Deterministic Linkage Overview

Overview of Simple Matching

Nature of the Data

Data Cleaning and Standardization

Russell S. Kirby and Craig A. Mason
CDC / CMS Training February 2014

Deterministic Matching

- A series of common identifying fields are selected across two datasets
  - Records are matched based on these fields
  - Identical values across all these fields to be matched
- Easiest, quickest linkage strategy
- Ideal for some situations
  - Pre-existing common ID numbers
  - Very high quality/cleaned, standardized data
  - Homogenous population where loss is random, acceptable

Deterministic Matching

- May result in significant bias
  - Non-traditional spellings in names
- May result in errors due to non-links
  - Many non-links can result in greater bias than a few erroneous pairings
- Note that many of the more elaborate record linkage approaches still ultimately come down to deterministic matches of some fields or modified fields

Nature of the Data

- **Do people show up as a single record (birth certificate) or as multiple records (hearing screening)?**

Nature of the Data

- Do people show up as a single record (birth certificate) or as multiple records (hearing screening)?
- **How many do you expect to match?**
  - Everyone in one or the other file?
  - 70% of one file, half of the other?

Nature of the Data

- Do people show up as a single record (birth certificate) or as multiple records (hearing screening)?
- How many do you expect to match?
  - Everyone in one or the other file?
  - 70% of one file, half of the other?
- **What's the quality of the data?**
  - **Overall records**

Nature of the Data

- Do people show up as a single record (birth certificate) or as multiple records (hearing screening)?
- How many do you expect to match?
  - Everyone in one or the other file?
  - 70% of one file, half of the other?
- **What's the quality of the data?**
  - Overall records
  - **Individual fields**

Identifying Fields to Use

- High Quality
  - Poor quality data entry
  - Missing data
  - Biased or poorly operationalized data
- High Variability
  - Gender is not very useful
  - SSN can be very useful
    - Not always allowed to use even when available
    - Also has potential data quality issues

Simple Data Cleaning

- What type of errors do you expect to see?
  - Some can be identified and cleaned
  - DOB=4/13/3011
- Some will never be found
  - Data looks legitimate
  - May prove problematic as a linkage field
- Help identify cleaning and linkage strategies

Simple Data Cleaning

- Out-of-range values
- Invalid characters
  - *Values in the wrong field or column*
- Formatting (e.g., dates, names)
  - Leading or trailing spaces
  - Date formats
  - Missing data codes
- Spelling errors (e.g., Miammi)

Standardization Strategies

- Not primarily correcting errors
  - Removing potential ambiguities
- Recode multiple forms of the same value into a single format
  - “Street”, “St.”, “ST” as “St”
  - Standardize all abbreviations
- Remove all capitalizations or standardize to first letter only
  - “DeLean” to “Delean”

Standardization Strategies

- Periods
  - Middle initial “A.” versus “A”
- Remove all hyphenation and spaces in names
  - “Cobo-Lewis” to “Cobolewis”
- Possibly remove non-alphanumeric values
- Standardize various nicknames / alternative names
  - Robert, Bob, Bobby, all become “Robert”
- Maintain a copy of the raw value
  - CleanName, RawName

Iterative Passes and Rule-Based Linkage

Multiple Iterative Passes

- Conduct series of linkage attempts across two datasets
  - Each iteration typically includes only those records not matched in previous attempts
  - Each iteration attempts to capture true matches not yet identified, while limiting the number of new false matches
- Process
  - Start with most restrictive criteria (e.g., First, Middle, Last, DOB), move to less restrictive (e.g., First, Last, DOB)
  - Different combinations of identifiers at each step
  - Pairing different fields (Match ChildLastName on newborn screen with MotherLastName on electronic birth certificate (EBC))

Multiple Iterative Passes

- Hard to evaluate
  - Would the same results have been found had iterations been conducted in a different order?
  - Person A and B merged into one case (not good)
  - Person C matched with A in one scenario and B in another (worse)
- Track iteration number (metadata) and evaluate
  - For example, learn iteration #5 matched ChildLastName on newborn screen with MotherLastName on EBC and captured a large number of records

Rule-Based Matching

- Match two data files based on Social Security Number, First Name, Last Name, Date of Birth…
- Each possible match compared to multiple rules to determine whether records are matched
  - (1) If SSN agree, then ***match***
  - (2) If FirstName, LastName, DOB agree, then ***match***
  - Can be numerous different rules

Rule-Based Matching

- Algorithm goes through the rules
- As soon as a rule is satisfied, the pair is classified as correct match and the process moves to the next possible match

Rule-Based Matching

- Benefits
  - Can be fast and efficient
  - Can incorporate knowledge about various fields (e.g., SSN often missing, but very accurate when available)
- Generally no weights or cutoff scores
  - May have rules indicating ***manual review***
- Similar to multiple iterative passes (*but not exactly*)

Basics of Linking Data Deterministically

Which variables are common to both datasets??

