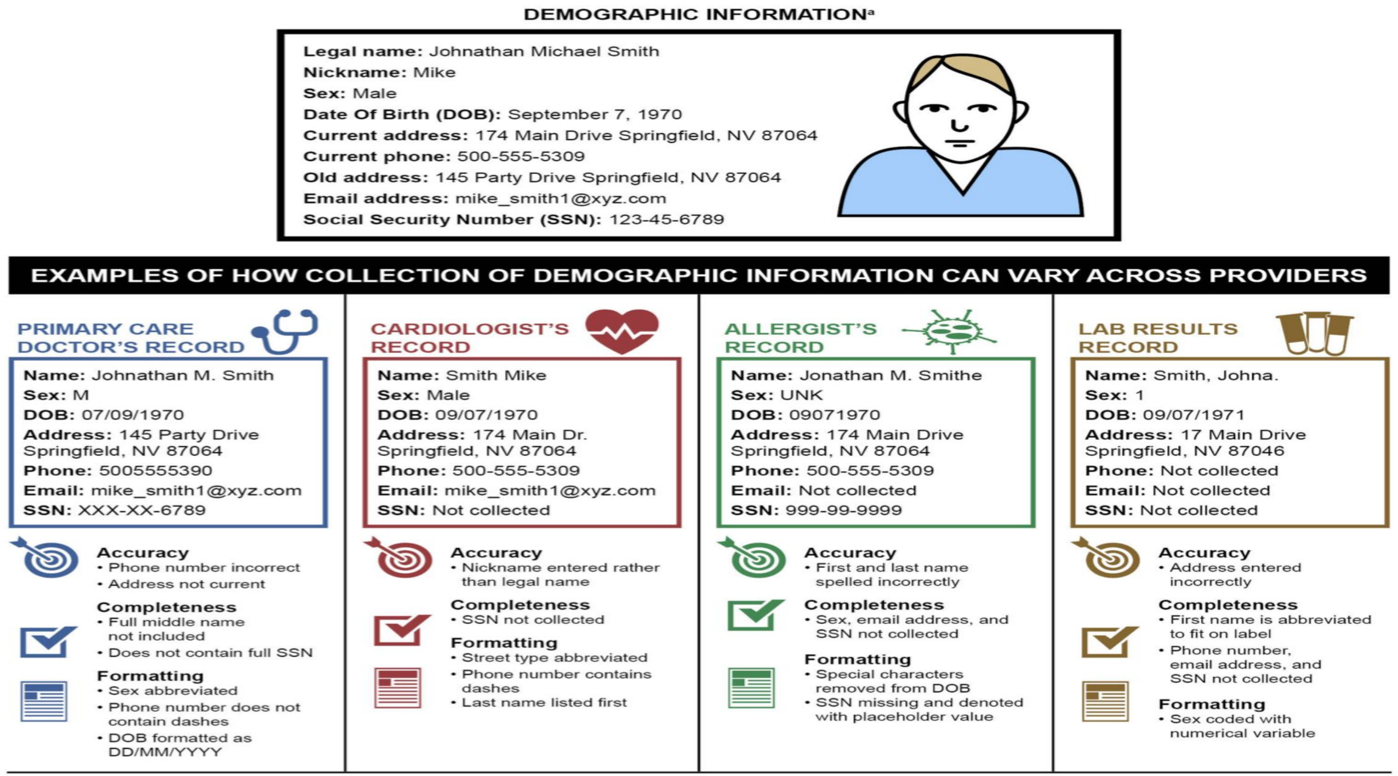
2. When a record is missing one of the three deterministic variables, the deterministic method cannot be used, even if other variables strongly indicate that it is the duplication of another record.

2. Minor spelling errors and writing errors (even if those errors can be easily distinguished by human eyes) can hugely affect the accuracy of the deterministic method.

To solve these problems, allow us to introduce you, the probabilistic algorithm!

