Data Linkage: A Primer

AAPOR Webinar
November 17th, 2022
Content of this session

- Introduction to Data/Record Linkage (RL)
- Identifiers
- Preprocessing
- Increasing the Efficiency of the Matching Step (Blocking)
- String comparators
- Probabilistic RL
- Application
- Appendix (RL Software, Literature)
Content of this session

- Introduction to Data/Record Linkage (RL)
- Identifiers
- Preprocessing
- Increasing the Efficiency of the Matching Step (Blocking)
- String comparators
- Probabilistic RL
- Application
- Appendix (RL Software, Literature)
Definition of Record Linkage

- RL is finding records in different data sets that represent the same entity and link them.

- RL is also known as *data matching, entity resolution, object identification, duplicate detection, identity uncertainty, merge-purge.*
The basic record linkage process

Data file A  
\[\rightarrow\]  
Comparison  
\[\rightarrow\]  
Classification  
\[\rightarrow\]  
Links  
\[\rightarrow\]  
Non links
5 Main Applications of Record Linkage

1. Merging of two data files
2. Identifying the intersection of the two data sets
3. Updating of data files (with the data row of the other data files)
4. Impute missing data
5. Deduplicate a file
Merging of two data files

- Merging of data files for microanalyses (e.g. survey- or registry data)
- Follow-up of cohorts (e.g. linkage with Cancer registry)
- Retrospective construction of panels
- Merging of panel waves
- Validation of answers in surveys: Comparing individual provided information's with registry data.
- Bias – detection in surveys: Supply data for nonrespondents.
- Supply external data for imputation or weighting of survey data
- Adding contact information to survey samples.
Identifying the intersection of the two data files

- Discovery of undercoverage within a census.
- Discovery of overcoverage and undercoverage in sampling frames.
- Examination of the reidentification risk of micro data files.
- Discovery of underreporting in registries (e.g. linkage with mortality registry).
- Dropping of duplicates as part of data cleansing.
Record Linkage Challenges (Christen 2012)

- Major challenge is that (clean) unique entity identifiers are not available in the databases to be linked.
  - Real world data are dirty (typographical errors and variations, missing and out-of-date values, different coding schemes, etc.)
- Data may require significant amounts of processing and data cleaning prior to linkage
- Scalability
  - Naïve comparison of all record pairs is computationally intensive
  - Remove likely non-matches as efficiently as possible
- No training data in many linkage applications
  - No record pairs with known true match status
- Privacy and confidentiality
  - Personal information, like names and addresses, are commonly required for linking
The extended record linkage process

1. Data file A
2. Data file B
3. Preprocessing file A
4. Preprocessing file B
5. Reduction of the search space
6. Comparison
7. Classification
8. Links
9. Non links
Content of this session

- Introduction to Data/Record Linkage (RL)
- Identifiers
- Preprocessing
- Increasing the Efficiency of the Matching Step (Blocking)
- String comparators
- Probabilistic RL
- Application
- Appendix (RL Software, Literature)
Identifiers

- Typical identifiers:
  - People: first and last name, address, birth date, sex
  - Establishments / firms: name, legal form, address

- The higher the number of different manifestations of an identifier, the better its suitability for a comparison.

- Complex identifiers should be parsed into its separate components

- Means of getting clean identifiers in the first place
Content of this session

- Introduction to Data/Record Linkage (RL)
- Identifiers
- **Preprocessing**
- Increasing the Efficiency of the Matching Step (Blocking)
- String comparators
- Probabilistic RL
- Application
- Appendix (RL Software, Literature)
Importance of Preprocessing

“In situations of reasonably high-quality data, preprocessing can yield a greater improvement in matching efficiency than string comparators and ‘optimized parameters’. In some situations, 90% of the improvement in matching efficiency may be due to preprocessing.” (Winkler 2009, p. 370)

“Inability or lack of time and resources for cleaning up files in preparation of matching are often the main reasons that matching projects fail.” (Winkler 2009, p. 366)
Shares of effort within linkage process

- 5% matching and linking efforts
- 20% checking that the computer matching is correct
- 75% cleaning and parsing the two input files

(see Gill 2001, p. 31)
Preprocessing: Workflow

1. Raw data files
2. Data definition
3. Parsing
4. Data cleaning
5. Normalization
6. Generation of derived variables
7. Filtering
8. Data files for linkage
Preprocessing: Workflow

- Parsing
- Data definition
- Raw data files
- Data cleaning
- Normalization
- Generation of derived variables
- Data files for linkage
- Filtering
Creating a data definition

A data definition records attributes for each identifier that are assigned to them conceptually.

