Implementing Probabilistic Record Linkage

- Standardizing
- Blocking and matching variables
- Calculating the agreement index
- Choosing $M$ and $U$ probabilities
- Estimating $M$ and $U$ probabilities using EM
- Clerical editing
- Estimating the false match rate
- Estimating the false nonmatch rate
Matching Software

• Commercial ($$$$$-$$$$$$)
  – Automatch/Vality/Ascential/IBM WebSphere Information Integration
    (grew out of Jaro’s work at the Census Bureau)
  – DataFlux/ SAS Data Quality Server
  – Oracle
  – Others

• Custom software (0-$$)
  – C/Fortran Census SRD-maintained software
  – Java implementation used in Domingo-Ferrer, Abowd, and Torra (2006)
  – Java Data Mining API
Software differences

• Each software is an empirical/practical implementation driven by specific needs

• Terminology tends to differ:
  – “standardize”, “schema”, “simplify”
  – “block”, “exact match”
  – “comparator function”, “match definition”
<table>
<thead>
<tr>
<th>Feature</th>
<th>Custom/ SRD</th>
<th>Vality/Ascent ial</th>
<th>SAS DQ</th>
</tr>
</thead>
<tbody>
<tr>
<td>Standardizing</td>
<td>standard</td>
<td>Yes</td>
<td>PROC DQSCHEME</td>
</tr>
<tr>
<td>Blocking and matching variables</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes (blocking = “exact match”)</td>
</tr>
<tr>
<td>Calculating the agreement index</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Choosing $M$ and $U$ probabilities</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Estimating $M$ and $U$ probabilities using EM</td>
<td>Yes (eci)</td>
<td>(external)</td>
<td>No</td>
</tr>
<tr>
<td>Matching</td>
<td>matcher</td>
<td>Yes</td>
<td>PROC DQMATCH</td>
</tr>
</tbody>
</table>
STANDARDIZING
Standardizing

• Standardization is a necessary preprocessing step for all data to be linked via probabilistic record linking

• A standardizer:
  – Parses text fields into logical components (first name, last name; street number, street name, etc.)
  – Standardizes the representation of each parsed field (spelling, numerical range, capitalization, etc.)

• Commercial standardizers have very high value-added compared to home-grown standardizers but are very expensive
How to Standardize

• Inspect the file to refine strategy
• Use commercial software
• Write custom software (SAS, Fortran, C)
• Apply standardizer
• Inspect the file to refine strategy
Standardizing Names

Alternate spellings

1. Dr. William J. Smith, MD
2. Bill Smith
3. W. John Smith, MD
4. W.J. Smith, Jr.
5. Walter Jacob Smith, Sr.
# Standardized Names

<table>
<thead>
<tr>
<th>Pre</th>
<th>First</th>
<th>Mid</th>
<th>Last</th>
<th>Post1</th>
<th>Post2</th>
<th>Alt1</th>
<th>Std1</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Dr</td>
<td>William</td>
<td>J</td>
<td>Smith</td>
<td>MD</td>
<td></td>
<td>BWILL</td>
</tr>
<tr>
<td>2</td>
<td></td>
<td>Bill</td>
<td></td>
<td>Smith</td>
<td></td>
<td>William</td>
<td>BWILL</td>
</tr>
<tr>
<td>3</td>
<td></td>
<td>W</td>
<td>John</td>
<td>Smith</td>
<td>MD</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
<td>W</td>
<td>J</td>
<td>Smith</td>
<td>Jr</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Walter</td>
<td>Jacob</td>
<td>Smith</td>
<td></td>
<td>Sr</td>
<td></td>
<td>WALT</td>
</tr>
</tbody>
</table>
Standardizing Addresses

Many different pieces of information

1. 16 W Main Street #16
2. RR 2 Box 215
3. Fuller Building, Suite 405, 2nd door to the right
4. 14588 Highway 16W
## Standardized Addresses

<table>
<thead>
<tr>
<th>Pre</th>
<th>Hsnpm</th>
<th>Stnm</th>
<th>RR</th>
<th>Box</th>
<th>Post1</th>
<th>Post2</th>
<th>Unit</th>
<th>Unit</th>
<th>Bldg</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>W</td>
<td>16</td>
<td>Main</td>
<td></td>
<td>St</td>
<td></td>
<td>16</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
<td>2</td>
<td>215</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>405</td>
<td>Fuller</td>
</tr>
<tr>
<td>4</td>
<td>14588</td>
<td>Hwy</td>
<td>16</td>
<td></td>
<td></td>
<td>W</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Standardizing and language