Real-world Data is “Dirty”

Figure 1: Examples of Data Quality Issues That Can Affect Patient Record Matching



Source: GAO analysis. | GAO-19-197

Probabilistic Algorithms

Bipartite Record Linkage

Duplicate Elimination

↓

Record Linkage Problem:
Choose candidates have similar records

Improve Data Quality

EM-Based Probabilistic Record Linkage Model

1. Notation

Data Source A*B; Matched set M , Unmatched set U, Possible matched set P

Record Pairs: $r_{i,j} = (r_i, r_j)$, Component; $f_1, f_2, \ldots, f_n$

Component wise comparison $c_{i,j} = [c_1^{i,j}, c_2^{i,j}, \ldots, c_n^{i,j}]$ $c_k^{i,j} = C_k(r_i.f_k, r_j.f_k)$ $C_I(value_1, value_2) = \begin{cases} 0 & \text{if } value_1 = value_2 \\ 1 & \text{otherwise} \end{cases}$

2. Probabilistic Record Linkage Model

$\text{Prob}\{r_{i,j} \mid M\} = \prod_{k=1}^{n} m_k^{c_k^{i,j}} (1 - m_k)^{1-c_k^{i,j}}$, and $\text{Prob}\{r_{i,j} \mid U\} = \prod_{k=1}^{n} u_k^{c_k^{i,j}} (1 - u_k)^{1-c_k^{i,j}}$

Conditional probability $m_k = \text{Prob}\{c_k^{i,j} = 0 \mid r_{i,j} \in M\}$ $u_k = \text{Prob}\{c_k^{i,j} = 0 \mid r_{i,j} \in U\}$

Composite weight $L(r_{i,j}) = \sum_{k=1}^{n} w_k^{i,j}$ $w_k^{i,j} = \begin{cases} log(m_k / u_k) & \text{if } c_k^{i,j} = 0 \\ log((1 - m_k)/(1 - u_k)) & \text{if } c_k^{i,j} = 1 \end{cases}$.

Two threshold values t1< t2 $r_{i,j} \in M$ if $L(r_{i,j}) \geq t_2$; $r_{i,j} \in P$ if $t_1 < L(r_{i,j}) < t_2$ $r_{i,j} \in U$ if $L(r_{i,j}) \leq t_1$

3. EM based Probabilistic Record Linkage Model

$g_l = [1,0]$ if c_l represents a *matched* record pair.

Expectation step:

$$g_m(c_l) = \frac{p \prod_{k=1}^{n} m_k^{c_k^l} (1 - m_k)^{1-c_k^l}}{p \prod_{k=1}^{n} m_k^{c_k^l} (1 - m_k)^{1-c_k^l} + (1 - p) \prod_{k=1}^{n} u_k^{c_k^l} (1 - u_k)^{1-c_k^l}}$$

g_l is replaced by $(g_m(c_l), g_u(c_l))$

Maximum step:

$$ln\, f(y \mid \phi) = \sum_{l=1}^{N} g_l \cdot (ln\,\text{Prob}\{c_l \mid M\}, ln\,\text{Prob}\{c_l \mid U\})^T + \sum_{l=1}^{N} g_l \cdot (ln\, p, ln(1 - p))^T .$$