It should encompass: variable name, variable type, data type, missing code, code list, variable range, among others…

Example of data definition for sex:

1. Variable name: sex
2. Variable type: categorical, coded
3. Data type: byte
4. Code list: 1 male 2 female 3 not determinable 9 missing
Preprocessing: Workflow

1. Parsing
2. Raw data files
3. Data definition
4. Data cleaning
5. Normalization
6. Generation of derived variables
7. Filtering
8. Data files for linkage
Parsing

- Parsing is the decomposition of a complex variable into single components.
- Subsequently, the single components can be composed to a standard form or can be used as single match variables.
- In simple cases the decomposition takes place through delimiter or through simple regular expressions.
  - Example: field with zip code and place name
### Example: Parsing of addresses

<table>
<thead>
<tr>
<th>Address</th>
</tr>
</thead>
<tbody>
<tr>
<td>39B Lexington Str. 01705 Chicago/ Wheaton</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>town name</th>
<th>district</th>
<th>zip code</th>
<th>street name</th>
<th>strqual</th>
<th>hnr</th>
<th>hnr_affix</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chicago</td>
<td>Wheaton</td>
<td>01705</td>
<td>Lexington</td>
<td>Street</td>
<td>39</td>
<td>B</td>
</tr>
</tbody>
</table>
Lookup tables for standardization

- Typical are tables for tokens in establishment names, personal names and addresses.

<table>
<thead>
<tr>
<th>Token</th>
<th>Replacement</th>
</tr>
</thead>
<tbody>
<tr>
<td>str</td>
<td>STR</td>
</tr>
<tr>
<td>Street</td>
<td>STR</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>Dr.</td>
<td>DR</td>
</tr>
<tr>
<td>Dcstr.</td>
<td>DR</td>
</tr>
<tr>
<td>Doctor</td>
<td>DR</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>Co</td>
<td>CO</td>
</tr>
<tr>
<td>Company</td>
<td>CO</td>
</tr>
<tr>
<td>Cmpy</td>
<td>CO</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>sen.</td>
<td>SENIOR</td>
</tr>
<tr>
<td>SENIOR</td>
<td>SENIOR</td>
</tr>
<tr>
<td>Junior</td>
<td>JUNIOR</td>
</tr>
</tbody>
</table>
Data cleaning: Overview

1. Evaluation of identifiers against data definition
2. Checking plausibility of variable values
3. Checking records for consistency
4. Standardization
5. Deduplication
Checking records for consistency

- Searching for consistency errors: values of least two variables contradict each other, while each value on its own is allowed.

- Examination through formulating and examination of edit-rules (brief: edits)

- For continuous variables mostly equations, for categorical variables mostly if-then rules
  - Ratio edits: $y/x = z$
  - Balance edits: $y + x = z$
  - Consistency edits: if AGE = 15 then STATUS != married

- Alternative: Examination through comparison with lookup tables, which contain pairs (or triples etc.) of variable values, e.g. zip code-place-street
Standardization

- Standardization of different representations of the same information in uncoded categorical variables.
- Without standardization there are more false negatives.
- However, over-standardization leads to risk of false positives.
- Always rule-based: application of a number of replacement rules to the identifier.
- Common implementation of the rules with regular expressions (search & replace).
Typical lookup tables for standardization

Entries and their corresponding standard representations:

- Abbreviations in street names (St: Street; Dr: Drive; Blvd: Boulevard)
- Abbreviations and frequent words in establishment names (b.o.: branch office; gen.: general)
- Nickname (Bob: Robert; Jim: James)
- Title (Dr: Doctor)
- Name affixes (v: von; sen: senior)

⇒ Highly country and language specific
Generating of derived variables

- Usually used to get appropriate blocking variables
- Typically over-standardized variants of existing identifiers
- Examples:
  - Phonetic codes of first and surname
  - Initial letters of first and surname
  - Truncation of zip codes to 3 or 4 digits
Statistics New Zealand: Standard preprocessing of surname

1. Take *surname*
2. Capitalize
3. Remove spaces
4. Set to missing if surname contains “unknown”
5. Remove any characters other than alphabetic characters
6. Name the resulting field *surname1*
7. Define new variable *initial_surname* = first character of *surname1*
8. Define new variable *soundex_surname* = Soundex code of *surname1* (See Statistics of New Zealand 2006, p.50)
Preprocessing: key take-aways

- Preprocessing is always specific for the concrete application.
  - Example: Establishment vs. individual data

- Expenditure of time for preprocessing often exceeds efforts of the record linkage (comparison, classification).

- Especially with bad data quality preprocessing is the most important factor for the success of linkage projects.