• Standardizers are language- and “country”-specific
  – Address tokens may differ: “street”, “rue”, “calle”, “Straße”
  – Address components may differ:
    • 123 Main Street
      Normal, IL 61790
    • L 7,1
      D-68161 Mannheim
    • 1234 URB LOS OLMOS
      PONCE PR 00731-1235
Standardizing and language (2)

• Names differ
  – Juan, John, Johann, Yohan
• Variations of names differ:
  – Sepp, Zé, Joe -> Joseph
• Frequencies of names differ (will be important later)
  – Juan is frequent in Mexico, infrequent in Germany
Custom standardization

• Standardization may depend on the particular application

• Example OPM project
  – “Department of Defense”
  – “Department of Commerce”
  – The token “Department of” does not have distinguishing power, but comprises the majority of the “business name”
  – Similar: “Service”, “Bureau”
Implementing the Basic Matching Methodology

• Identifying comparison strategies:
  – Which variables to compare
  – String comparator metrics
  – Number comparison algorithms
  – Search and blocking strategies

• Ensuring computational feasibility of the task
  – Choice of software/hardware combination
  – Choice of blocking variables (runtimes quadratic in size of block)

• Estimating necessary parameters
Determination of Match Variables

• Must contain relevant information
• Must be informative (distinguishing power!)
• May not be on original file, but can be constructed (frequency, history information)
Blocking and Matching

• The essence of a probabilistic record link is iterating passes of the data files in which blocking variables (must match exactly) and matching variables (used to compute the agreement index) change roles.

• Blocking variables reduce the computational burden but increase the false non-match rate => solved by multiple passes

• As records are linked, the linked records are removed from the input files and the analyst can use fewer blocking variables to reduce the false non-matches.

• Matching variables increase the computational burden and manage the tradeoff between false match and false non-match errors
SSN Name Editing
Example

<table>
<thead>
<tr>
<th>Coded Name</th>
<th>Coded SSN</th>
<th>EIN</th>
<th>Earnings</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lesli Kay</td>
<td>1</td>
<td>A</td>
<td>$10</td>
</tr>
<tr>
<td>Leslie Kay</td>
<td>21</td>
<td>A</td>
<td>$10</td>
</tr>
<tr>
<td>Lesly Kai</td>
<td>31</td>
<td>B</td>
<td>$11</td>
</tr>
</tbody>
</table>

1’s tenure with A:
1’s employment history

Separations too high
Accessions too high

© 2011 John M. Abowd, Lars Vilhuber, all rights reserved
# SSN Name Editing Example

<table>
<thead>
<tr>
<th>Coded Name</th>
<th>Coded SSN</th>
<th>EIN</th>
<th>Earnings</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lesli Kay</td>
<td>1</td>
<td>A</td>
<td>$10</td>
</tr>
<tr>
<td>Leslie Kay</td>
<td>2</td>
<td>A</td>
<td>$10</td>
</tr>
<tr>
<td>Lesly Kai</td>
<td>3</td>
<td>B</td>
<td>$11</td>
</tr>
</tbody>
</table>

- **Separations too high**
- **Accessions too high**

![Diagram showing separations and accessions]

**T(1) T(2) T(3)**
Computed and observed variables

- Reclassification of information
- Blocking on a-priori information

<table>
<thead>
<tr>
<th>File</th>
<th>Name</th>
<th>SSN</th>
<th>Earn</th>
<th>Period</th>
<th>Gender</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>Lesli Kay</td>
<td>1</td>
<td>$10</td>
<td>T(2)</td>
<td>M</td>
<td>Hole</td>
</tr>
<tr>
<td>B</td>
<td>Leslie Kay</td>
<td>2</td>
<td>$10</td>
<td>T(2)</td>
<td>M</td>
<td>Plug</td>
</tr>
<tr>
<td>B</td>
<td>Lesly Kai</td>
<td>3</td>
<td>$10</td>
<td>T(4)</td>
<td>F</td>
<td>Plug</td>
</tr>
</tbody>
</table>

- Blocking: Earn, Period, Gender
- Match on: Name, SSN
Iterating

• First pass may block on a possibly miscoded variable

<table>
<thead>
<tr>
<th>File</th>
<th>Name</th>
<th>SSN</th>
<th>Earn</th>
<th>Period</th>
<th>Gender</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>Lesli Kay</td>
<td>1</td>
<td>$10</td>
<td>T(2)</td>
<td>M</td>
<td>Hole</td>
</tr>
<tr>
<td>B</td>
<td>Leslie Kay</td>
<td>2</td>
<td>$10</td>
<td>T(2)</td>
<td>F</td>
<td>Plug</td>
</tr>
<tr>
<td>B</td>
<td>Lesly Kai</td>
<td>3</td>
<td>$10</td>
<td>T(4)</td>
<td>F</td>
<td>Plug</td>
</tr>
</tbody>
</table>

