Record Linking, II

John M. Abowd and Lars Vilhuber
April 2007

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

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, 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

	Pre	First	Mid	Last	Post1	Post 2	Alt1	Std1
1	Dr	William	J	Smith	MD			BWILL
2		Bill		Smith			William	BWILL
3		W	John	Smith	MD			
4		W	J	Smith		Jr		
4		Walter	Jacob	Smith		Sr		WALT

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

	Pre 2	Hsnm	Stnm	RR	Box	Post1	Post2	Unit 1	Unit 2	Bldg
1	W	16	Main			St		16		
2				2	215					
3									405	Fuller
4		14588	Hwy	16			W			

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

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

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

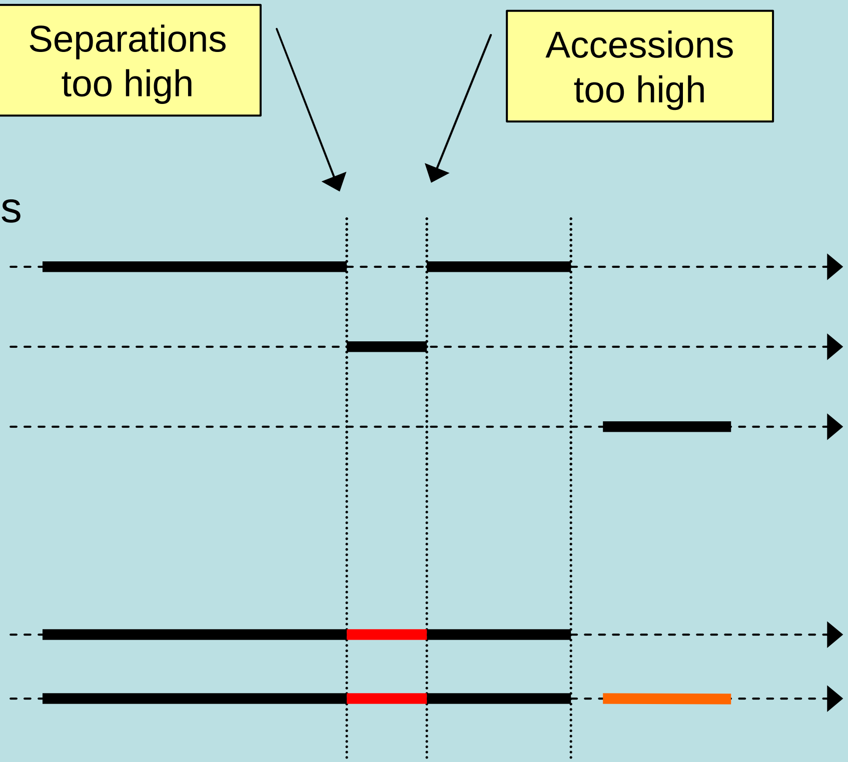
Determination of Match Variables

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

SSN Name Editing

Example

Coded Name	Coded SSN	EIN	Earnings
Leslie Kay	1	A	$10
Leslie Kay	~~2~~ 1	A	$10
Lesly Kai	~~3~~ 1	B	$11



1’s tenure with A:

1’s employment history

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)

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 variable
- 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 γ 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 $R(\gamma) = P(\gamma|M) / P(\gamma|U)$
- 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 γ.
- Implies that the agreement index can be written as:

$$\ln R(\gamma) = \sum_{k=1}^{K} \ln\left(\frac{P(\gamma_k \mid M)}{P(\gamma_k \mid U)}\right)$$

Choosing *m* and *u* Probabilities

- Define

  $m_k = P(\gamma_k|M)$

  $u_k = P(\gamma_k|U)$

- These probabilities are often assessed using *a priori* information or estimated from an expensive clerically edited link.
  - *m* often set *a priori* to 0.9
  - *u* often set *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(agree on characteristic X| M)=
  - 0.9 if X = first, last name, age
  - 0.8 if X = house no., street name, other characteristic
- P(agree on characteristic X| U)=
  - 0.1 if X = first, last name, age
  - 0.2 if X = house no., street name, other characteristic

Note that *distinguishing power* of first name (R(first)=0.9/0.1=9) is larger than the street name (R(street)=0.8/0.2=4)

Marginal Probabilities: Better Estimates of P(agree|M)

- P(agree|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(agree|M) based on known truth sample

Estimating *m* and *u* Using Matched Data

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

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

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

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

Estimating *m* and *u* Probabilities Using EM

- Based on Winkler 1988 "Using the EM Algorithm for Weight Computation in the Fellegi-Sunter Model of Record Linkage," *Proceedings of the Section on Survey Research Methods*, American Statistical Association, 667-671.
- 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.
- Some help is available from Belin, T. R., and Rubin, D. B. (1995), "A Method for Calibrating False-Match Rates in Record Linkage," *Journal of the American Statistical Association*, 90, 694-707.

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

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
- Some portions draw on Winkler (1995), “Matching and Record Linkage,” in B.G. Cox et. al. (ed.), *Business Survey Methods*, New York, J. Wiley, 355-384.
- Examples are all purely fictitious, but inspired by true cases presented in the above lecture, in Abowd & Vilhuber (2005).