- Budget enough resources for preprocessing.

- Neither is there a universally suitable software for this, nor is there a comprehensive textbook.
Content of this session

- Introduction to Data/Record Linkage (RL)
- Identifiers
- Preprocessing
- Increasing the Efficiency of the Matching Step (Blocking)
- String comparators
- Probabilistic RL
- Application
- Appendix (RL Software, Literature)
The efficiency problem

- With $n$ records in file $A$ and $m$ records in file $B$, $n \times m$ pairs have to be compared.

- $100\,000 \times 100\,000 = 10\,000\,000\,000$ (10 billion) comparisons

- With 10,000 comparisons per second this takes 278 hours or 11.6 days
Standard technique: Traditional blocking

- According to its values, a variable partitions both data files into subsets, called blocks or pockets.
- The A- and the B-file are partitioned using the same (blocking) variable.
- Only pairs of records belonging to the same block within a certain file are compared.
Example: No blocking

|                |                 |                |                |                |                |                |                |                |                |                |                |                |                |                |                |
|----------------|-----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|
|                | Aldrin, San Diego, f | Pearce, New York, f | Johnson, Chicago, m | McDuff, Springfield, f | Fisher, Flint, m | Grgic, Little Rock, m | Miller, Lincoln, m | Powell, Los Angeles, f | Lassiter, Los Angeles, f | Harper, Seattle, f | McDowell, Boston, f | Martinez, Austin, f | Seinfeld, New York, f | Taylor, Portland, m | Hazard, Richmond, f | Brooks, New Orleans, m | Zarini, Cleveland, m |
| Johnson, Chigago, m | McDuff, Springfield, f | Fisher, Flint, m | Grgic, Little Rock, f | Miller, Lincoln, m | Powell, Los Angeles, f | Harper, Seattle, f | McDowell, Boston, f | Martinez, Austin, f | Seinfeld, New York, f | Taylor, Portland, m | Hazard, Richmond, f | Brooks, New Orleans, m | Zarini, Cleveland, m |
### Blocking by sex

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Johnson, Chigago</td>
<td>m</td>
<td></td>
<td></td>
<td></td>
<td>m</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>McDuff, Springfield</td>
<td>f</td>
<td></td>
<td></td>
<td></td>
<td>f</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Fisher, Flint</td>
<td>m</td>
<td></td>
<td></td>
<td></td>
<td>m</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Grgic, Little Rock</td>
<td>f</td>
<td></td>
<td></td>
<td></td>
<td>f</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Miller, Lincoln</td>
<td>m</td>
<td></td>
<td></td>
<td></td>
<td>m</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Powell, Los Angeles</td>
<td>f</td>
<td></td>
<td></td>
<td></td>
<td>f</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Harper, Seattle</td>
<td>f</td>
<td></td>
<td></td>
<td></td>
<td>f</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>McDowell, Boston</td>
<td>f</td>
<td></td>
<td></td>
<td></td>
<td>f</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Martinez, Austin</td>
<td>f</td>
<td></td>
<td></td>
<td></td>
<td>f</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Seinfeld, New York</td>
<td>m</td>
<td></td>
<td></td>
<td></td>
<td>m</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Taylor, Portland</td>
<td>m</td>
<td></td>
<td></td>
<td></td>
<td>m</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Hazard, Richmond</td>
<td>f</td>
<td></td>
<td></td>
<td></td>
<td>f</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Brooks, New Orleans</td>
<td>m</td>
<td></td>
<td></td>
<td></td>
<td>m</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Zarini, Cleveland</td>
<td>m</td>
<td></td>
<td></td>
<td></td>
<td>m</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Blocking by sex and location

<table>
<thead>
<tr>
<th>Name</th>
<th>City</th>
<th>Sex</th>
</tr>
</thead>
<tbody>
<tr>
<td>Aldrin, San Diego</td>
<td></td>
<td>f</td>
</tr>
<tr>
<td>Peace, New York</td>
<td></td>
<td>f</td>
</tr>
<tr>
<td>Johnson, Chicago</td>
<td></td>
<td>m</td>
</tr>
<tr>
<td>McDuff, Springfield</td>
<td></td>
<td>m</td>
</tr>
<tr>
<td>Fisher, Flint</td>
<td></td>
<td>m</td>
</tr>
<tr>
<td>Grgic, Little Rock</td>
<td></td>
<td>f</td>
</tr>
<tr>
<td>Miller, Lincoln</td>
<td></td>
<td>m</td>
</tr>
<tr>
<td>Powell, Los Angeles</td>
<td></td>
<td>f</td>
</tr>
<tr>
<td>Harper, Seattle</td>
<td></td>
<td>f</td>
</tr>
<tr>
<td>McDowell, Boston</td>
<td></td>
<td>f</td>
</tr>
<tr>
<td>Martinez, Austin</td>
<td></td>
<td>f</td>
</tr>
<tr>
<td>Seinfeld, New York</td>
<td></td>
<td>m</td>
</tr>
<tr>
<td>Taylor, Portland</td>
<td></td>
<td>m</td>
</tr>
<tr>
<td>Hazard, Richmond</td>
<td></td>
<td>f</td>
</tr>
<tr>
<td>Brooks, New Orleans</td>
<td></td>
<td>m</td>
</tr>
<tr>
<td>Zarini, Cleveland</td>
<td></td>
<td>m</td>
</tr>
</tbody>
</table>
Content of this session