• Block pass 1: Earn, Period, Gender
• Block pass 2: Earn, Period
Understanding Comparators

• Comparators need to account for
  – Typographical error
  – Significance of slight variations in numbers (both absolute and relative)
  – Possible variable inversions (first and last name flipped)
Soundex: History

• Used for historical analysis, archiving
• Origins in early 20\textsuperscript{th} century
• Available in many computer programs (SQL, SAS, etc.)
• Official “American” Soundex at National Archives:
  http://www.archives.gov/research/census/soundex.html
String Comparators: Soundex

- The first letter is copied unchanged
- Subsequent letters:
  
  - bfpv -> "1"  
  - cgjkqsxzç -> "2"  
  - dt -> "3"  
  - l -> "4"  
  - mnñ -> "5"  
  - r -> "6 "

- Other characters are ignored
- Repeated characters treated as single character.
- 4 chars, zero padded.
- For example, "SMITH" or "SMYTHE" would both be encoded as "S530".
String Comparators: Jaro

• First returns a value based on counting insertions, deletions, transpositions, and string length

• Total agreement weight is adjusted downward towards the total disagreement weight by some factor based on the value

• Custom adjustments (Winkler and others)
Comparing Numbers

• A difference of “34” may mean different things:
  – Age: a lot (mother-daughter? Different person)
  – Income: little
  – SSN or EIN: no meaning

• Some numbers may be better compared using string comparators
Number of Matching Variables

• In general, the distinguishing power of a comparison increases with the number of matching variables
• Exception: variables are strongly correlated, but poor indicators of a match
• Example: General business name and legal name associated with a license.
Determination of Match Parameters

• Need to determine the conditional probabilities $P(agree | M)$, $P(agree | U)$ for each variable comparison

• Methods:
  – Clerical review
  – Straight computation (Fellegi and Sunter)
  – EM algorithm (Dempster, Laird, Rubin, 1977)
  – Educated guess/experience
  – For $P(agree | U)$ and large samples (population): computed from random matching
Determination of Match Parameters (2)

- Fellegi & Sunter provide a solution when $\gamma$ represents three variables. The solution can be expressed as marginal probabilities $m_k$ and $u_k$.
- In practice, this method is used in many software applications.
- For $k>3$, method-of-moments or EM methods can be used.
Calculating the Agreement Index

- We need to compute $P(\gamma | M)$, $P(\gamma | U)$ and the agreement ratio $R(\gamma) = \frac{P(\gamma | M)}{P(\gamma | U)}$.
- The agreement index is $\ln R(\gamma)$.
- The critical assumption is conditional independence:
  
  $P(\gamma | M) = P(\gamma_1 | M) P(\gamma_2 | M) \ldots P(\gamma_K | M)$
  
  $P(\gamma | U) = P(\gamma_1 | U) P(\gamma_2 | U) \ldots P(\gamma_K | U)$

  where the subscript indicates an element of the vector $\gamma$.
- Implies that the agreement index can be written as:

  \[
  \ln R(\gamma) = \sum_{k=1}^{K} \ln \left( \frac{P(\gamma_k | M)}{P(\gamma_k | U)} \right)
  \]
Choosing $m$ and $u$ Probabilities

• Define

$$m_k = P(γ_k | M)$$
$$u_k = P(γ_k | U)$$

• These probabilities are often assessed using \textit{a priori} information or estimated from an expensive clerically edited link.
  – $m$ often set \textit{a priori} to 0.9
  – $u$ often set \textit{a priori} to 0.1

• Neither of these assumptions has much empirical support
Some Rules of Thumb

• Gender

\[ m_k = P(\gamma_k | M) \] is a function of the data (random miscodes of gender variable)

\[ u_k = P(\gamma_k | U) = 0.5 \] (unconditional on other variables). This may not be true for certain blocking variables: age, veteran status, etc. will affect this value

• Exact identifiers (SSN, SIN)

\[ m_k = P(\gamma_k | M) \] will depend on verification by the data provider. For example, embedded checksums will move this probability closer to 1.

\[ u_k = P(\gamma_k | U) << 0.1 \]
Marginal Probabilities: Educated Guesses for *Starting* Values