Steps

Bipartite Record Linkage

Package: RecordLinkage

1. Generating record pairs
2. Weight calculation: EM algorithm
3. Pattern classification

Outcomes

Bipartite Record Linkage

RowID.1	891091	RowID.2	933461	Weight
EnterpriseID.1	15378772	EnterpriseID.2	15631724	83.2270147
newLAST.1	PADILLA	newLAST.2	PADILLA	
newFIRST.1	GRIFFIN	newFIRST.2	GRIFFIN	
newMIDDLE.1	NA	newMIDDLE.2	NA	
DOB.1	12/10/90	DOB.2	12/10/90	
numericalDOB.1	33217	numericalDOB.2	33217	
monthDOB.1	12	monthDOB.2	12	
dayDOB.1	10	dayDOB.2	10	
yearDOB.1	1990	yearDOB.2	1990	
GENDER.1	F	GENDER.2	M	
SSN.1	816531571	SSN.2	816531571	
ADDRESS1.1	603 RUGBY ROAD	ADDRESS1.2	603 RUGBY ROAD	
ADDRESS2.1	2FL	ADDRESS2.2	NA	
CITY.1	BROOKLYN	CITY.2	BROOKLYN	
newSTATE.1	NY	newSTATE.2	NY	
PHONE.1	516-513-5249	PHONE.2	516-513-5249	
PHONE2.1	516-513-5249	PHONE2.2	516-513-5249	

Problems

Bipartite Record Linkage

We cannot resolve the matching pattern for three records.

Machine Learning

Decision Models:

Some Definitions:

1. Blocking: used to reduce the number of comparisons. Since potentially every record in one dataset has to be compared with every record in a second dataset, it groups similar records together and therefore partitions the datasets into smaller blocks (clusters).

2. Comparison Variables and functions:

Machine Learning

Assume that n common fields, $f_1, f_2, \ldots, f_n$, of each record from sources A and B are chosen for comparison. For each record pair $r_{i,j} = (r_i, r_j)$, the field-wise comparison results in a vector of n values, $c_{i,j} = [c_1^{i,j}, c_2^{i,j}, \ldots, c_n^{i,j}]$ where $c_k^{i,j} = C_k(r_i.f_k, r_j.f_k)$ and C_k is the comparison function that compares the values of the record field f_k. The vector, $c_{i,j}$, is called a *comparison vector* and the set of all the comparison vectors is called the *comparison space*. A comparison function C_k is a mapping from the Cartesian product of the domain(s), D_k, for the field f_k to a comparison domain R_k; formally, $C_k : D_k \times D_k \rightarrow R_k$. One example of a simple comparison function is

$$C_I(v_1, v_2) = \begin{cases} 0 & \text{if } v_1 = v_2 \\ 1 & \text{otherwise} \end{cases} \tag{1}$$

where $R_I = \{0, 1\}$. The value computed by C_I is called a *binary comparison value*. Two additional types of comparison values produced by comparison functions are *categorical* and *continuous*.

Machine Learning

Inductive Learning-based Decision Models

1. A training set of patterns, in which the class of each pattern is known a priori, is used to build a model that can be used afterwards to predict the class of each unclassified pattern.

2. Ex: decision tree

3. Advantage: handle continuous or numeric comparison vectors well

4. Disadvantage: The accuracy depends on the representativeness of the training data

Machine Learning

Clustering-based Decision Models

1. use the k-means clustering to group record pairs into three clusters: matched, unmatched, and possibly matched

2. Advantage: training data is not required

3. Disadvantage: the possibly matched record pairs do not necessarily form a distinctive cluster in real applications

4. Ex: The 3-cluster k-means algorithm thus leads to a large cluster of the possibly matched record pairs

Machine Learning

Enhanced Clustering-based Decision Models

Step:

Use a clustering algorithm to partition the record pairs into matched and unmatched clusters initially.

Form a third cluster (possibly matched) in a fuzzy region between the two main clusters.

Introduce a distance-based metric used for identifying the fuzzy region.

Machine Learning

Clustering Algorithm

k-means: easy implementation and computational efficiency when k is small

Good results can be achieved if all points are distributed around k well separated clusters.

The shape of these k clusters depends on the distance measure used. For example, if the Euclidean distance metric is used, the shape of the clusters is spherical for 3-dimensional data.

Other clustering algorithms, such as model-based clustering can also be used.

Machine Learning

Performance Metrics:

To compare different decision models, we need some performance metrics and an empirical experiment has been conducted.

Difficulties

1. Many records are incomplete or contain errors.

2. The scale of the problem requires efficient matching algorithms.

3. How to achieve both high efficiency and accuracy?