- Introduction to Data/Record Linkage (RL)
- Identifiers
- Preprocessing
- Increasing the Efficiency of the Matching Step (Blocking)
- String comparators
- Probabilistic RL
- Application
- Appendix (RL Software, Literature)
String similarities

- Function of a pair of character strings with similarity as function value.

- Common: Standardization of the function value to the interval [0-1] (0: no agreement; 1: complete agreement).

- Variations of the following classifications of string similarity functions are commonly used:
  - Phonetics
  - Edit-distances
  - n-grams
  - Jaro‘s string comparator
**Edit-distances: Principle**

- An edit-distance between two strings $a$ and $b$ is the lowest number of permitted edit-operations needed to transfer $a$ to $b$.
- A certain edit-distance variant is defined by the set of permitted operations.
- For the Levenshtein-distance, for example insertions, deletions and substitutions are allowed.
- Common: Normalization using the sum of the length of the strings.
- Similarities are obtained by $1 - \text{LD}_{\text{norm}}$. 
# Levenshtein-distance: Examples

<table>
<thead>
<tr>
<th>Names</th>
<th>Edit operations</th>
<th>Norm. distance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Neumann</td>
<td>1 x substitution</td>
<td>1 x 2/14 = 0.14</td>
</tr>
<tr>
<td>Naumann</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Maier</td>
<td>2 x substitutions</td>
<td>2 x 2/10 = 0.40</td>
</tr>
<tr>
<td>Meyer</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mohr</td>
<td>1 x deletion, 1 x substitution</td>
<td>2 x 2/9 = 0.44</td>
</tr>
<tr>
<td>Moore</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Acri</td>
<td>1 x insertion</td>
<td>4 x 2/11 = 0.73</td>
</tr>
<tr>
<td>Ascheri</td>
<td>3 x deletions</td>
<td></td>
</tr>
<tr>
<td>Adams</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Adams</td>
<td>1 x insertion</td>
<td>1 x 2/9 = 0.22</td>
</tr>
</tbody>
</table>
Content of this session

- Introduction to Data/Record Linkage (RL)
- Identifiers
- Preprocessing
- Increasing the Efficiency of the Matching Step (Blocking)
- String comparators
- Probabilistic RL
- Application
- Appendix (RL Software, Literature)
Every pair of records is compared and represented using a vector of components that describe similarity between individual record fields (i.e. identifiers)

- E.g., “name agrees”, “name disagrees”, “name missing on one or both records”
Principles (I)

- Simply summing up identifier matches cannot be optimal.
- Different identifiers differ in how strongly an agreement is indicative for a link.

<table>
<thead>
<tr>
<th>Name</th>
<th>Sex</th>
<th>Residence</th>
<th>Date of birth</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tom McDonalds</td>
<td>m</td>
<td>Albuquerque</td>
<td>12/06/1966</td>
</tr>
</tbody>
</table>
Assigning appropriate weights to identifiers before summing up would be a better method.

In order to weight identifiers it must be quantified for each identifier how strongly an agreement indicates a link.

How likely is an agreement within the matches compared to within the non-matches?
The term probabilistic record linkage results from the fact that two conditional probabilities are considered:

\[ m_i = P(a_i = b_i \mid M) \]

m-probability for \( i \): Probability for agreement of records \( a \) and \( b \) for identifier \( i \) within the matches.

\[ u_i = P(a_i = b_i \mid U) \]

u-probability for \( i \): Probability for agreement of records \( a \) and \( b \) for identifier \( i \) within the non-matches.
Central likelihood ratio (I)

- The weighting of the identifiers is done by this ratio of likelihoods:

$$\frac{m_i}{u_i} = \frac{P(a_i = b_i \mid M)}{P(a_i = b_i \mid U)}$$

- The rarer an agreement occurs within the non-matches compared to the matches, the more strongly does an agreement within the identifier indicate a link.