- $P(\text{agree on characteristic } X | M) =$
  - 0.9 if $X = \text{first, last name, age}$
  - 0.8 if $X = \text{house no., street name, other characteristic}$

- $P(\text{agree on characteristic } X | U) =$
  - 0.1 if $X = \text{first, last name, age}$
  - 0.2 if $X = \text{house no., street name, other characteristic}$

Note that *distinguishing power* of first name ($R(\text{first}) = 0.9/0.1 = 9$) is larger than the street name ($R(\text{street}) = 0.8/0.2 = 4$)
Match scores from Pass1

Kernel = epanechnikov, bandwidth = 1.0000
Marginal Probabilities: Better Estimates of $P(\text{agree} \mid M)$

- $P(\text{agree} \mid M)$ can be improved after a first match pass by a clerical review of match pairs:
  - Draw a sample of pairs
  - Manual review to determine “true” match status
  - Recompute $P(\text{agree} \mid M)$ based on known truth sample
Estimating $m$ and $u$ Using Matched Data

- If you have two files $\alpha$ and $\beta$ that have already been linked (perhaps clerically, perhaps with an exact link) then these estimates are available:

\[
\hat{m}_k = \frac{\sum_{(a,b) \in L} \gamma_k(a,b) = 1}{\sum_{\forall (a,b)} 1[(a,b) \in L]}
\]

\[
\hat{u}_k = \frac{\sum_{(a,b) \in U} \gamma_k(a,b) = 1}{\sum_{\forall (a,b)} 1[(a,b) \in U]}
\]

where $a \in \alpha, b \in \beta, \gamma(a,b) \in \Gamma$. 

© 2011 John M. Abowd, Lars Vilhuber, all rights reserved
Estimating $m$ and $u$ Probabilities Using EM


- Uses the identity

  $$P(\gamma) = P(\gamma|M)P(M) + P(\gamma|U)P(U)$$

- Imposes conditional independence
Clerical Editing

• Once the $m$ and $u$ probabilities have been estimated, cutoffs for the U, C, and L sets must be determined.

• This is usually done by setting preliminary cutoffs then clerically refining them.

• Often the $m$ and $u$ probabilities are tweaked as a part of this clerical review.
Estimating the False Match Rate

• This is usually done by clerical review of a run of the automated matcher.

Estimating the False Nonmatch Rate

• This is much harder.
• Often done by a clerical review of a sample of the non-match records.
• Since false nonmatching is relatively rare among the nonmatch pairs, this sample is often stratified by variables known to affect the match rate.
• Stratifying by the agreement index is a very effective way to estimate false nonmatch rates.
Post-processing

• Once matching software has identified matches, further processing may be needed:
  – Clean up
  – Carrying forward matching information
  – Reports on match rates
Example: Abowd and Vilhuber (2005)


• **Goal of the study**: Assess the impact of measurement error in tenure on aggregate measures of turnover


• Appendix A has a detailed description of matching passes and construction of additional blocking variables
Example: CPS-Decennial 2000 match


- *Goal of the study*: assess employment status on two independent questionnaires.
Software notes

• Idiosyncratic implementations
  – Merge two files in Stata/SAS/SQL/etc. (outer join or wide file), within blocks
  – For given m/u, compute agreement index on a per-variable basis for all
  – Order records within blocks, select match record
  – Generically, this works with numeric variables, but has issues when working with character variables
    • Typically, only Soundex available
    • Implementing string comparators (Jaro) may be difficult
Software notes: character matching

• SAS Data Quality
  – Is focused on “data quality”, “data cleaning”
  – Has functions for standardization and string matching
  – String matching is variant of Soundex: simplified strings based on generic algorithms
  – Issues: no probabilistic foundation, useful primarily for sophisticated string matching

• Combination of SAS DQ and idiosyncratic processing can yield powerful results
Software notes: SQL

• SQL languages are used in many contexts where matching is frequent
  – Oracle 10g R1 has Jaro-Winkler edit-distance implemented.
  – MySQL allows for some integration (possibly see http://androidaddicted.wordpress.com/2010/06/01/jaro-winkler-sql-code/)
  – PostGreSQL add-ons http://pgsimilarity.projects.postgresql.org/
Software notes: R

• R has a recent package (untested by us) that seems a fairly complete suite

• Does not include standardization
Acknowledgements

• This lecture is based in part on a 2000 lecture given by William Winkler, William Yancey and Edward Porter at the U.S. Census Bureau


• Examples are all purely fictitious, but inspired by true cases presented in the above lecture, in Abowd & Vilhuber (2005).