Do a PROC Contents

Mother's information

Birth_mom_legal = Screen_mom_legal_last

Birth_mom_mid = Screen_mom_mid

Birth_mom_first = Screen_mom_first

Birth_mother_dob=Screen_mother_dob

Infant's information

Birth_child_last = Screen_child_last

Birth_child_mid = Screen_child_mid

Birth_child_first = Screen_child_first

_Birth_gender=_Screen_gender

Birth_child_dob=Screen_date

Other information

Birth_zip_code=Screen_zip_code

Birth_hosp=Screen_hosp

Missing data

Look for missing data in linkage variables

Ranking of linkage variables

Which variables are the “best” variables?

- How much missing data in each variable?
- What do you know about the variables?

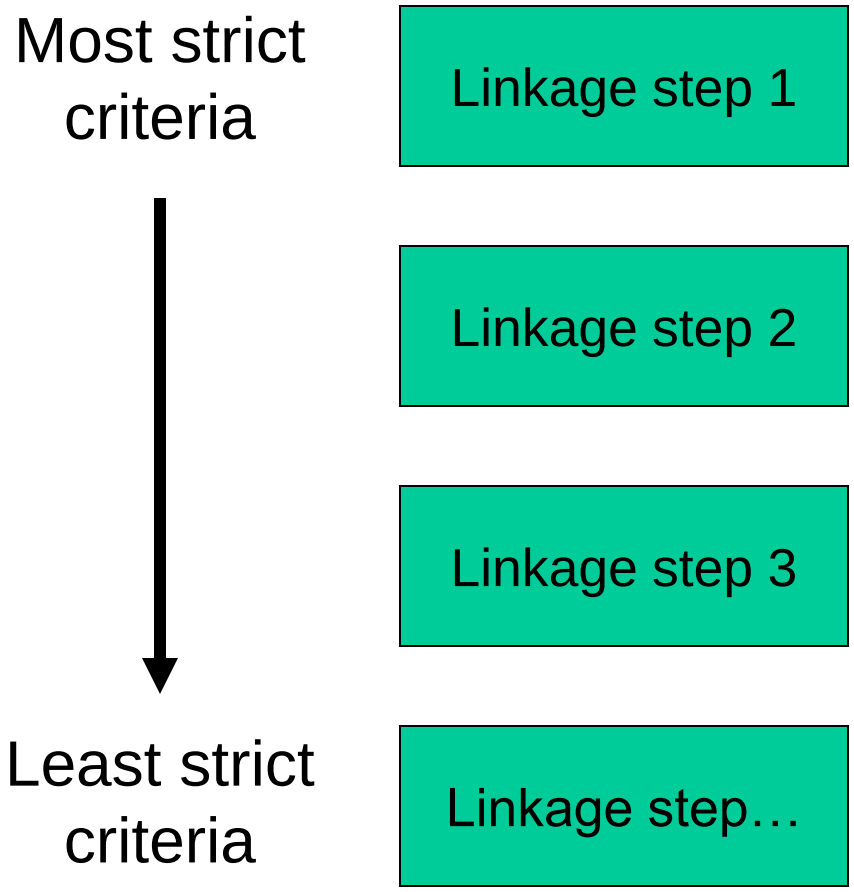
Our ranking

Fill in here

The art of creating a linkage algorithm

- Most discriminating combination of variables first
- Loosen criteria as go along

The art of creating a linkage algorithm



Create id in data set

- Allows you to easily merge back with original data
- Easy as:

```
data new;
set old;
id=_n_;
run;
```

Sort by chosen linkage variables

- What happens when you don’t use by variables??

- Let’s take a look . . .

Merge by chosen linkage variables

- Create data set with only linked records
- Keep track of the “link level” – level of linkage where records matched

Re-merge to get unlinked datasets

- Unlinked data sets contain only variables from that data set
- Unlinked records sent to next level of linkage algorithm

Last step

- Combine all linked data sets
- Investigate unlinked records
  - Look for systematic errors responsible for non-linking
  - Look for biases
- Evaluate quality of links in linked records