- Therefore, \( \frac{m_i}{u_i} \) quantifies how strongly an agreement within an identifier \( i \) indicates a link.
The Fellegi-Sunter Approach: finding links

Given $m_i$ and $u_i$, link status is determined by considering the likelihood ratio (also known as the match weight or match score):

$$R = \frac{p(\gamma|M)}{p(\gamma|U)}.$$ 

Choose thresholds $T_1$ and $T_2$:

- Pairs with $R \geq T_1$ are linked
- Pairs with $R \leq T_2$ are not linked
- Pairs with $T_1 > R > T_2$ are sent for clerical review
Generally, the identifier *sex* agrees for 99% of the matches; within non-matches agreement usually occurs in 50% of cases.

In the case of the identifier *surname* typically agrees in about 80% among the matches and in 0.1% among non-matches.

An agreement on *surname* indicates a classification as a link much more strongly than an agreement on *sex*.

A disagreement on *sex* indicates a classification as a non-link much more strongly than a disagreement on *surname*.

<table>
<thead>
<tr>
<th>Variable (characteristics)</th>
<th>m-prob</th>
<th>u-prob</th>
<th>m / u</th>
<th>(1-m) / (1-u)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sex</td>
<td>0.99</td>
<td>0.5</td>
<td>1.98</td>
<td>0.02</td>
</tr>
<tr>
<td>Last name</td>
<td>0.80</td>
<td>0.001</td>
<td>800</td>
<td>0.2</td>
</tr>
</tbody>
</table>
Content of this session

- Introduction to Data/Record Linkage (RL)
- Identifiers
- Preprocessing
- Increasing the Efficiency of the Matching Step (Blocking)
- String comparators
- Probabilistic RL
- Application
- Appendix (RL Software, Literature)
Linking Deutsche Bundesbank Company Data

With a lot of help by Dr. Christopher-Johannes Schild and Sebastian Seltman
Data/Record Linkage: Goals of Bundesbank

- Improve data quality
- Increase analytical value of data
- More general and flexible Record Linkage System
- Historicized matching tables
Bundesbank’s relevant microdata sources and their connections (excerpt)
Bundesbank’s relevant microdata sources: Company Data
Company Data I: Microdatabase Direct Investment (MiDi)

- Information on inward foreign direct investments (FDI) as well as outward FDI
  - Granular information on FDI from domestic companies to companies located in other countries and incoming FDI from foreign owned companies to domestic and foreign owned companies
- Statistical units: reports that contain the investment relationship between the transaction parties
- Micro data is available as a panel
Company Data II: Corporate Balance Sheets (Ustan)

- Originates from Bundesbank’s refinancing activities.
- Information on earnings and financing.
- Detailed information on companies:
  - balance sheets, profit and loss accounts
  - comprehensive income statements („Ergebnisverwendungsrechnung“)
  - statement of changes in fixed assets („Anlagengitter“)
- Bias towards enterprises with higher credit-worthiness
In sum there are 15 company data to be linked:

- 7 analytical datasets from Bundesbank covering different time frames between 1987 – 2021.
- 8 master datasets covering different time frames between 1980-2021:
  - 5 from Bundesbank
  - 1 from BvD (reference data complemented by ZEW data)
  - 1 from Global Legal Entity Identifier Foundation (LEI)
  - 1 from the Statistical Agency, the (Statistical) Business Register of Germany
Numbers of companies of sum data data sets

Unique entities (pooled years)

Source

<table>
<thead>
<tr>
<th>Source</th>
<th>Filtered, RL input</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>AWMUS</td>
<td>294077</td>
<td>448712</td>
</tr>
<tr>
<td>BAKIS</td>
<td>1188714</td>
<td>1651542</td>
</tr>
<tr>
<td>BVD</td>
<td>7197013</td>
<td>12923930</td>
</tr>
<tr>
<td>JANIS</td>
<td>883400</td>
<td>885875</td>
</tr>
<tr>
<td>MIDI</td>
<td>51906</td>
<td>54212</td>
</tr>
<tr>
<td>RIAD</td>
<td>1012007</td>
<td>1726305</td>
</tr>
<tr>
<td>SIFCT</td>
<td>73213</td>
<td>74106</td>
</tr>
<tr>
<td>SITS</td>
<td>131333</td>
<td>133264</td>
</tr>
<tr>
<td>URS</td>
<td>2497231</td>
<td>6072512</td>
</tr>
<tr>
<td>USTAN</td>
<td>197680</td>
<td>197741</td>
</tr>
</tbody>
</table>
Data linkage

**Company data (non financial institutions (NFI))**:  

There is **no common unique firm identifier** in Germany.  

(Company business register-ID not stable)

We have to match firm data...  

- ... that do not have a common unique identifier / key  
- ... by using **alternative identifiers** (such as names, addresses, sectors, legal forms)

**RDSC has matched several NFI-microdatasets** (from Statistics, Banking Supervision and external data) with an advanced machine learning algorithm and generated a **matching table** (with probabilistic matching scores)
The basic record linkage process

Data file A → Comparison → Classification → Links, Non links

Data file B
There is no perfect world

- In a perfect world

<table>
<thead>
<tr>
<th>True Positive (TP)</th>
<th>True Negative (TN)</th>
</tr>
</thead>
</table>

- But we do not live in a perfect world

<table>
<thead>
<tr>
<th>True Positive (TP)</th>
<th>False Positive (FP)</th>
</tr>
</thead>
<tbody>
<tr>
<td>False Negative (FN)</td>
<td>True Negative (TN)</td>
</tr>
</tbody>
</table>
The extended record linkage process

- Data file A
  - Preprocessing file A
  - Reduction of the search space
  - Comparison
  - Classification
  - Links
  - Non links
- Data file B
  - Preprocessing file B
Record Linkage Process

Input Data A → Preprocessing → Blocking → Match-Candidates → Testdata, Trainingsdata → Classification-model → Automatic Evaluation → Postprocessing

Input Data B → Preprocessing → Blocking → Match-Candidates → Predictors (Features) → Manual Review

THX to Christopher-Johannes Schild
The “Fun” with Company Names: BMW as an example I

- BMW is an abbreviation for *Bayerische Motoren Werke* (…)
- This name is grammatically incorrect (in German, compound words must not contain spaces), which is why the name's grammatically correct form is *Bayerische Motorenwerke*
- *Bayerische Motorenwerke* translates into English as *Bavarian Motor Works*.
- The suffix AG, short for *Aktiengesellschaft*, signifies an incorporated entity which is owned by shareholders, thus akin to "Inc." (US) or PLC, "Public Limited Company" (UK).

(source: https://en.wikipedia.org/wiki/BMW)
The “Fun” with Company Names: BMW as an example II

- BMW
- BMW AG
- BMW Aktiengesellschaft
- Bayerische Motoren Werke
- Bayerische Motoren Werke AG
- Bayerische Motoren Werke Aktiengesellschaft
- Bayerische Motorenwerke
- Bayerische Motorenwerke AG
- Bayerische Motorenwerke Aktiengesellschaft

is the same company (and only the German possibilities are shown).
The “Fun” with Company Names: BMW as an example III

- In 2020 Germany had 7,389 Aktiengesellschaften.
- One of them is Bayer AG

- Bayer is clearly not BMW, but what is happening if you compare the names
- Bayer Aktiengesellschaft with BMW Aktiengesellschaft?
- Seems to be really close, right?
The “Fun” with Company Names: BMW as an example IV

- Last Sunday 677 BMW partners were found on [https://www.bmw.de/de/fastlane/bmw-partner.html](https://www.bmw.de/de/fastlane/bmw-partner.html)

- It is not always clear, if they use BMW in their names.

- You need a clear definition of what you mean by company, firm, establishment...
Preprocessing: Firm Names as one Example

- Remove known variation in different correct notations,
  - such as standardizing the German word “Gesellschaft” to its most common abbreviation “Ges”
  - and “&,” “+,” “und,” “and” etc to “UND”.
- Replacing German Umlauts “ä,” “ö,” “ü” by their common non-Umlaut replacements “ae,” “oe,” “ue” as well as capitalizing.
- Legal form information is extracted from the firm name field and removed from the firm name.

Doll, Gabor-Toth, Schild (2021): p. 8
Filter variables: 1. cleaned company name, 2. cleaned company name tokens, truncated, 3. city, 4. postal Code, 5. street name, 6. NACE (2 digits), 7. telephone, 8. founding year, 9. legal form.

Combination of these variables comes to a total of 1,130 blocking keys.

Overall, the blocking procedure reduces the number of comparisons from the order of roughly $N = 10^{13}$ to about $C = 10^8$ candidate pairs.

Classification Model

- **Comparison Features:**
  - A lot of different comparison features are used: name-based, location-based (including geo-references), digits from founding year and the sector code.

- **Groundtruth:**
  - based on common IDs

- **Training / Test Split**

- **Match Prediction**
  - The First Level “base”-Models are:
    1. random forest
    2. “extreme gradient boosting trees”-model (XGBoost)
    3. logistic regression
  - Second Level Model takes the first level model scores as features, plus 3 string comparison features.

Doll, Gabor-Toth, Schild (2021): pp. 10-14
Bias-Variance Tradeoff: Example I

THX to Sebastian Seltmann
Bias-Variance Tradeoff: Example II

Low Bias
High Variance

High Bias
Low Variance

Low Variance
Low Bias

Overfitting

Underfitting

Good Balance

THX to Sebastian Seltmann
Classification

- **Bias vs Overfitting**

THX to Christopher-Johannes Schild
Classification

\[ \text{true negative (TN)} \]

\[ \text{false negative (FN)} \]

\[ \text{true positive (TP)} \]

\[ \text{false positive (FP)} \]

THX to Christopher-Johannes Schild
Model Evaluation Measures: Scoring Functions I

- What is the share of correctly predicted cases?
  - Accuracy = (TN + TP) / Total
  - Total = TP+TN+FP+FN

- Which share of the true default cases is correctly predicted?
  - Sensitivity (or Recall) = TP / (TP + FN)

- Which share of the true non-default cases is correctly predicted?
  - Specificity = TN / (TN + FP)

THX to Sebastian Seltmann
Model Evaluation Measures: Scoring Functions II

- What is the share of correct “default” predictions?
  - Precision = TP / (TP + FP)

- Can the model identify true “default” cases without many false alarms?
  - F1-Score = 2 * \( \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \)
  - F1-Score provides a tradeoff between precision and recall

- How sure is the model in its predictions?
  - Log Loss = \(-\frac{1}{N} \sum_{i=1}^{N} [y_i \log p(y_i) + (1 - y_i) \log(1 - p(y_i))]
  - Penalizes confident incorrect predictions

Confusion Matrix

<table>
<thead>
<tr>
<th>Predictions</th>
<th>Actual Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>No Default</td>
<td>No Default</td>
</tr>
<tr>
<td>Default</td>
<td>FP</td>
</tr>
</tbody>
</table>

THX to Sebastian Seltmann
Evaluation

- Precision = \( \frac{TP}{TP + FP} \) = 98.0%
- Recall / Coverage = \( \frac{TP}{TP + FN} \) = 93.3%

THX to Christopher-Johannes Schild
Model Evaluation Measures: ROC Curve

- Receiver Operating Curve (ROC)
  - Plots the \textbf{tradeoff} between \textbf{Sensitivity} and \textbf{Specificity} for different probability thresholds
  - \textbf{Decrease threshold}: increase TP, but also FP
  - \textbf{Increase threshold}: increase TN, but also FN
- \textbf{AUC} (Area under the Curve)
  - \textbf{Probability} that a randomly chosen \textbf{positive} case (e.g., “default”) receives from the model a \textbf{higher score} (predicted probability) than a randomly chosen \textbf{negative} case (e.g., “non-default”)

THX to Sebastian Seltmann
Precision / Recall Curves, 1st and 2nd Level Model

Doll, Gabor-Toth, Schild (2021): p. 16
Matching Overlap of three Master Datasets (BvD, URS and JANIS)

Matching Overlap of three Bundesbank Datasets (1 Master, 2 Analytical)

Doll, Gabor-Toth, Schild (2021): p. 28
References


Data Linkage: A Primer

Appendix

(Software, Literature)
Software overview

- **Other (free) programs (see Appendix):**
  - Big Match
  - GRLS
  - The Link King
  - Link Plus
  - FRIL
  - Open Refine
  - Relais
  - R-Paket „RecordLinkage“
  - TDGen
Freely Extensible Biomedical Record Linkage (FEBRL)

- Project “Parallel Large Scale Techniques for High-Performance record linkage“
- Australian National University (ANU), Department of Computer Science
- Peter Christen
- Project: datamining.anu.edu.au/projects/linkage.html
- Version 0.4.2, 2013
- Download: sourceforge.net/projects/febrl
FEBRL: Features

- Freely available and expandable (open source license): **Python**
- Preprocessing module
- Probabilistic record linkage
- Further classification techniques
- Different blocking techniques
- Many string similarity functions
- Geocoding
- Blindfolded/Privacy Preserving Record Linkage
- Frequency weights
Merge ToolBox (MTB) is a Java application developed by the German RLC

Current version: 0.742, November 2012

Free use for academic purposes

To be found at: http://record-linkage.de/-Downloads--software.htm

Counseling by the German RLC
MTB: Features

- Probabilistic record linkage
- Many string similarity functions
- Several blocking techniques implemented
- Frequency weights
- 1-1 matching
- Parameter estimation using EM-algorithm
- Array-matching
- Privacy Preserving Record Linkage using Bloom Filters
XML-configuration files allow replicable MTB runs.

Particularly helpful during testing or if data files have to be divided for size-related reasons.

In the batch-mode configurations can be run successively and automatically.

After initially creating a configuration, (copies of) the XML-file can be adapted with external editors (e.g. Emacs, Notepad++, WinEdt)
BigMatch (U.S. Census Bureau)

- Useful for matching a very large file against a moderate size file
- Outputs all records from the large file that were stored as probable matches to the same record in the moderate file
- Functions as a preprocessing step to extract smaller files from very large files
  - Smaller files can then be efficiently processed using standard linkage software
- Written in portable C, can be run on Linux, Windows, MacIntosh, HP machines running the VMS operating system
- Allows one to run several different blocking criteria
- Developers: William Yancey / Bill Winkler
Generalized Record Linkage System (GRLS)

- Record linkage Software from Statistics Canada
- www.statcan.gc.ca
- Probabilistic record linkage
- Costs: about $30,000 one time, annual fee/charge
- Software + Support
- Follow-up software G-Link does not need local DBMS any more.
The Link King

- Kevin Campbell, Washington State Division of Alcohol and Substance Abuse
- www.the-link-king.com
- Version 7.1.21, 2012
Link King: Features

- Probabilistic record linkage
- String similarity functions
- Base SAS license necessary
- Blocking variables cannot be chosen freely
Link Plus

- U.S. Department of Health and Human Services, Centers for Disease Control and Prevention (CDC)
- [www.cdc.gov/cancer/npcr/tools/registryplus/lp_tech_info.htm](http://www.cdc.gov/cancer/npcr/tools/registryplus/lp_tech_info.htm)
- Version 2.0, 2007
- Developed for the implementation of the „National Program of Cancer Registries“ (NPCR)
Link Plus: Features

- Probabilistic record linkage
- Edit-distances
- Flexible blocking
- Frequency weights
- 1-1 matching
Pawel Jurczyk, Department of Mathematics and Computer Science, Emory University

fril.sourceforge.net

Version 2.1.5, 2011
FRIL: Features

- Probabilistic record linkage
- String similarity functions
- Different blocking techniques (e.g. sorted neighborhood)
- Parameter estimation using EM-algorithms
Record Linkage At IStat (Relais)

- L’Istituto nazionale di statistica
- [www.istat.it/it/strumenti/metodi-e-software/software/relais](http://www.istat.it/it/strumenti/metodi-e-software/software/relais)
- Version 2.2
Relais: Features

- Probabilistic record linkage
- Rule-based matching
- String similarity functions
- Blocking and Sorted Neighbourhood
- 1-1 Matching
- Parameter estimation using EM-algorithms
R-Package „Record Linkage“

- Record linkage: Detecting duplicate data project; Murat Sariyar/Andreas Borg, IMBEI, Uni Mainz
- r-forge.r-project.org/projects/recordlinkage
- Version 0.4-1, 2012
Record linkage: Features

- Probabilistic record linkage
- String similarity functions
- Blocking
- Parameter estimation using EM-algorithms
- Further classification techniques (machine learning)
Test Data Generator (TDGen)

- Free software developed by the GRLC
- Download of software and English manual from record-linkage.de/-Downloads--software.htm
- Tool to take arbitrary test file and generate a garbeled version of it by introducing simulated errors.
- Flexible control over error insertion probabilities.
- Things to control:
  - Fraction of erroneous rows
  - Number of erroneous columns in each erroneous row (defined manually or by Poisson or uniform distribution)
  - Error types (typographical errors, OCR-type errors, phonetic errors by reverse-modelling their encoding rules and many more)
Literature Review: Record Linkage Overviews


Literature Review: Introductions


Literature Review: Bibliographies and Surveys


Literature Review: Preprocessing

Literature Review: Blocking Methods


  *IEEE Transactions on Knowledge and Data Engineering* 24(9) 1537–1555.


Literature Review: Rule-Based Record Linkage


Literature Review: Distance-Based Record Linkage


Lit. Review: Comparisons of String Similarity Functions


Literature Review: Probabilistic Record Linkage


Lit. Review: Adjustment of Weights for String Similarities


Literature Review: Frequency Weights


Literature Review: Parameter Estimation


Lit. Review: Threshold Selection/Error Rate Selection


Literature Review: 1-to-1 Assignment


Lit. Review: Bias due to Imperfect Record Linkage


Literature Review: Record Linkage Software


Lit. Review: Privacy-Preserving Record Linkage

